INTERNATIONAL STANDARD ISO/IEC/IEEE 9945:2009
TECHNICAL CORRIGENDUM 1

Published 2013-08-15

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОМИССИЯ • COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

Information technology — Portable Operating System Interface (POSIX®) Base Specifications, Issue 7

TECHNICAL CORRIGENDUM 1

Technologies de l'information — Spécifications de base de l'interface pour la portabilité des systèmes (POSIX®), Issue 7

RECTIFICATIF TECHNIQUE 1

Technical Corrigendum 1 to ISO/IEC 9945:2009 was prepared by The Open Group and the Portable Applications Standards Committee of the Computer Society of the IEEE (as IEEE Std 1003.1-2008/Cor 1-2013). It was adopted by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces*, in parallel with its approval by the ISO/IEC national bodies, under the "fast-track procedure" defined in the Partner Standards Development Organization cooperation agreement between ISO and IEEE.

ICS 35.060 **Ref. No. ISO/IEC/IEEE 9945:2009/Cor.1:2013(E)**

Published in Switzerland

ISO/IEC/IEEE 9945:2009/Cor.1:2013(E)
(Corrigendum to
IEEE Std 1003.1-2008)

The Open Group Technical Standard
Base Specifications, Issue 7

IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)

Base Specifications, Issue 7—Technical Corrigendum 1

Sponsor

Portable Applications Standards Committee
of the
IEEE Computer Society

and
The Open Group

Approved 6 February 2013

IEEE-SA Standards Board

Approved 25 October 2012

The Open Group

Abstract: This Technical Corrigendum addresses problems discovered since the approval of IEEE Std 1003.1™ -2008.

Keywords: application program interface (API), argument, asynchronous, basic regular expression (BRE), batch job, batch system, built-in utility, byte, child, command language interpreter, CPU, extended regular expression (ERE), FIFO, file access control mechanism, IEEE 1003.1™, input/output (I/O), job control, network, portable operating system interface (POSIX®)

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Published 15 March 2013 by The Open Group. Printed in the United Kingdom by The Open Group.

The Open Group
Apex Plaza, Forbury Road, Reading, Berkshire RG1 1AX, U.K.

PDF: ISBN 978-0-7381-8265-0 STD698161

This standard has been prepared by the Austin Group. Feedback relating to the material contained within this standard may be submitted by using the Austin Group web site at www.opengroup.org/austin/defectform.html.

IEEE prohibits discrimination, harassment, and bullying. For more information, visit www.ieee.org/web/aboutus/whatis/policies/p9-26.html.

The Open Group is a global consortium that enables the achievement of business objectives through IT standards. With more than 400 member organizations, The Open Group has a diverse membership that spans all sectors of the IT community – customers, systems and solutions suppliers, tool vendors, integrators, and consultants, as well as academics and researchers – to:

- Capture, understand, and address current and emerging requirements, and establish policies and share best practices
- Facilitate interoperability, develop consensus, and evolve and integrate specifications and open source technologies
- Offer a comprehensive set of services to enhance the operational efficiency of consortia
- Operate the industry's premier certification service

Further information on The Open Group is available at www.opengroup.org.

The Open Group publishes a wide range of technical documentation, most of which is focused on development of Open Group Standards and Guides, but which also includes white papers, technical studies, certification and testing documentation, and business titles. Full details and a catalog are available at www.opengroup.org/bookstore.

Readers should note that updates – in the form of Corrigenda – may apply to any publication. This information is published at www.opengroup.org/corrigenda.

Notice and Disclaimer of Liability Concerning the Use of IEEE Documents: IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

Use of an IEEE Standard is wholly voluntary. IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon any IEEE Standard document.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained in its standards is free from patent infringement. IEEE Standards documents are supplied "**AS IS"**.

The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE standard is subjected to review at least every ten years. When a document is more than ten years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

Translations: The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE should be considered the approved IEEE standard.

Official Statements: A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

Comments on Standards: Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide consulting information or advice pertaining to IEEE Standards documents. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important to ensure that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to comments or questions except in those cases where the matter has previously been addressed. Any person who would like to participate in evaluating comments or revisions to an IEEE standard is welcome to join the relevant IEEE working group at http://standards.ieee.org/develop/wg/.

Comments on standards should be submitted to the following address:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854
USA

Photocopies: Authorization to photocopy portions of any individual standard for internal or personal use is granted by The Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to the Copyright Clearance Center. To arrange for payment of the licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Notice to users

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

This document is copyrighted by the IEEE. It is made available for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making this document available for use and adoption by public authorities and private users, the IEEE does not waive any rights in copyright to this document.

Updating of IEEE documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect. In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE-SA website at http://standards.ieee.org/index.html or contact the IEEE at the address listed previously. For more information about the IEEE Standards Association or the IEEE standards development process, visit the IEEE-SA website at http://standards.ieee.org/index.html.

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: http://standards.ieee.org/findstds/errata/index.html. Users are encouraged to check this URL for errata periodically.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE-SA website at http://standards.ieee.org/about/sasb/patcom/patents.html. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Participants

IEEE Std 1003.1™-2008/Cor 1-2013 was prepared by the Austin Group, sponsored by the Portable Applications Standards Committee of the IEEE Computer Society, The Open Group, and ISO/IEC JTC 1/SC22.

The Austin Group

At the time this IEEE standard was completed, the Austin Group had the following membership:

Andrew Josey, *Chair*

Donald W. Cragun, *Organizational Representative, IEEE PASC*
Nicholas Stoughton, *Organizational Representative, ISO/IEC JTC 1/SC22*
Mark S. Brown, *Organizational Representative, The Open Group*

Cathy Fox, *Technical Editor*

Austin Group Technical Reviewers

Eric Blake
Paolo Bonzini
Mark S. Brown
Geoff Clare
Donald W. Cragun
Ulrich Drepper
Laszlo Ersek
Hal Finkel
Mark Harris
Tom Honermann
Ross Johnson
Andrew Josey
Alexey Khoroshilov
Antoine Leca
Vincent Lefèvre
Matt McCutchen
Christoph Anton Mitterer
Peter O'Gorman
Peter Petrov
James C. Pugsley
Jörg Schilling
Ed Schouten
Konrad Schwarz
Martin Sebor
Nicolas Sitbon
Nicholas Stoughton
Jilles Tjoelker
Nathan Weeks
David A. Wheeler

Austin Group Working Group Members

Hans Aberg
Eitan Adler
Tanaka Akira
Matthias Andree
Matt W. Benjamin
Guido Berhoerster
Eric Blake
Hans Boehm
Paolo Bonzini
Davide Brini
Andries E. Brouwer
Mark S. Brown
David Butenhof
Albert Cahalan
Stephane Chazelas
Geoff Clare
Garrett Cooper
Donald W. Cragun
Matthew Dempsky
Casper Dik
Niall Douglas
Ulrich Drepper
Lawrence D.K.B. Dwyer
Paul Eggert
Julian Elischer
Laszlo Ersek
Bruce Evans
Sean C. Farley
Hal Finkel
Glenn Fowler
Mike Frysinger
Pierre Gaston
Thorsten Glaser
Philip Guenther
Bruno Haible
Mark Harris
David Holland
Tom Honermann
Ross Johnson
Andrew Josey
Dan Kegel
Michael Kerrisk
Alexey Khoroshilov
Josh Knight
Jonathan A. Kollasch
Bruce Korb
David Korn
Terry Lambert
Antoine Leca
Vincent Lefèvre
Wojtek Lerch
Scott Lurndal
Roger Marquis
Matt McCutchen
Esmaeil Mirzaee
Christoph Anton Mitterer
Joseph S. Myers
Alexander Nasonov
Jonathan Nieder
Carlos O'Donell
Peter O'Gorman
Peter Petrov
Wayne Pollock
James C. Pugsley
Chet Ramey
Xavier Roche

Jeroen Ruigrok van der Werven
Bart Schaefer
Jörg Schilling
Nico Schottelius
Ed Schouten
Konrad Schwarz
Jens Schweikhardt
Martin Sebor
Glen Seeds
Thor Lancelot Simon
Keld Simonsen
Nicolas Sitbon
Johannes Sixt
Paul Smith
Nicholas Stoughton
Marcel Telka
Jilles Tjoelker
Fred J. Tydeman
Nathan Weeks
David A. Wheeler
Mats D Wichmann
Garrett Wollman
Jörg Wunsch

The Open Group

When The Open Group approved the Base Specifications, Issue 7, Technical Corrigendum 1 on **25 October 2012**, the membership of The Open Group Base Working Group was as follows:

Andrew Josey, *Chair*
Mark S. Brown, *Austin Group Liaison*

Cathy Fox, *Technical Editor*

Base Working Group Members

Eric Blake
Mark S. Brown
David Butenhof
Geoff Clare
Donald W. Cragun
Lawrence D.K.B. Dwyer
Darrin Johnson
James C. Pugsley
William L. Toth
Kevin Van Vechten

Portable Applications Standards Committee (PASC)

Joseph M. Gwinn, *Chair*
Andrew Josey, *Functional Chair (Interpretations)*
Curtis Royster Jr., *Functional Chair (Logistics)*
Nicholas Stoughton, *Secretary*

IEEE

The following members of the individual balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Mark S. Brown
Michael Browne
Keith Chow
Donald W. Cragun
Thomas Dineen
Andrew Fieldsend
David Fuschi
Randall Groves
Joseph Gwinn
Werner Hoelzl
Andrew Josey
Piotr Karocki
Fernando Lucas Rodriguez
Greg Luri
Peter Petrov
Bartien Sayogo
Stephen Schwarm
Gil Shultz
Walter Struppler
Oren Yuen

When the IEEE-SA Standards Board approved this standard on 6 February 2013, it had the following membership:

John Kulick, *Chair*
Richard H. Hulett, *Past Chair*
Konstantinos Karachalios, *Secretary*

Masayuki Ariyoshi	Mark Halpin	Ron Petersen
Peter Balma	Gary Hoffman	Gary Robinson
Farooq Bari	Paul Houzé	Jon Walter Rosdahl
Ted Burse	Jim Hughes	Adrian Stephens
Wael William Diab	Michael Janezic	Peter Sutherland
Stephen Dukes	Joseph L. Koepfinger*	Yatin Trivedi
Jean-Philippe Faure	David J. Law	Phil Winston
Alexander Gelman	Oleg Logvinov	Yu Yuan

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Richard DeBlasio, *DOE Representative*
Michael Janezic, *NIST Representative*

Don Messina, *IEEE Standards Program Manager, Document Development*

Michael Kipness, *IEEE Standards Program Manager, Technical Program Development*

Introduction

This introduction is not part of IEEE Std 1003.1™-2008/Cor1-2013 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®) Base Specifications, Issue 7—Technical Corrigendum 1.

This Technical Corrigendum addresses issues raised in defect reports and interpretation requests submitted up to 17 May 2011 that meet all of the following criteria: (a) They are in the scope of the approved standard. (b) They contain no new APIs (functions/utilities); however, they may add enumeration symbols, non-function #defines, and reserve additional namespaces. (c) They address contradictions between different parts of the standard, or add consistency between the standard and overriding standards, or address security-related problems.

Contents

IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)

Base Specifications, Issue 7—Technical Corrigendum 1

NOTE: The editing instructions contained in this corrigendum define how to merge the material contained therein into the existing base standard to form the comprehensive standard.

1. Changes to Base Definitions

This section contains the set of changes to the text of the Base Definitions.

[*Note to reviewers: References to defect reports are provided to aid reviewers.*]

Change Number: Frontmatter/TC1/D5/0001 [246]

On Page: ix Line: 0 Section: Typographical Conventions

In the table of conventions, first column, change from:

Filename

to:

Filename, Pathname

Rationale: Austin Group Defect Report(s) applied: 246.
See http://austingroupbugs.net/view.php?id=246

Change Number: Frontmatter/TC1/D5/0002 [341]

On Page: ix Line: none Section: Typographical Conventions

Change:

'\'

in Note 2 to:

'\''

Rationale: Austin Group Defect Report(s) applied: 341.
See http://austingroupbugs.net/view.php?id=341

Change Number: XBD/TC1/D5/0001 [68]

On Page: 9 Line: 247 Section: 1.7.1 Codes

Change from:

[MX]IEC 60559 Floating-Point
The functionality described is optional. The functionality described is also an extension to the ISO C standard.

Where applicable, functions are marked with the MX margin legend in the SYNOPSIS section. Where additional semantics apply to a function, the material is identified by use of the MX margin legend.[/MX]

to:

[MX]IEC 60559 Floating-Point
The functionality described is optional. The functionality described is mandated by the ISO C standard only for implementations that define __STDC_IEC_559__.[/MX]

[MXX]IEC 60559 Floating-Point Extension
The functionality described is part of the IEC 60559 Floating-Point option, but is an extension to the ISO C standard.[/MXX]

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XBD/TC1/D5/0002 [291]

On Page: 21 Line: 730 Section: 2.1.5.1 Subprofiling Considerations

Change from:

filename argument

to:

pathname argument

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XBD/TC1/D5/0003 [82]

On Page: 24 Line: 830,839 Section: 2.1.5.2 XSI Option Groups

Add to the list of options (in alphabetical order):

_POSIX_THREAD_ROBUST_PRIO_INHERIT
_POSIX_THREAD_ROBUST_PRIO_PROTECT

Rationale: Austin Group Defect Report(s) applied: 82.
See http://austingroupbugs.net/view.php?id=82

Change Number: XBD/TC1/D5/0004 [291]

On Page: 54 Line: 1647 Section: 3.136 Dot

Change from:

In the context of naming files, the filename consisting of a single dot character ('.').

to:

In the context of naming files, the filename consisting of a single <period> character ('.').

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XBD/TC1/D5/0005 [291]

On Page: 55 Line: 1650 Section: 3.137 Dot-Dot

Change from:

The filename consisting solely of two dot characters ("..").

to:

The filename consisting solely of two <period> characters ("..").

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XBD/TC1/D5/0006 [371,425]

On Page: 60 Line: 1763 Section: 3.166 File Descriptor

Change from:

The value of a file descriptor is from zero to {OPEN_MAX}. A process can have no more than {OPEN_MAX} file descriptors open simultaneously.

to:

The value of a newly-created file descriptor is from zero to {OPEN_MAX}-1. A file descriptor can have a value greater than or equal to {OPEN_MAX} if the value of {OPEN_MAX} has decreased (see sysconf()) since the file descriptor was opened.

Rationale: Austin Group Defect Report(s) applied: 371,425.
See http://austingroupbugs.net/view.php?id=371
See http://austingroupbugs.net/view.php?id=425

Change Number: XBD/TC1/D5/0007 [291]

On Page: 60 Line: 1782 Section: 3.170 Filename

Change from:

A name consisting of 1 to {NAME_MAX} bytes used to name a file. The characters composing the name may be selected from the set of all character values excluding the <slash> character and the null byte. The filenames dot and dot-dot have special meaning. A filename is sometimes referred to as a "pathname component".

to:

A sequence of bytes consisting of 1 to {NAME_MAX} bytes used to name a file. The bytes composing the name shall not contain the <NUL> or <slash> characters. In the context of a pathname, each filename shall be followed by a <slash> or a <NUL> character; elsewhere, a filename followed by a <NUL> character forms a string (but not necessarily a character string). The filenames dot and dot-dot have special meaning. A filename is sometimes referred to as a "pathname component". See also XBD Section 3.266 Pathname.

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XBD/TC1/D5/0008 [291]

On Page: 60 Line: 1786 Section: 3.170

After Section 3.170, add a new section and then renumber existing sections:

3.1xx Filename String

A string consisting of a filename followed by a <NUL> character.

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XBD/TC1/D5/0009 [291]

On Page: 75 Line: 2145 Section: 3.266 Pathname

Change from:

A character string that is used to identify a file. In the context of POSIX.1-2008, a pathname may be limited to {PATH_MAX} bytes, including the terminating null byte. It has an optional beginning <slash>, followed by zero or more filenames separated by <slash> characters. A pathname may optionally contain one or more trailing <slash> characters. Multiple successive <slash> characters are considered to be the same as one <slash>, except for the case of exactly two leading <slash> characters.

to:

A string that is used to identify a file. In the context of <current version>, a pathname may be limited to {PATH_MAX} bytes, including the terminating null byte. It has optional beginning <slash> characters, followed by zero or more filenames separated by <slash> characters. A pathname can optionally contain one or more trailing <slash> characters. Multiple successive <slash> characters are considered to be the same as one <slash>, except for the case of exactly two leading <slash> characters.

Note: If a pathname consists of only bytes corresponding to characters from the portable filename character set (see XBD Section 3.276), <slash> characters, and a single terminating <NUL> character, the pathname will be usable as a character string in all supported locales; otherwise, the pathname might only be a string (rather than a character string). Additionally, since the single-byte encoding of the <slash> character is required to be the same across all locales and to not occur within a multi-byte character, references to a <slash> character within a pathname are well-defined even when the pathname is not a character string. However, this property does not necessarily hold for the remaining characters within the portable filename character set.

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XBD/TC1/D5/0010 [291]

On Page: 77 Line: 2193 Section: 3.275

After Section 3.275, add a new section and then renumber existing

sections:

3.2xx Portable Filename

A filename consisting only of characters from the portable filename character set.

Note: Applications should avoid using filenames that have the <hyphen> character as the first character since this may cause problems when filenames are passed as command line arguments.

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XBD/TC1/D5/0011 [291]

On Page: 77 Line: 2199 Section: 3.276 Portable Filename Character Set

Add a sentence at the end of the section:

See also XBD Section 3.266 Pathname.

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XBD/TC1/D5/0012 [181]

On Page: 97 Line: 2670 Section: 3.396 Thread

Change from:

Each thread has its own thread ID, scheduling priority and policy, errno value, thread-specific key/value bindings, and the required system resources to support a flow of control.

to:

Each thread has its own thread ID, scheduling priority and policy, errno value, floating point environment, thread-specific key/value bindings, and the required system resources to support a flow of control.

Rationale: Austin Group Defect Report(s) applied: 181.
See http://austingroupbugs.net/view.php?id=181

Change Number: XBD/TC1/D5/0013 [186]

On Page: 98 Line: 2685 Section: 3.399 Thread-Safe

Change from:

A function that may be safely invoked concurrently by multiple threads.

to:

A thread-safe function can be safely invoked concurrently with other calls to the same function, or with calls to any other thread-safe functions, by multiple threads.

Rationale: Austin Group Defect Report(s) applied: 186.
See http://austingroupbugs.net/view.php?id=186

Change Number: XBD/TC1/D5/0014 [135]

On Page: 108 Line: 2900 Section: 4.4 File Access Permissions

Delete the last paragraph in the section:

POSIX.1-2008 does not provide a way to open a directory for searching. It is unspecified whether directory search permission is granted based on the file access modes of the directory's file descriptor or on the mode of the directory at the time the directory is searched.

Rationale: Austin Group Defect Report(s) applied: 135.
See http://austingroupbugs.net/view.php?id=135
Text is removed that predated the addition of the O_SEARCH flag.

Change Number: XBD/TC1/D5/0015 [291]

On Page: 109 Line: 2915 Section: 4.7 Filename Portability

Change from:

Portable filenames shall not have the <hyphen> character as the first character since this may cause problems when filenames are passed as command line arguments.

to:

Note: Applications should avoid using filenames that have the <hyphen> character as the first character since this may cause problems when filenames are passed as command line arguments.

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XBD/TC1/D5/0016 [83]

On Page: 112 Line: 3040 Section: 4.12 Pathname Resolution

Change from:

A pathname that begins with two successive <slash> characters may be interpreted in an implementation-defined manner, although ...

to:

If a pathname begins with two successive <slash> characters, the first component following the leading <slash> characters may be interpreted in an implementation-defined manner, although ...

Rationale: Austin Group Defect Report(s) applied: 83.
See http://austingroupbugs.net/view.php?id=83

Change Number: XBD/TC1/D5/0017 [340]

On Page: 121 Line: 3303 Section: 5 File Format Notation

Change from:

Escape Sequences represent non-graphic characters.

to:

Escape Sequences represent non-graphic characters and the escape character (<backslash>).

Rationale: Austin Group Defect Report(s) applied: 340.
See http://austingroupbugs.net/view.php?id=340

Change Number: XBD/TC1/D5/0018 [302]

On Page: 127 Line: 3583 Section: 6.1 Portable Character Set

Change from:

... if an application accesses any pair of locales where the character encodings differ, or accesses data from an application running in a locale which has different encodings from the application's current locale, the results are unspecified.

to:

... if an application uses any pair of locales where the character encodings differ, or accesses data from an application using a locale which has different encodings from the locales used by the application, the results are unspecified.

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302

Change Number: XBD/TC1/D5/0019 [291]

On Page: 128 Line: 3589 Section: 6.1 Portable Character Set

After L3589 insert a new bullet:

* The encoded values associated with <slash> and <period> shall be invariant across all locales supported by the implementation.

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XBD/TC1/D5/0020 [216]

On Page: 128 Line: 3596 Section: 6.2 Character Encoding

Change from:

The POSIX locale contains the characters in Table 6-1 (on page 125), ...

to:

The POSIX locale contains the characters in Table 6-1 (on page 125)
and Table 6-2 (on page 130), ...

Rationale: Austin Group Defect Report(s) applied: 216.
See http://austingroupbugs.net/view.php?id=216

Change Number: XBD/TC1/D5/0021 [291]

On Page: 128 Line: 3619 Section: 6.2 Character Encoding

Add a sentence:

Likewise, the byte values used to encode <period> and <slash> shall not occur as part of any other character in any locale.

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XBD/TC1/D5/0022 [216]

On Page: 129 Line: 3661 Section: 6.4 Character Set Description File

Change from:

... the symbolic names and their corresponding encoding values shall be included in the file.

to:

... the symbolic names and their corresponding encoding values shall be included in the file. The encoding values shall each be represented in a single byte.

Rationale: Austin Group Defect Report(s) applied: 216.
See http://austingroupbugs.net/view.php?id=216

Change Number: XBD/TC1/D5/0023 [302]

On Page: 135 Line: 3820-3822 Section: 7.1 General

Change from:

The behavior of some of the C-language functions defined in the System Interfaces volume of POSIX.1-2008 shall also be modified based on the current locale, as defined by the last call to setlocale().

to:

The behavior of some of the C-language functions defined in the System Interfaces volume of POSIX.1-2008 shall also be modified based on a locale selection. The locale to be used by these functions can be selected in the following ways:

1. For functions such as isalnum_l() that take a locale object as an argument, a locale object can be obtained from newlocale() or duplocale() and passed to the function.

2. For functions that do not take a locale object as an argument, the current locale for the thread can be set by calling uselocale() or the global locale for the process can be set by calling setlocale(). Such functions shall use the current locale of the calling thread if one has been set for that thread; otherwise, they shall use the global locale.

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when per-thread locales were added.

Change Number: XBD/TC1/D5/0024 [302]

On Page: 135 Line: 3839 Section: 7.1 General

Change from:

Applications can select the desired locale by invoking the setlocale() function (or equivalent) with the appropriate value. If the function is invoked with an empty string, such as:

setlocale(LC_ALL, "");

the value of the corresponding environment variable is used.

to:

Applications can select the desired locale by calling the newlocale() or setlocale() function with the appropriate value. If the function is invoked with an empty string, such as:

newlocale(LC_ALL_MASK, "", (locale_t)0);

or:

setlocale(LC_ALL, "");

the value of the corresponding environment variable is used.

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when per-thread locales were added.

Change Number: XBD/TC1/D5/0025 [22,427]

On Page: 164 Line: 5141-5146 Section: 7.3.6 LC_MESSAGES

On L5142 in the description of the yesexpr keyword, change from:

... the acceptable affirmative response ...

to:

... acceptable affirmative responses ...

On L5144-5145 in the description of the noexpr keyword, change from:

... the acceptable negative response ...

to:

... acceptable negative responses ...

Rationale: Austin Group Defect Report(s) applied: 22,427.
See http://austingroupbugs.net/view.php?id=22
See http://austingroupbugs.net/view.php?id=427

Change Number: XBD/TC1/D5/0026 [167]

On Page: 173 Line: 5476 Section: 8.1 Environment Variable Definition

Change from:

... manipulating the environ variable ...

to:

... assigning a new value to the environ variable ...

Rationale: Austin Group Defect Report(s) applied: 167.
See http://austingroupbugs.net/view.php?id=167

Change Number: XBD/TC1/D5/0027 [167]

On Page: 173 Line: 5478 Section: 8.1 Environment Variable Definition

Add a new paragraph after L5478:

If the application modifies the pointers to which environ points, the behavior of all interfaces described in the System Interfaces volume of POSIX.1-2008 is undefined.

Rationale: Austin Group Defect Report(s) applied: 167.
See http://austingroupbugs.net/view.php?id=167

Change Number: XBD/TC1/D5/0028 [291]

On Page: 175 Line: 5566 Section: 8.2 Internationalization Variables

In the description of NLSPATH, change from:

filename

to:

pathname

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XBD/TC1/D5/0029 [302]

On Page: 177 Line: 5636 Section: 8.2 Internationalization Variables

Change from:

At runtime, these values are bound to the locale of a process by calling the setlocale() function.

to:

These environment variables are used by the newlocale() and setlocale() functions, and by the standard utilities.

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302

Change Number: XBD/TC1/D5/0030 [139]

On Page: 178 Line: 5674 Section: 8.3 Other Environment Variables

In the description of the PATH variable, change from:

When a non-zero-length prefix is applied to this filename, a <slash> shall be inserted between the prefix and the filename.

to:

When a non-zero-length prefix is applied to this filename, a <slash> shall be inserted between the prefix and the filename if the prefix did not end in <slash>.

Rationale: Austin Group Defect Report(s) applied: 139.
See http://austingroupbugs.net/view.php?id=139
This change is for consistency with the previous change to CDPATH made in Interpretation 1003.1-2001 #199

Change Number: XBD/TC1/D5/0031 [84]

On Page: 178 Line: 5687 Section: 8.3 Other Environment Variables

At the end of the PATH variable, add a new paragraph:

Since colon is a separator in this context, directory names that might be used in PATH should not include a colon character.

Rationale: Austin Group Defect Report(s) applied: 84.
See http://austingroupbugs.net/view.php?id=84

Change Number: XBD/TC1/D5/0032 [305,427]

On Page: 186 Line: 5995 Section: 9.3.5 RE Bracket Expression

At L5995, add a new paragraph:

8. If a bracket expression contains at least three list elements, where

the first and last list element are the same single-character element of <period>, <equals-sign>, or <colon>, then it is unspecified whether the bracket expression will be treated as a collating symbol, equivalence class, or character class, respectively; treated as a matching list expression; or rejected as an error.

Rationale: Austin Group Defect Report(s) applied: 305,427.
See http://austingroupbugs.net/view.php?id=305
See http://austingroupbugs.net/view.php?id=427

Change Number: XBD/TC1/D5/0033 [134]

On Page: 197 Line: 6402 Section: 10.1 Directory Structure and Files

Change from:

/dev/null An infinite data source and data sink. ...

to:

/dev/null An empty data source and infinite data sink. ...

Rationale: Austin Group Defect Report(s) applied: 134.
See http://austingroupbugs.net/view.php?id=134
Improve the description of /dev/null.

Change Number: XBD/TC1/D5/0034 [79]

On Page: 201 Line: 6531-6542 Section: 11.1.4 Terminal Access Control

On L6531 change from:

... if the reading process is ignoring or blocking the SIGTTIN signal, ...

to:

... if the reading process is ignoring the SIGTTIN signal or the reading thread is blocking the SIGTTIN signal, ...

On L6539 change from:

... and the process is ignoring or blocking the SIGTTOU signal, ...

to:

... and the process is ignoring the SIGTTOU signal or the writing thread is blocking the SIGTTOU signal, ...

On L6540 change from:

If TOSTOP is set, and the process group of the writing process is orphaned, and the writing process is not ignoring or blocking the SIGTTOU signal, ...

to:

If TOSTOP is set, the process group of the writing process is orphaned, the writing process is not ignoring the SIGTTOU signal, and the writing thread is not blocking the SIGTTOU signal, ...

Rationale: Austin Group Defect Report(s) applied: 79.
See http://austingroupbugs.net/view.php?id=79

Change Number: XBD/TC1/D5/0035 [271]

On Page: 205 Line: 6710 Section: 11.1.11 Closing A Terminal Device File

Change from:

The last process to close a terminal device file shall cause any output to be sent to the device and any input to be discarded.

to:

The last process to close a terminal device file shall cause any output to be sent to the device and shall cause any input to be discarded.

Rationale: Austin Group Defect Report(s) applied: 271.
See http://austingroupbugs.net/view.php?id=271
The text can mean two things:
1. "any output to be sent" and "any input" shall be discarded.
2. "any output" will be sent and "any input" shall be discarded.

Change Number: XBD/TC1/D5/0036 [79]

On Page: 211 Line: 6962 Section: 11.2.5 Local Modes

Change from:

Processes that are blocking or ignoring SIGTTOU signals are excepted and allowed to produce output, and the SIGTTOU signal shall not be sent.

to:

If the writing process is ignoring the SIGTTOU signal or the writing thread is blocking the SIGTTOU signal, the process is allowed to produce output, and the SIGTTOU signal shall not be sent.

Rationale: Austin Group Defect Report(s) applied: 79.
See http://austingroupbugs.net/view.php?id=79

Change Number: XBD/TC1/D5/0037 [171]

On Page: 216 Line: 7120 Section: 12.2 Utility Syntax Guidelines

In Guideline 5 change from:

Options without option-arguments should be accepted when grouped behind one '-' delimiter.

to:

One or more options without option-arguments, followed by at most one option that takes an option-argument, should be accepted when grouped behind one '-' delimiter.

Rationale: Austin Group Defect Report(s) applied: 171.
See http://austingroupbugs.net/view.php?id=171

Change Number: XBD/TC1/D5/0038 [98]

On Page: 220 Line: 7234 Section: <aio.h>

In the DESCRIPTION section, add FSC|SIO shading to the declaration of aio_fsync().

Rationale: Austin Group Defect Report(s) applied: 98.
See http://austingroupbugs.net/view.php?id=98

Change Number: XBD/TC1/D5/0039 [291]

On Page: 231 Line: 7575 Section: <dirent.h>

In the DESCRIPTION section, change from:

d_name[] Name of entry.

to:

d_name[] Filename string of entry.

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XBD/TC1/D5/0040 [291]

On Page: 231 Line: 7577 Section: <dirent.h>

In the DESCRIPTION section, change from:

The character array d_name is of unspecified size, but the number of bytes preceding the terminating null byte shall not exceed {NAME_MAX}.

to:

The array d_name is of unspecified size, but shall contain a filename of at most {NAME_MAX} bytes followed by a terminating null byte.

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XBD/TC1/D5/0041 [291]

On Page: 232 Line: 7612 Section: <dirent.h>

In the RATIONALE section, change from:

... number of characters provided matches (or only slightly exceeds) the length of the filename.

to:

... number of bytes provided matches (or only slightly exceeds) the length of the filename string.

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XBD/TC1/D5/0042 [206]

On Page: 232 Line: 7629 Section: <dirent.h>

In the CHANGE HISTORY section, change from:

... fopendir() ...

to:

... fdopendir() ...

Rationale: Austin Group Defect Report(s) applied: 206.
See http://austingroupbugs.net/view.php?id=206

Change Number: XBD/TC1/D5/0043 [324]

On Page: 236 Line: 7739 Section: <errno.h>

In the DESCRIPTION section, change from:

[ENOTDIR] Not a directory.

to:

[ENOTDIR] Not a directory or a symbolic link to a directory.

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324

Change Number: XBD/TC1/D5/0044 [274]

On Page: 239 Line: 7848 Section: <fcntl.h>

In the DESCRIPTION section, after:

The <fcntl.h> header shall define the following symbolic constants for use as the file access modes for open(), openat(), and fcntl().

add:

The values shall be unique, except that O_EXEC and O_SEARCH may have equal values.

Rationale: Austin Group Defect Report(s) applied: 274.
See http://austingroupbugs.net/view.php?id=274

Change Number: XBD/TC1/D5/0045 [78,432]

On Page: 240 Line: 7874-7879 Section: <fcntl.h>

In the DESCRIPTION section, delete L7874 to L7875.

The <fcntl.h> header shall define the following symbolic constants as values for the flag used by open() and openat():

In the DESCRIPTION section, move L7876 to L7879:

O_CLOEXEC The FD_CLOEXEC flag associated with the new descriptor shall be set to close the file descriptor upon execution of an exec family function.

O_DIRECTORY Fail if not a directory.
O_NOFOLLOW Do not follow symbolic links.

into the section at L7829 to L7834, so that it reads:

The <fcntl.h> header shall define the following symbolic constants as file creation flags for use in the oflag value to open() and openat(). The values shall be bitwise-distinct and shall be suitable for use in #if preprocessing directives.

O_CLOEXEC The FD_CLOEXEC flag associated with the new descriptor shall be set to close the file descriptor upon execution of an exec family function.

O_CREAT Create file if it does not exist.

O_DIRECTORY Fail if not a directory.

O_EXCL Exclusive use flag.

O_NOCTTY Do not assign controlling terminal.

O_NOFOLLOW Do not follow symbolic links.

O_TRUNC Truncate flag.

Rationale: Austin Group Defect Report(s) applied: 78,432.
See http://austingroupbugs.net/view.php?id=78
See http://austingroupbugs.net/view.php?id=432

Change Number: XBD/TC1/D5/0046 [346]

On Page: 249 Line: 8198 Section: <float.h>

In the DESCRIPTION section, add a paragraph within the *_MAX_EXP bullet:

[CX]Additionally, FLT_MAX_EXP shall be at least as large as FLT_MANT_DIG, DBL_MAX_EXP shall be at least as large as DBL_MANT_DIG, and LDBL_MAX_EXP shall be at least as large as LDBL_MANT_DIG; which has the effect that FLT_MAX, DBL_MAX, and LDBL_MAX are integral.[/CX]

Rationale: Austin Group Defect Report(s) applied: 346.
See http://austingroupbugs.net/view.php?id=346

Change Number: XBD/TC1/D5/0047 [346]

On Page: 250 Line: 8224 Section: <float.h>

In the RATIONALE section, change from:

None.

to:

All known hardware floating-point formats satisfy the property that the exponent range is larger than the number of mantissa digits. The ISO C standard permits a floating-point format where this property is not true, such that the largest finite value would not be integral; however, it is unlikely that there will ever be hardware support for such a floating-point format, and it introduces boundary cases that portable programs should not have to be concerned with (for example, a non-integral DBL_MAX means that ceil() would have to worry about overflow). Therefore, this standard imposes an additional requirement that the largest representable finite value is integral.

Rationale: Austin Group Defect Report(s) applied: 346.
See http://austingroupbugs.net/view.php?id=346

Change Number: XBD/TC1/D5/0048 [403]

On Page: 254 Line: 8333 Section: <ftw.h>

In the DESCRIPTION section, change from:

FTW_F File.

to:

FTW_F Non-directory file.

Rationale: Austin Group Defect Report(s) applied: 403.
See http://austingroupbugs.net/view.php?id=403

Change Number: XBD/TC1/D5/0049 [24]

On Page: 258 Line: 8441-8442 Section: <grp.h>

In the DESCRIPTION, add XSI shading to endgrent() and getgrent().

Rationale: Austin Group Defect Report(s) applied: 24.
See http://austingroupbugs.net/view.php?id=24

Change Number: XBD/TC1/D5/0050 [211]

On Page: 261 Line: 8510- Section: <inttypes.h>

In the DESCRIPTION section change from:

The <inttypes.h> header shall define at least the following type:

imaxdiv_t Structure type that is the type of the value returned by the imaxdiv() function.

to:

The <inttypes.h> header shall define at least the following types:

imaxdiv_t Structure type that is the type of the value returned by the imaxdiv() function.

[CX]wchar_t As described in <stddef.h>.[/CX]

Rationale: Austin Group Defect Report(s) applied: 211.
See http://austingroupbugs.net/view.php?id=211

Change Number: XBD/TC1/D5/0051 [107]

On Page: 264 Line: 8623 Section: <langinfo.h>

In the DESCRIPTION section, replace <nl/types.h> with <nl_types.h>.

Rationale: Austin Group Defect Report(s) applied: 107.
See http://austingroupbugs.net/view.php?id=107
Corrected an editorial error in referencing <nl_types.h>.

Change Number: XBD/TC1/D5/0052 [108]

On Page: 268 Line: 8756 Section: <limits.h>

In the DESCRIPTION section, change from:

Many of the symbols listed here are not defined by the ISO/IEC 9899: 1999 standard. Such symbols are not shown as CX shaded.

to:

(Shading block as follows on L8756-8757)

Many of the symbols listed here are not defined by the ISO/IEC 9899: 1999 standard. Such symbols are not shown as CX shaded, except under the heading "Numerical Limits".

Rationale: Austin Group Defect Report(s) applied: 108.
See http://austingroupbugs.net/view.php?id=108

Change Number: XBD/TC1/D5/0053 [291]

On Page: 272 Line: 8933 Section: <limits.h>

In the DESCRIPTION (<limits.h> {NAME_MAX}) change from:

... bytes in a filename (not including the terminating null).

to:

... bytes in a filename (not including the terminating null of a filename string).

Rationale: Austin Group Defect Report(s) applied: 291.

See http://austingroupbugs.net/view.php?id=291

Change Number: XBD/TC1/D5/0054 [182,427]

On Page: 274 Line: 9020 Section: <limits.h>

In the DESCRIPTION section, Minimum Values, change from:

A conforming implementation shall provide values at least this large.

to:

For each of these limits, a conforming implementation shall provide a value at least this large or shall have no limit.

Rationale: Austin Group Defect Report(s) applied: 182,427.
See http://austingroupbugs.net/view.php?id=182
See http://austingroupbugs.net/view.php?id=427

Change Number: XBD/TC1/D5/0055 [291]

On Page: 275 Line: 9060 Section: <limits.h>

In the DESCRIPTION ({_POSIX_NAME_MAX}), change from:

... bytes in a filename (not including the terminating null).

to:

... bytes in a filename (not including the terminating null of a filename string).

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XBD/TC1/D5/0056 [371]

On Page: 275 Line: 9066 Section: <limits.h>

In the DESCRIPTION section, change from:

Maximum number of files that one process can have open at any one time.

to:

A value one greater than the maximum value that the system may assign to a newly-created file descriptor.

Rationale: Austin Group Defect Report(s) applied: 371.
See http://austingroupbugs.net/view.php?id=371

Change Number: XBD/TC1/D5/0057 [291]

On Page: 277 Line: 9182 Section: <limits.h>

In the DESCRIPTION ({_XOPEN_NAME_MAX}), change from:

... bytes in a filename (not including the terminating null).

to:

... bytes in a filename (not including the terminating null of a filename string).

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XBD/TC1/D5/0058 [108]

On Page: 278 Line: 9218 Section: <limits.h>

In the DESCRIPTION section, add CX shading to:

LONG_BIT (P278 L9218-9220)
WORD_BIT (P279 L9260-9262)
SSIZE_MAX (P279 L9242-9244)

Rationale: Austin Group Defect Report(s) applied: 108.
See http://austingroupbugs.net/view.php?id=108

Change Number: XBD/TC1/D5/0059 [291]

On Page: 280 Line: 9311 Section: <limits.h>

In the RATIONALE section, change "filenames" to "pathnames".

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XBD/TC1/D5/0060 [301,427]

On Page: 284 Line: 9450 Section: <locale.h>

In the DESCRIPTION section change from:

... a special object descriptor used by the uselocale() function.

to:

... a special object descriptor used by the duplocale() and uselocale() functions.

Rationale: Austin Group Defect Report(s) applied: 301,427.
See http://austingroupbugs.net/view.php?id=301
See http://austingroupbugs.net/view.php?id=427

Change Number: XBD/TC1/D5/0061 [355]

On Page: 303 Line: 10141 Section: <netinet/in.h>

In the DESCRIPTION section, change from:

Values of this type shall be cast by applications to struct sockaddr
for use with socket functions.

to:

Pointers to this type shall be cast by applications to struct sockaddr *
for use with socket functions.

Rationale: Austin Group Defect Report(s) applied: 355.
See http://austingroupbugs.net/view.php?id=355

Change Number: XBD/TC1/D5/0062 [208]

On Page: 328 Line: 10963 Section: <signal.h>

In the DESCRIPTION section, change the margin marker notation for the
SIG_HOLD macro from:

OB CX

to:

OB XSI

Rationale: Austin Group Defect Report(s) applied: 208.
See http://austingroupbugs.net/view.php?id=208
SIG_HOLD shading and margin marker notation is updated from
OB CX to OB XSI as it is used with the sigset function which is an
XSI function.

Change Number: XBD/TC1/D5/0063 [80]

On Page: 330 Line: 11032 Section: <signal.h>

In the DESCRIPTION section, in the table of signals, change the shading
on SIGPROF from OB XSR to OB XSI.

Rationale: Austin Group Defect Report(s) applied: 80.
See http://austingroupbugs.net/view.php?id=80

Change Number: XBD/TC1/D5/0064 [157]

On Page: 331 Line: 11055 Section: <signal.h>

In the DESCRIPTION section, change the shading margin marker from XSI
to CX on the following text:

The storage occupied by sa_handler and sa_sigaction may overlap, and a
conforming application shall not use both simultaneously.

Rationale: Austin Group Defect Report(s) applied: 157.
See http://austingroupbugs.net/view.php?id=157

Change Number: XBD/TC1/D5/0065 [291,427]

On Page: 351 Line: 11796 Section: <stdio.h>

In the DESCRIPTION section, at P351 L11796 (XBD <stdio.h> FILENAME_MAX), change from "filename string" to "pathname".

Rationale: Austin Group Defect Report(s) applied: 291,427.
See http://austingroupbugs.net/view.php?id=291
See http://austingroupbugs.net/view.php?id=427

Change Number: XBD/TC1/D5/0066 [197]

On Page: 356 Line: 11992 Section: <stdlib.h>

In the DESCRIPTION section, add the CX margin marker and shading to the getsubopt() function prototype.

Rationale: Austin Group Defect Report(s) applied: 197.
See http://austingroupbugs.net/view.php?id=197
The getsubopt() function is an extension to the ISO C standard.

Change Number: XBD/TC1/D5/0067 [355]

On Page: 382 Line: 12813-12817 Section: <sys/socket.h>

In the DESCRIPTION section, change from:

When a sockaddr_storage structure is cast as a sockaddr structure, the ss_family field of the sockaddr_storage structure shall map onto the sa_family field of the sockaddr structure. When a sockaddr_storage structure is cast as a protocol-specific address structure, the ss_family field shall map onto a field of that structure that is of type sa_family_t and that identifies the protocol's address family.

to:

When a pointer to a sockaddr_storage structure is cast as a pointer to a sockaddr structure, the ss_family field of the sockaddr_storage structure shall map onto the sa_family field of the sockaddr structure. When a pointer to a sockaddr_storage structure is cast as a pointer to a protocol-specific address structure, the ss_family field shall map onto a field of that structure that is of type sa_family_t and that identifies the protocol's address family.

Rationale: Austin Group Defect Report(s) applied: 355.
See http://austingroupbugs.net/view.php?id=355

Change Number: XBD/TC1/D5/0068 [207]

On Page: 388 Line: 13042 Section: <sys/stat.h>

In the DESCRIPTION section, add the XSI margin marker, and shade blkcnt_t and blksize_t.

Rationale: Austin Group Defect Report(s) applied: 207.
See http://austingroupbugs.net/view.php?id=207

Add notation to show that blkcnt_t and blksize_t are XSI extensions.

Change Number: XBD/TC1/D5/0069 [210]

On Page: 398 Line: 13360f Section: <sys/types.h>

In the DESCRIPTION section, remove the XSI margin marker and shading from fsblkcnt_t and fsfilcnt_t.

Rationale: Austin Group Defect Report(s) applied: 210.
See http://austingroupbugs.net/view.php?id=210
The statvfs structure, where fsblkcnt_t and fsfilcnt_t are used, is now in base.

Change Number: XBD/TC1/D5/0070 [28]

On Page: 399 Line: 13389 Section: <sys/types.h>

In the DESCRIPTION section, change from:

<blank name> Also used to identify a trace stream attributes object

to:

trace_attr_t Used to identify a trace stream attributes object

(The existing OB TRC shading should apply to the whole line.)

Rationale: Austin Group Defect Report(s) applied: 28.
See http://austingroupbugs.net/view.php?id=28

Change Number: XBD/TC1/D5/0071 [376]

On Page: 399 Line: 13415 Section: <sys/types.h>

In the DESCRIPTION section, between L13415 and 13416, add a bullet:

* dev_t shall be an integer type.

Rationale: Austin Group Defect Report(s) applied: 376.
See http://austingroupbugs.net/view.php?id=376

Change Number: XBD/TC1/D5/0072 [210]

On Page: 399 Line: 13417 Section: <sys/types.h>

In the DESCRIPTION section, remove the XSI margin marker and shading from fsblkcnt_t and fsfilcnt_t.

Rationale: Austin Group Defect Report(s) applied: 210.
See http://austingroupbugs.net/view.php?id=210
The statvfs structure, where fsblkcnt_t and fsfilcnt_t are used, is now in base.

Change Number: XBD/TC1/D5/0073 [327]

On Page: 399 Line: 13420 Section: <sys/types.h>

In the DESCRIPTION section, change from:

time_t and clock_t shall be integer or real-floating types.

to:

clock_t shall be an integer or real-floating type. [CX]time_t shall be an integer type.[/CX]

Rationale: Austin Group Defect Report(s) applied: 327.
See http://austingroupbugs.net/view.php?id=327

Change Number: XBD/TC1/D5/0074 [355]

On Page: 403 Line: 13517 Section: <sys/un.h>

In the DESCRIPTION section, change from:

Values of this type shall be cast by applications to struct sockaddr for use with socket functions.

to:

Pointers to this type shall be cast by applications to struct sockaddr * for use with socket functions.

Rationale: Austin Group Defect Report(s) applied: 355.
See http://austingroupbugs.net/view.php?id=355

Change Number: XBD/TC1/D5/0075 [357,427]

On Page: 418 Line: 13996 Section: <tgmath.h>

In the DESCRIPTION section, in the table at L13996-13997, scalbln() should sort before scalbn().

Rationale: Austin Group Defect Report(s) applied: 357,427.
See http://austingroupbugs.net/view.php?id=357
See http://austingroupbugs.net/view.php?id=427

Change Number: XBD/TC1/D5/0076 [212]

On Page: 422 Line: 14153 Section: <time.h>

In the DESCRIPTION section, add the CX margin marker notation and shading to L14153-14155. (The <time.h> header.... TIMER_ABSTIME ...)

Rationale: Austin Group Defect Report(s) applied: 212.
See http://austingroupbugs.net/view.php?id=212

Change Number: XBD/TC1/D5/0077 [212]

On Page: 422 Line: 14159 Section: <time.h>

In the DESCRIPTION section, change from:

... a progrm defines ...

to:

... a program defines ...

Rationale: Austin Group Defect Report(s) applied: 212.
See http://austingroupbugs.net/view.php?id=212
This is an editorial correction.

Change Number: XBD/TC1/D5/0078 [311]

On Page: 430 Line: 14446 Section: <unistd.h>

In the DESCRIPTION section, after:

For implementations conforming to POSIX.1-2008, the value shall be 200809L.

add (to the same paragraph):

For profile implementations that define _POSIX_SUBPROFILE (see Section 2.1.5.1) in <unistd.h>, _POSIX2_VERSION may be left undefined or be defined with the value -1 to indicate that the Shell and Utilities volume is not supported. In this case a call to sysconf(_SC_2_VERSION) shall return either 200809L or -1 indicating that the Shell and Utilities volume is or is not, respectively, supported at runtime.

At L14455 after:

The following symbolic constants, if defined in <unistd.h>, shall have a value of -1, 0, or greater, unless otherwise specified below.

add:

For profile implementations that define _POSIX_SUBPROFILE (see Section 2.1.5.1) in <unistd.h>, constants described below as always having a value greater than zero need not be defined and, if defined, may have a value of -1, 0, or greater.

Rationale: Austin Group Defect Report(s) applied: 311.
See http://austingroupbugs.net/view.php?id=311

Change Number: XBD/TC1/D5/0079 [209]

On Page: 439-440 Line: 14870-14875 Section: <unistd.h>

In the DESCRIPTION section, add the XSI margin marker and shading for L14870 to 14875 (symbolic constants F_LOCK ... F_ULOCK).

Rationale: Austin Group Defect Report(s) applied: 209.
See http://austingroupbugs.net/view.php?id=209

Change Number: XBD/TC1/D5/0080 [360]

On Page: 443 Line: 15048 Section: <unistd.h>

In the DESCRIPTION section, delete L15048 (ctermid() declaration in XBD <unistd.h>).

At L15124 (<unistd.h>), add a sentence:

Implementations may also include the ctermid() prototype as defined in <stdio.h>.

Tag the entire paragraph starting on L15124 including the above new sentence with the OB margin marking.

Rationale: Austin Group Defect Report(s) applied: 360.
See http://austingroupbugs.net/view.php?id=360
Some implementations have had a declaration for ctermid() and this should still be allowed in Issue 7. Having ctermid() and pthread_atfork() prototypes in <unistd.h>, however, are namespace pollution issues that should be corrected in the next revision.

Change Number: XBD/TC1/D5/0081 [380]

On Page: 455 Line: 15531-15532 Section: <wchar.h>

In the DESCRIPTION section,in the wcpcpy() and wcpncpy() prototype declarations change from:

restrict*

to:

*restrict

Rationale: Austin Group Defect Report(s) applied: 380.
See http://austingroupbugs.net/view.php?id=380

2. Changes to System Interfaces

This section contains the set of changes to the text of the System Interfaces.

[*Note to reviewers: References to defect reports are provided to aid reviewers.*]

Change Number: XSH/TC1/D5/0001 [316]

On Page: 477 Line: 16210 Section: 2.3 Error Numbers

Change from:

The ERRORS section on each reference page specifies which error conditions shall be detected by all implementations ("shall fail") and which may be optionally detected by an implementation ("may fail"). If no error condition is detected, the action requested shall be successful.

to:

The ERRORS section on each reference page specifies which error conditions shall be detected by all implementations ("shall fail") and which may be optionally detected by an implementation ("may fail"). If no error condition is detected, the action requested shall be successful. If an error condition is detected, the action requested may have been partially performed, unless otherwise stated.

Rationale: Austin Group Defect Report(s) applied: 316.
See http://austingroupbugs.net/view.php?id=316

Change Number: XSH/TC1/D5/0002 [146]

On Page: 482 Line: 16419 Section: 2.3 Error Numbers

For the [ENOTDIR] description, change from:

[ENOTDIR] Not a directory. A component of the specified pathname exists, but it is not a directory, when a directory was expected.

to:

[ENOTDIR] Not a directory. A component of the specified pathname exists, but it is not a directory, when a directory was expected; or an attempt was made to create a non-directory file, and the specified pathname contains at least one non-<slash> character and ends with one or more trailing <slash> characters.

Rationale: Austin Group Defect Report(s) applied: 146.
See http://austingroupbugs.net/view.php?id=146

Change Number: XSH/TC1/D5/0003 [112]

On Page: 482 Line: 16430 Section: 2.3 Error Numbers

Change from:

[ENOTSUP] Not supported. The implementation does not support this feature of the Realtime Option Group.

to:

[ENOTSUP] Not supported. The implementation does not support the requested feature or value.

Rationale: Austin Group Defect Report(s) applied: 112.
See http://austingroupbugs.net/view.php?id=112

Change Number: XSH/TC1/D5/0004 [77]

On Page: 485 Line: 16549 Section: 2.4.1 Signal Generation and Delivery

Change from:

When any stop signal (SIGSTOP, SIGTSTP, SIGTTIN, SIGTTOU) is generated for a process, any pending SIGCONT signals for that process shall be

discarded. Conversely, when SIGCONT is generated for a process, all pending stop signals for that process shall be discarded. When SIGCONT is generated for a process that is stopped, the process shall be continued, even if the SIGCONT signal is blocked or ignored. If SIGCONT is blocked and not ignored, it shall remain pending until it is either unblocked or a stop signal is generated for the process.

to:

When any stop signal (SIGSTOP, SIGTSTP, SIGTTIN, SIGTTOU) is generated for a process or thread, all pending SIGCONT signals for that process or any of the threads within that process shall be discarded. Conversely, when SIGCONT is generated for a process or thread, all pending stop signals for that process or any of the threads within that process shall be discarded. When SIGCONT is generated for a process that is stopped, the process shall be continued, even if the SIGCONT signal is ignored by the process or is blocked by all threads within the process and there are no threads in a call to a sigwait() function selecting SIGCONT. If SIGCONT is blocked by all threads within the process, there are no threads in a call to a sigwait() function selecting SIGCONT, and SIGCONT is not ignored by the process, the SIGCONT signal shall remain pending on the process until it is either unblocked by a thread or a thread calls a sigwait() function selecting SIGCONT, or a stop signal is generated for the process or any of the threads within the process.

Rationale: Austin Group Defect Report(s) applied: 77.
See http://austingroupbugs.net/view.php?id=77

Change Number: XSH/TC1/D5/0005 [66]

On Page: 489 Line: 16717 Section: 2.4.3 Signal Actions

In the section "Pointer to a Function", change from:

When signal-catching functions are invoked asynchronously with process execution, the behavior of some of the functions defined by this volume of POSIX.1-200x is unspecified if they are called from a signal-catching function.

to:

If the process is multi-threaded, or if the process is single threaded and a signal handler is executed other than as the result of:

* The process calling abort(), raise(), kill(), pthread_kill(), or sigqueue() to generate a signal that is not blocked

* A pending signal being unblocked and being delivered before the call that unblocked it returns

the behavior is undefined if the signal handler refers to any object other than errno with static storage duration other than by assigning a value to an object declared as volatile sig_atomic_t, or if the signal handler calls any function defined in this standard other than one of the functions listed in the following table.

Rationale: Austin Group Defect Report(s) applied: 66.
See http://austingroupbugs.net/view.php?id=66

Change Number: XSH/TC1/D5/0006 [102]

On Page: 489 Line: 16722 Section: 2.4.3 Signal Actions

Add pthread_self() and pthread_kill() to the list of async-signal-safe functions.

Rationale: Austin Group Defect Report(s) applied: 102.
See http://austingroupbugs.net/view.php?id=102

Change Number: XSH/TC1/D5/0007 [103]

On Page: 489 Line: 16722 Section: 2.4.3 Signal Actions

Add pthread_sigmask() to the list of async-signal-safe functions.

Rationale: Austin Group Defect Report(s) applied: 103.
See http://austingroupbugs.net/view.php?id=103

Change Number: XSH/TC1/D5/0008 [234]

On Page: 489 Line: 16750 Section: 2.4.3 Signal Actions

In the table of functions that shall be async-signal-safe, add fchdir() to the list.

Rationale: Austin Group Defect Report(s) applied: 234.
See http://austingroupbugs.net/view.php?id=234

Change Number: XSH/TC1/D5/0009 [312]

On Page: 492 Line: 16878 Section: 2.5.1 Interaction of File Descriptors and Standard I/O Streams

Delete the text:

Otherwise, the result is undefined.

Rationale: Austin Group Defect Report(s) applied: 312.
See http://austingroupbugs.net/view.php?id=312
Since L16842-16843 cover both handles, it is clear that the statements on L16878 and L16893 are redundant (as well as contradictory).

Change Number: XSH/TC1/D5/0010 [312]

On Page: 493 Line: 16893 Section: 2.5.1 Interaction of File Descriptors and Standard I/O Streams

Delete the text:

If the rules above are not followed, the result is unspecified.

Rationale: Austin Group Defect Report(s) applied: 312.
See http://austingroupbugs.net/view.php?id=312
Since L16842-16843 cover both handles, it is clear that the

statements on L16878 and L16893 are redundant (as well as contradictory).

Change Number: XSH/TC1/D5/0011 [345,428]

On Page: 497 Line: 17067 Section: 2.7.1 IPC General Description

Add a new paragraph:

In addition to the ipc_perm structure, each associated data structure includes several time_t fields for recording timestamps of particular operations. When an operation is described as setting a timestamp to the current time, that particular timestamp member of the associated data structure shall be set to the largest time_t value which is not greater than the current time.

Rationale: Austin Group Defect Report(s) applied: 345,428.
See http://austingroupbugs.net/view.php?id=345
See http://austingroupbugs.net/view.php?id=428

Change Number: XSH/TC1/D5/0012 [109,429]

On Page: 507 Line: 17490 Section: 2.9.1 Thread-Safety

Add mblen() and mbtowc() in alphabetic order to the list of functions that need not be thread-safe in the table on P507 L17490-17510; remove wcstombs() from the same table.

On L17511, change from:

The wcrtomb() and wcsrtombs() functions need not be thread-safe if passed a NULL ps argument.

to:

The mbrlen(), mbrtowc(), mbsnrtowcs(), mbsrtowcs(), wcrtomb(), wcsnrtombs(), and wcsrtombs() functions need not be thread-safe if passed a NULL ps argument.

Rationale: Austin Group Defect Report(s) applied: 109,429.
See http://austingroupbugs.net/view.php?id=109
See http://austingroupbugs.net/view.php?id=429

Change Number: XSH/TC1/D5/0013 [89]

On Page: 508 Line: 17536-17542 Section: 2.9.3 Thread Mutexes

On L17536-L17538, change from:

* It returns successfully from [...] with m as the mutex argument.

to:

* It calls [...] with m as the mutex argument and the call returns zero or EOWNERDEAD.

On L17538 (after the 3rd bullet item) add a new bullet item:

* It calls pthread_mutex_setprioceiling() with m as the mutex argument and the call returns [EOWNERDEAD].

On L17539-17542 (last 2 bullet items), change from:

* It returns (successfully or not) from pthread_cond_wait() with m as the mutex argument (except as explicitly indicated otherwise for certain errors).

* It returns (successfully or not) from pthread_cond_timedwait() with m as the mutex argument (except as explicitly indicated otherwise for certain errors).

to:

* It calls pthread_cond_wait() with m as the mutex argument and the call returns zero or certain error numbers (see pthread_cond_timedwait()).

* It calls pthread_cond_timedwait() with m as the mutex argument and the call returns zero or certain error numbers (see pthread_cond_timedwait()).

Rationale: Austin Group Defect Report(s) applied: 89.
See http://austingroupbugs.net/view.php?id=89

Change Number: XSH/TC1/D5/0014 [410]

On Page: 514 Line: 17760 Section: 2.9.5.2 Cancellation Points

Add strerror_l() to the list, between strerror() and strerror_r().

Add strftime_l() to the list, between strftime() and symlink().

Rationale: Austin Group Defect Report(s) applied: 410.
See http://austingroupbugs.net/view.php?id=410

Change Number: XSH/TC1/D5/0015 [376]

On Page: 540 Line: 18820 Section: 2.12.1 Defined Types

In the table, change from:

dev_t Arithmetic type used for device numbers.

to:

dev_t Integer type used for device numbers.

Rationale: Austin Group Defect Report(s) applied: 376.
See http://austingroupbugs.net/view.php?id=376

Change Number: XSH/TC1/D5/0016 [327]

On Page: 541 Line: 18867 Section: 2.12.1 Defined Types

Change from:

time_t Integer or real-floating type used for time in seconds, as defined

in the ISO C standard.

to:

time_t Integer type used for time in seconds, as defined in the ISO C standard.

Rationale: Austin Group Defect Report(s) applied: 327.
See http://austingroupbugs.net/view.php?id=327

Change Number: XSH/TC1/D5/0017 [74]

On Page: 541 Line: 18881-18886 Section: 2.12.3 Pointer Types

Delete Section 2.12.3:

2.12.3 Pointer Types

All function pointer types shall have the same representation as the type pointer to void. Conversion of a function pointer to void * shall not alter the representation. A void * value resulting from such a conversion can be converted back to the original function pointer type, using an explicit cast, without loss of information.

Note: The ISO C standard does not require this, but it is required for POSIX conformance.

Rationale: Austin Group Defect Report(s) applied: 74.
See http://austingroupbugs.net/view.php?id=74

Change Number: XSH/TC1/D5/0018 [464]

On Page: 559 Line: 19380 Section: accept()

In the DESCRIPTION section, change from:

address_len Points to a socklen_t structure ...

to:

address_len Either a null pointer, if address is a null pointer, or a pointer to a socklen_t object ...

Rationale: Austin Group Defect Report(s) applied: 464.
See http://austingroupbugs.net/view.php?id=464

Change Number: XSH/TC1/D5/0019 [461]

On Page: 561 Line: 19464 Section: access()

In the DESCRIPTION section, change from:

... the current working directory is used ...

to:

... the current working directory shall be used ...

Rationale: Austin Group Defect Report(s) applied: 461.
See http://austingroupbugs.net/view.php?id=461

Change Number: XSH/TC1/D5/0020 [324]

On Page: 561 Line: 19482 Section: access()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of the path prefix is not a directory, ...

to:

A component of the path prefix names an existing file that is neither a directory nor a symbolic link to a directory, ...

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0021 [278]

On Page: 562 Line: 19491 access()

In the ERRORS section, add (after the [EBADF] error):

[ENOTDIR] The path argument is not an absolute path and fd is a file descriptor associated with a non-directory file.

Rationale: Austin Group Defect Report(s) applied: 278.
See http://austingroupbugs.net/view.php?id=278

Change Number: XSH/TC1/D5/0022 [278]

On Page: 562 Line: 19504 Section: access()

In the ERRORS section, delete:

[ENOTDIR] The path argument is not an absolute path and fd is neither AT_FDCWD nor a file descriptor associated with a directory.

Rationale: Austin Group Defect Report(s) applied: 278.
See http://austingroupbugs.net/view.php?id=278

Change Number: XSH/TC1/D5/0023 [291]

On Page: 562 Line: 19512-19513 Section: access()

In the EXAMPLES section, change both instances of "filename" to "pathname".

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XSH/TC1/D5/0024 [320]

On Page: 564 Line: 19589 Section: acos()

Change:

... and either a NaN (if supported), or an implementation-defined value shall be returned.

to:

... and a NaN shall be returned.

Rationale: Austin Group Defect Report(s) applied: 320.
See http://austingroupbugs.net/view.php?id=320

Change Number: XSH/TC1/D5/0025 [320]

On Page: 566 Line: 19645 Section: acosh()

In the RETURN VALUE section, change from:

... and either a NaN (if supported), or an implementation-defined value shall be returned.

to:

... and a NaN shall be returned.

Rationale: Austin Group Defect Report(s) applied: 320.
See http://austingroupbugs.net/view.php?id=320

Change Number: XSH/TC1/D5/0026 [98]

On Page: 573 Line: 19785 Section: aio_fsync()

In the SYNOPSIS section, shade the synopsis and add the margin code FSC|SIO.

On L19788, in the DESCRIPTION section, change from:

The aio_fsync() function shall asynchronously force all I/O operations associated with the file indicated by the file descriptor aio_fildes member of the aiocb structure referenced by the aiocbp argument and queued at the time of the call to aio_fsync() to the synchronized I/O completion state.

to:

The aio_fsync() function shall asynchronously perform a file synchronization operation, as specified by the op argument, for I/O operations associated with the file indicated by the file descriptor aio_fildes member of the aiocb structure referenced by the aiocbp argument and queued at the time of the call to aio_fsync().

On L19793, in the DESCRIPTION section, add SIO shading to:

If op is O_DSYNC, all currently queued I/O operations shall be completed as if by a call to fdatasync(); that is, as defined for synchronized I/O data integrity completion.

On L19794, in the DESCRIPTION section, add FSC shading to:

If op is O_SYNC, all currently queued I/O operations shall be completed as if by a call to fsync();

and add FSC SIO shading to:

... that is, as defined for synchronized I/O file integrity completion.

On L19797, in the DESCRIPTION section, change from:

... operation queued by aio_fsync() fails, then, as for fsync() and fdatasync(), outstanding I/O operations are not guaranteed to have been completed.

to:

... operation queued by aio_fsync() fails, then outstanding I/O operations are not guaranteed to have been completed.

On L19825, in the ERRORS section, add SIO shading to:

[EINVAL] This implementation does not support synchronized I/O for this file.

and add another [EINVAL] error after it, shaded FSC:

[EINVAL] The aio_fildes member of the aiocb structure refers to a file on which an fsync() operation is not possible.

Rationale: Austin Group Defect Report(s) applied: 98.
See http://austingroupbugs.net/view.php?id=98

Change Number: XSH/TC1/D5/0027 [98]

On Page: 574 Line: 19826 Section: aio_fsync()

In the ERRORS section, change from:

[EINVAL] A value of op other than O_DSYNC or O_SYNC was specified.

to:

[EINVAL] A value of op other than O_DSYNC or O_SYNC was specified, or O_DSYNC was specified and the implementation does not provide runtime support for the Synchronized Input and Output option, or O_SYNC was specified and the implementation does not provide runtime support for the File Synchronization option.

On L19829, in the ERRORS section, change from:

The error is returned in the error status for the asynchronous fsync() operation.

to:

The error is returned in the error status for the asynchronous operation.

Rationale: Austin Group Defect Report(s) applied: 98.
See http://austingroupbugs.net/view.php?id=98

Change Number: XSH/TC1/D5/0028 [317]

On Page: 582 Line: 20076 Section: aio_write()

Change:

If O_APPEND is set for the file descriptor, write operations append to the file in the same order as the calls were made.

to:

If O_APPEND is set for the file descriptor, or if aio_fildes is associated with a device that is incapable of seeking, write operations append to the file in the same order as the calls were made, except under circumstances described in Section 2.8.2.

Rationale: Austin Group Defect Report(s) applied: 317.
See http://austingroupbugs.net/view.php?id=317

Change Number: XSH/TC1/D5/0029 [324]

On Page: 588 Line: 20264 Section: alphasort()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of dir is not a directory.

to:

A component of dir names an existing file that is neither a directory nor a symbolic link to a directory.

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0030 [404]

On Page: 588 Line: 20264 Section: alphasort()

In the ERRORS section, add to the end of the "shall fail" errors for scandir():

[EOVERFLOW] One of the values to be returned or passed to a callback function cannot be represented correctly.

Rationale: Austin Group Defect Report(s) applied: 404.
See http://austingroupbugs.net/view.php?id=404

Change Number: XSH/TC1/D5/0031 [393]

On Page: 588 Line: 20268 Section: alphasort()

In the ERRORS section, [EMFILE] error, change from:

[EMFILE] {OPEN_MAX} file descriptors are currently open in the calling process.

to:

[EMFILE] All file descriptors available to the process are currently open.

Rationale: Austin Group Defect Report(s) applied: 393.
See http://austingroupbugs.net/view.php?id=393

Change Number: XSH/TC1/D5/0032 [291]

On Page: 588 Line: 20294 Section: alphasort()

In the APPLICATION USAGE section, change from:

If dir contains filenames that contain characters outside the domain of the collating sequence of the current locale, the alphasort() function need not provide a total ordering.

to:

If dir contains filenames that do not form character strings, or which contain characters outside the domain of the collating sequence of the current locale, the alphasort() function need not provide a total ordering. This condition is not possible if all filenames within the directory consist only of characters from the portable filename character set.

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XSH/TC1/D5/0033 [86,429]

On Page: 591 Line: 20382 Section: asctime()

In the RATIONALE section, change from:

The standard developers decided to mark the asctime() and asctime_r() functions obsolescent even though they are in ...

to:

The standard developers decided to mark the asctime() and asctime_r() functions obsolescent even though asctime() is in ...

Rationale: Austin Group Defect Report(s) applied: 86,429.
See http://austingroupbugs.net/view.php?id=86
See http://austingroupbugs.net/view.php?id=429

Change Number: XSH/TC1/D5/0034 [320]

On Page: 593 Line: 20436 Section: asin()

In the RETURN VALUE section, change from:

... and either a NaN (if supported), or an implementation-defined value shall be returned.

to:

... and a NaN shall be returned.

Rationale: Austin Group Defect Report(s) applied: 320.
See http://austingroupbugs.net/view.php?id=320

Change Number: XSH/TC1/D5/0035 [68]

On Page: 593 Line: 20438 Section: asin()

In the RETURN VALUE section, change from:

[MX]If x is subnormal, a range error may occur and x should be returned.[/MX]

to:

[MX]If x is subnormal, a range error may occur[/MX] [MXX]and x should be returned.[/MXX]

[MX]If x is not returned, asin(), asinf(), and asinl() shall return an implementation-defined value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.[/MX]

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0036 [68]

On Page: 595 Line: 20496 Section: asinh()

In the RETURN VALUE section, change from:

[MX]If x is subnormal, a range error may occur and x should be returned.[/MX]

to:

[MX]If x is subnormal, a range error may occur[/MX] [MXX]and x should be returned.[/MXX]

[MX]If x is not returned, asinh(), asinhf(), and asinhl() shall return an implementation-defined value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.[/MX]

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0037 [68]

On Page: 599 Line: 20598 Section: atan()

In the RETURN VALUE section, change from:

[MX]If x is subnormal, a range error may occur and x should be returned.[/MX]

to:

[MX]If x is subnormal, a range error may occur[/MX] [MXX]and x should be returned.[/MXX]

[MX]If x is not returned, atan(), atanf(), and atanl() shall return an implementation-defined value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.[/MX]

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0038 [68,428]

On Page: 601 Line: 20655 Section: atan2()

In the RETURN VALUE section, change from:

[MX]If the result underflows, a range error may occur and y/x should be returned.[/MX]

to:

If the correct value would cause underflow, a range error may occur, and atan2(), atan2f(), and atan2l() shall return an implementation-defined value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively. [MXX]If IEC 60559 Floating-Point is supported, y/x should be returned.[/MXX]

Rationale: Austin Group Defect Report(s) applied: 68,428.
See http://austingroupbugs.net/view.php?id=68
See http://austingroupbugs.net/view.php?id=428

Change Number: XSH/TC1/D5/0039 [320]

On Page: 605 Line: 20745 Section: atanh()

In the RETURN VALUE section, change from:

... and either a NaN (if supported), or an implementation-defined value shall be returned.

to:

... and a NaN shall be returned.

Rationale: Austin Group Defect Report(s) applied: 320.
See http://austingroupbugs.net/view.php?id=320

Change Number: XSH/TC1/D5/0040 [68]

On Page: 605 Line: 20747 Section: atanh()

In the RETURN VALUE section, change from:

[MX]If x is subnormal, a range error may occur and x should be returned.[/MX]

to:

[MX]If x is subnormal, a range error may occur[/MX] [MXX]and x should be returned.[/MXX]

[MX]If x is not returned, atanh(), atanhf(), and atanhl() shall return an implementation-defined value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.[/MX]

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0041 [75]

On Page: 614 Line: 20979 Section: basename()

In the DESCRIPTION section, change from:

The basename() function may modify the string pointed to by path, and may return a pointer to static storage that may then be overwritten by a subsequent call to basename().

to:

The basename() function may modify the string pointed to by path, and may return a pointer to internal storage. The returned pointer might be invalidated or the storage might be overwritten by a subsequent call to basename().

Rationale: Austin Group Defect Report(s) applied: 75.
See http://austingroupbugs.net/view.php?id=75

Change Number: XSH/TC1/D5/0042 [146]

On Page: 616 Line: 21046 Section: bind()

In the DESCRIPTION section, add a new paragraph after the existing third paragraph:

If the address family of the socket is AF_UNIX and the pathname in address names a symbolic link, bind() shall fail and set errno to [EADDRINUSE].

Rationale: Austin Group Defect Report(s) applied: 146.
See http://austingroupbugs.net/view.php?id=146

Change Number: XSH/TC1/D5/0043 [146]

On Page: 617 Line: 21086 Section: bind()

In the ERRORS section, change from:

[ENOENT] A component of the pathname does not name an existing file or the pathname is an empty string.

to:

[ENOENT] A component of the path prefix of the pathname in address does not name an existing file or the pathname is an empty string.

[ENOENT] or [ENOTDIR] The pathname in address contains at least one non-<slash> character and ends with one or more trailing <slash> characters. If the pathname names an existing file, an [ENOENT] error shall not occur.

Rationale: Austin Group Defect Report(s) applied: 146.
See http://austingroupbugs.net/view.php?id=146

Change Number: XSH/TC1/D5/0044 [324]

On Page: 617 Line: 21088 Section: bind()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of the path prefix of the pathname in address is not a directory, ...

to:

A component of the path prefix of the pathname in address names an existing file that is neither a directory nor a symbolic link to a directory, ...

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0045 [324]

On Page: 640 Line: 21719 Section: catopen()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of the path prefix of the message catalog is not a directory, ...

to:

A component of the path prefix of the message catalog names an existing file that is neither a directory nor a symbolic link to a directory, ...

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0046 [346]

On Page: 645 Line: 21865 Section: ceil()

For ceil(), delete P645 L21865-21868 (DESCRIPTION), L21874-21875 (RETURN VALUE), and replace L21877-21882 (ERRORS) with:

No errors are defined.

At L21870 (RETURN VALUE), add:

[MX]The result shall have the same sign as x.[/MX]

At L21886 (APPLICATION USAGE), change "an int or long" to "an intmax_t", and replace L21889-21892 (APPLICATION USAGE) with:

These functions may raise the inexact floating-point exception if the result differs in value from the argument.

Rationale: Austin Group Defect Report(s) applied: 346.
See http://austingroupbugs.net/view.php?id=346

Change Number: XSH/TC1/D5/0047 [324]

On Page: 653 Line: 22116 Section: chdir()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of the pathname is not a directory; ...

to:

A component of the pathname names an existing file that is neither a directory nor a symbolic link to a directory; ...

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0048 [300]

On Page: 655 Line: 22172-22173 Section: chmod()

Delete the paragraph at L22172-22173 from the DESCRIPTION section of chmod().

Rationale: Austin Group Defect Report(s) applied: 300.
See http://austingroupbugs.net/view.php?id=300

Change Number: XSH/TC1/D5/0049 [461]

On Page: 655 Line: 22186 Section: chmod()

In the DESCRIPTION section, change from:

... the current working directory is used ...

to:

... the current working directory shall be used ...

Rationale: Austin Group Defect Report(s) applied: 461.
See http://austingroupbugs.net/view.php?id=461

Change Number: XSH/TC1/D5/0050 [324]

On Page: 656 Line: 22199 Section: chmod()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of the path prefix is not a directory, ...

to:

A component of the path prefix names an existing file that is neither a directory nor a symbolic link to a directory, ...

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0051 [278]

On Page: 656 Line: 22210 chmod()

In the ERRORS section, add (after the [EBADF] error):

[ENOTDIR] The path argument is not an absolute path and fd is a file descriptor associated with a non-directory file.

Rationale: Austin Group Defect Report(s) applied: 278.
See http://austingroupbugs.net/view.php?id=278

Change Number: XSH/TC1/D5/0052 [278]

On Page: 656 Line: 22222 Section: chmod()

In the ERRORS section, delete:

[ENOTDIR] The path argument is not an absolute path and fd is neither AT_FDCWD nor a file descriptor associated with a directory.

Rationale: Austin Group Defect Report(s) applied: 278.
See http://austingroupbugs.net/view.php?id=278

Change Number: XSH/TC1/D5/0053 [461]

On Page: 659 Line: 22353 Section: chown()

In the DESCRIPTION section, change from:

... the current working directory is used ...

to:

... the current working directory shall be used ...

Rationale: Austin Group Defect Report(s) applied: 461.
See http://austingroupbugs.net/view.php?id=461

Change Number: XSH/TC1/D5/0054 [324]

On Page: 660 Line: 22367 Section: chown()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of the path prefix is not a directory, ...

to:

A component of the path prefix names an existing file that is neither a directory nor a symbolic link to a directory, ...

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0055 [278]

On Page: 660 Line: 22379 chown()

In the ERRORS section, add (after the [EBADF] error):

[ENOTDIR] The path argument is not an absolute path and fd is a file descriptor associated with a non-directory file.

Rationale: Austin Group Defect Report(s) applied: 278.
See http://austingroupbugs.net/view.php?id=278

Change Number: XSH/TC1/D5/0056 [278]

On Page: 661 Line: 22393 Section: chown()

In the ERRORS section, delete:

[ENOTDIR] The path argument is not an absolute path and fd is neither AT_FDCWD nor a file descriptor associated with a directory.

Rationale: Austin Group Defect Report(s) applied: 278.
See http://austingroupbugs.net/view.php?id=278

Change Number: XSH/TC1/D5/0057 [401]

On Page: 664 Line: 22502 Section: clearerr()

In the DESCRIPTION section, add to the end of the section:

[CX]The clearerr() function shall not change the setting of errno if stream is valid.[/CX]

Rationale: Austin Group Defect Report(s) applied: 401.
See http://austingroupbugs.net/view.php?id=401

Change Number: XSH/TC1/D5/0058 [106]

On Page: 668 Line: 22669 Section: clock_gettime()

In the ERRORS section, add after the existing [EINVAL] error:

The clock_gettime() function shall fail if:

[EOVERFLOW] The number of seconds will not fit in an object of type time_t.

Rationale: Austin Group Defect Report(s) applied: 106.
See http://austingroupbugs.net/view.php?id=106

Change Number: XSH/TC1/D5/0059 [419]

On Page: 677 Line: 22921 Section: close()

In the ERRORS section, change from:

[EBADF] The fildes argument is not a valid file descriptor.

to:

[EBADF] The fildes argument is not an open file descriptor.

Rationale: Austin Group Defect Report(s) applied: 419.
See http://austingroupbugs.net/view.php?id=419

Change Number: XSH/TC1/D5/0060 [149]

On Page: 678 Line: 22962 Section: close()

In the APPLICATION USAGE section, add a new paragraph at the end of the section:

Implementations may use file descriptors that must be inherited into child processes for the child process to remain conforming, such as for message catalog or tracing purposes. Therefore, an application that calls close() on an arbitrary integer risks non-conforming behavior, and close() can only portably be used on file descriptor values that the application has obtained through explicit actions, as well as the three file descriptors corresponding to the standard file streams. In multi-threaded parent applications, the practice of calling close() in a loop after fork() and before an exec call in order to avoid a race condition of leaking an unintended file descriptor into a child process, is therefore unsafe, and the race should instead be combatted by opening all file descriptors with the FD_CLOEXEC bit set unless the file descriptor is intended to

be inherited across exec.

Rationale: Austin Group Defect Report(s) applied: 149.
See http://austingroupbugs.net/view.php?id=149

Change Number: XSH/TC1/D5/0061 [149]

On Page: 678 Line: 22968 Section: close()

In the RATIONALE section, add a new paragraph at the end of the section:

The standard developers rejected a proposal to add closefrom() to the standard. Because the standard permits implementations to use inherited file descriptors as a means of providing a conforming environment for the child process, it is not possible to standardize an interface that closes arbitrary file descriptors above a certain value while still guaranteeing a conforming environment.

Rationale: Austin Group Defect Report(s) applied: 149.
See http://austingroupbugs.net/view.php?id=149

Change Number: XSH/TC1/D5/0062 [324]

On Page: 691 Line: 23411 Section: connect()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of the path prefix of the pathname in address is not a directory, ...

to:

A component of the path prefix of the pathname in address names an existing file that is neither a directory nor a symbolic link to a directory, ...

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0063 [320]

On Page: 694 Line: 23510 Section: cos()

In the RETURN VALUE section, change from:

... and either a NaN (if supported), or an implementation-defined value shall be returned.

to:

... and a NaN shall be returned.

Rationale: Austin Group Defect Report(s) applied: 320.
See http://austingroupbugs.net/view.php?id=320

Change Number: XSH/TC1/D5/0064 [291]

On Page: 702 Line: 23739-23741 Section: creat()

In the EXAMPLES section, change both instances of "filename" to "pathname".

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XSH/TC1/D5/0065 [75,428]

On Page: 713 Line: 24015 Section: ctermid()

In the RETURN VALUE section, change from:

If s is a null pointer, the string shall be generated in an area that may be static (and therefore may be overwritten by each call), the address of which shall be returned. Otherwise, ...

to:

If s is a null pointer, the string shall be generated in a area that may be static, the address of which shall be returned. The application shall not modify the string returned. The returned pointer might be invalidated or the string content might be overwritten by a subsequent call to ctermid(). If s is not a null pointer, ...

Rationale: Austin Group Defect Report(s) applied: 75,428.
See http://austingroupbugs.net/view.php?id=75
See http://austingroupbugs.net/view.php?id=428

Change Number: XSH/TC1/D5/0066 [321,428]

On Page: 715 Line: 24085 Section: ctime()

In the DESCRIPTION section, change from:

Unlike ctime(), the thread-safe version of ctime_r() is not required to set tzname.

to:

Unlike ctime(), the ctime_r() function is not required to set tzname. If ctime_r() does not set tzname, it shall not set daylight and shall not set timezone.

Rationale: Austin Group Defect Report(s) applied: 321,428.
See http://austingroupbugs.net/view.php?id=321
See http://austingroupbugs.net/view.php?id=428

Change Number: XSH/TC1/D5/0067 [422]

On Page: 723 Line: 24328 Section: dirfd()

In the DESCRIPTION section, change from:

If any attempt is made to close the file descriptor, or to modify the state of the associated description, other than by means of closedir(), readdir(), readdir_r(), or rewinddir(), the behavior is undefined.

to:

If any attempt is made to close the file descriptor, or to modify the state of the associated description, other than by means of closedir(), readdir(), readdir_r(), rewinddir(), or [XSI]seekdir()[/XSI], the behavior is undefined.

Rationale: Austin Group Defect Report(s) applied: 422.
See http://austingroupbugs.net/view.php?id=422

Change Number: XSH/TC1/D5/0068 [75]

On Page: 725 Line: 24380 Section: dirname()

In the RETURN VALUE section, change from:

The dirname() function may modify the string pointed to by path, and may return a pointer to static storage that may then be overwritten by subsequent calls to dirname().

to:

The dirname() function may modify the string pointed to by path, and may return a pointer to internal storage. The returned pointer might be invalidated or the storage might be overwritten by a subsequent call to dirname().

Rationale: Austin Group Defect Report(s) applied: 75.
See http://austingroupbugs.net/view.php?id=75

Change Number: XSH/TC1/D5/0069 [74]

On Page: 728 Line: 24469 Section: dlclose()

Replace the NAME, SYNOPSIS, DESCRIPTION, RETURN VALUE, ERRORS, EXAMPLES, and APPLICATION USAGE sections of the dlclose() description on L24469-24513 with:

NAME

dlclose -- close a symbol table handle

SYNOPSIS

```
#include <dlfcn.h>
int dlclose(void *handle);
```

DESCRIPTION

The dlclose() function shall inform the system that the symbol table handle specified by handle is no longer needed by the application.

An application writer may use dlclose() to make a statement of intent on the part of the process, but this statement does not create any

requirement upon the implementation. When the symbol table handle is closed, the implementation may unload the executable object files that were loaded by dlopen() when the symbol table handle was opened and those that were loaded by dlsym() when using the symbol table handle identified by handle.

Once a symbol table handle has been closed, an application should assume that any symbols (function identifiers and data object identifiers) made visible using handle, are no longer available to the process.

Although a dlclose() operation is not required to remove any functions or data objects from the address space, neither is an implementation prohibited from doing so. The only restriction on such a removal is that no function nor data object shall be removed to which references have been relocated, until or unless all such references are removed. For instance, an executable object file that had been loaded with a dlopen() operation specifying the RTLD_GLOBAL flag might provide a target for dynamic relocations performed in the processing of other relocatable objects - in such environments, an application may assume that no relocation, once made, shall be undone or remade unless the executable object file containing the relocated object has itself been removed.

RETURN VALUE

If the referenced symbol table handle was successfully closed, dlclose() shall return 0. If handle does not refer to an open symbol table handle or if the symbol table handle could not be closed, dlclose() shall return a non-zero value. More detailed diagnostic information shall be available through dlerror().

ERRORS

No errors are defined.

EXAMPLES

The following example illustrates use of dlopen() and dlclose():

```
#include <dlfcn.h>
int eret;
void *mylib;
...
/* Open a dynamic library and then close it ... */
mylib = dlopen("mylib.so", RTLD_LOCAL | RTLD_LAZY);
...
eret = dlclose(mylib);
...
```

APPLICATION USAGE

A conforming application should employ a symbol table handle returned from a dlopen() invocation only within a given scope bracketed by a dlopen() operation and the corresponding dlclose() operation. Implementations are free to use reference counting or other techniques such that multiple calls to dlopen() referencing the same executable object file may return a pointer to the same data object as the symbol table handle.

Implementations are also free to re-use a handle. For these reasons, the value of a handle must be treated as an opaque data type by the application, used only in calls to dlsym() and dlclose().

Rationale: Austin Group Defect Report(s) applied: 74.
See http://austingroupbugs.net/view.php?id=74

Change Number: XSH/TC1/D5/0070 [75]

On Page: 730 Line: 24542 Section: dlerror()

In the RETURN VALUE section, add a new paragraph at the end of the section:

The application shall not modify the string returned. The returned pointer might be invalidated or the string content might be overwritten by a subsequent call to dlerror() in the same thread (if dlerror() is thread-safe) or in any thread (if dlerror() is not thread-safe).

On L24555, in the APPLICATION USAGE section, delete the following:

The messages returned by dlerror() may reside in a static buffer that is overwritten on each call to dlerror(). Application code should not write to this buffer. Programs wishing to preserve an error message should make their own copies of that message.

Rationale: Austin Group Defect Report(s) applied: 75.
See http://austingroupbugs.net/view.php?id=75

Change Number: XSH/TC1/D5/0071 [97]

On Page: 730 Line: 24539 Section: dlerror()

In the DESCRIPTION section, change from:

The dlerror() function need not be thread-safe.

to:

It is implementation-defined whether or not the dlerror() function is thread-safe. A thread-safe implementation shall return only errors that occur on the current thread.

Rationale: Austin Group Defect Report(s) applied: 97.
See http://austingroupbugs.net/view.php?id=97

Change Number: XSH/TC1/D5/0072 [133]

On Page: 730 Line: 24552 Section: dlerror()

In the EXAMPLES section, indent the printf() statement one tab stop.

Rationale: Austin Group Defect Report(s) applied: 133.
See http://austingroupbugs.net/view.php?id=133
The indentation in the EXAMPLE is improved.

Change Number: XSH/TC1/D5/0073 [74]

On Page: 732 Line: 24575 Section: dlopen()

Replace the NAME, SYNOPSIS, DESCRIPTION, and RETURN VALUE sections of

the dlopen() description on P732-733 L24575-24666:

NAME

dlopen -- open a symbol table handle

SYNOPSIS

```
#include <dlfcn.h>
void *dlopen(const char *file, int mode);
```

DESCRIPTION

The dlopen() function shall make the symbols (function identifiers and data object identifiers) in the executable object file specified by file available to the calling program.

The class of executable object files eligible for this operation and the manner of their construction are implementation-defined, though typically such files are shared libraries or programs.

Implementations may permit the construction of embedded dependencies in executable object files. In such cases, a dlopen() operation shall load those dependencies in addition to the executable object file specified by file. Implementations may also impose specific constraints on the construction of programs that can employ dlopen() and its related services.

A successful dlopen() shall return a symbol table handle which the caller may use on subsequent calls to dlsym() and dlclose().

The value of this symbol table handle should not be interpreted in any way by the caller.

The file argument is used to construct a pathname to the executable object file. If file contains a <slash> character, the file argument is used as the pathname for the file. Otherwise, file is used in an implementation-defined manner to yield a pathname.

If file is a null pointer, dlopen() shall return a global symbol table handle for the currently running process image. This symbol table handle shall provide access to the symbols from an ordered set of executable object files consisting of the original program image file, any executable object files loaded at program start-up as specified by that process image file (for example, shared libraries), and the set of executable object files loaded using dlopen() operations with the RTLD_GLOBAL flag. As the latter set of executable object files can change during execution, the set of symbols made available by this symbol table handle can also change dynamically.

Only a single copy of an executable object file shall be brought into the address space, even if dlopen() is invoked multiple times in reference to the executable object file, and even if different pathnames are used to reference the executable object file.

The mode parameter describes how dlopen() shall operate upon file with respect to the processing of relocations and the scope of visibility of the symbols provided within file. When an executable object file is brought into the address space of a process, it may contain references to symbols whose addresses are not known until the executable object file is loaded.

These references shall be relocated before the symbols can be

accessed. The mode parameter governs when these relocations take place and may have the following values:

RTLD_LAZY Relocations shall be performed at an implementation-defined time, ranging from the time of the dlopen() call until the first reference to a given symbol occurs. Specifying RTLD_LAZY should improve performance on implementations supporting dynamic symbol binding since a process might not reference all of the symbols in an executable object file. And, for systems supporting dynamic symbol resolution for normal process execution, this behavior mimics the normal handling of process execution.

RTLD_NOW All necessary relocations shall be performed when the executable object file is first loaded. This may waste some processing if relocations are performed for symbols that are never referenced. This behavior may be useful for applications that need to know that all symbols referenced during execution will be available before dlopen() returns.

Any executable object file loaded by dlopen() that requires relocations against global symbols can reference the symbols in the original process image file, any executable object files loaded at program start-up, from the initial process image itself, from any other executable object file included in the same dlopen() invocation, and any executable object files that were loaded in any dlopen() invocation and which specified the RTLD_GLOBAL flag. To determine the scope of visibility for the symbols loaded with a dlopen() invocation, the mode parameter should be a bitwise-inclusive OR with one of the following values:

RTLD_GLOBAL The executable object file's symbols shall be made available for relocation processing of any other executable object file. In addition, symbol lookup using dlopen(NULL, mode) and an associated dlsym() allows executable object files loaded with this mode to be searched.

RTLD_LOCAL The executable object file's symbols shall not be made available for relocation processing of any other executable object file.

If neither RTLD_GLOBAL nor RTLD_LOCAL are specified, the default behavior is unspecified.

If an executable object file is specified in multiple dlopen() invocations, mode is interpreted at each invocation.

If RTLD_NOW has been specified, all relocations shall have been completed rendering further RTLD_NOW operations redundant and any further RTLD_LAZY operations irrelevant.

If RTLD_GLOBAL has been specified, the executable object file shall maintain the RTLD_GLOBAL status regardless of any previous or future specification of RTLD_LOCAL, as long as the executable object file remains in the address space (see dlclose()).

Symbols introduced into the process image through calls to dlopen() may be used in relocation activities. Symbols so introduced may duplicate symbols already defined by the program or previous dlopen() operations. To resolve the ambiguities such a situation might present, the resolution of a symbol reference to symbol definition is based on a symbol resolution order. Two such resolution orders are defined: load order and dependency order. Load order establishes an ordering among symbol definitions, such that the first definition loaded (including definitions from the process image file and any dependent executable object files loaded with it) has priority over executable object files added later (by dlopen()). Load ordering is used in relocation processing. Dependency ordering uses a breadth-first order starting with a given executable object file, then all of its dependencies, then any dependents of those,

iterating until all dependencies are satisfied. With the exception of the global symbol table handle obtained via a dlopen() operation with a null pointer as the file argument, dependency ordering is used by the dlsym() function. Load ordering is used in dlsym() operations upon the global symbol table handle.

When an executable object file is first made accessible via dlopen(), it and its dependent executable object files are added in dependency order. Once all the executable object files are added, relocations are performed using load order. Note that if an executable object file or its dependencies had been previously loaded, the load and dependency orders may yield different resolutions.

The symbols introduced by dlopen() operations and available through dlsym() are at a minimum those which are exported as identifiers of global scope by the executable object file. Typically such identifiers shall be those that were specified in (for example) C source code as having extern linkage. The precise manner in which an implementation constructs the set of exported symbols for an executable object file is implementation-defined.

RETURN VALUE

Upon successful completion, dlopen() shall return a symbol table handle. If file cannot be found, cannot be opened for reading, is not of an appropriate executable object file format for processing by dlopen(), or if an error occurs during the process of loading file or relocating its symbolic references, dlopen() shall return a null pointer. More detailed diagnostic information shall be available through dlerror().

Rationale: Austin Group Defect Report(s) applied: 74.
See http://austingroupbugs.net/view.php?id=74

Change Number: XSH/TC1/D5/0074 [74]

On Page: 735 Line: 24689 Section: dlsym()

Replace the NAME, SYNOPSIS, DESCRIPTION, RETURN VALUE, ERRORS, EXAMPLES, APPLICATION USAGE, and RATIONALE sections of the dlsym() description on P735-736 L24689-24748 with:

NAME

dlsym -- get the address of a symbol from a symbol table handle

SYNOPSIS

```
#include <dlfcn.h>
void *dlsym(void *restrict handle, const char *restrict name);
```

DESCRIPTION

The dlsym() function shall obtain the address of a symbol (a function identifier or a data object identifier) defined in the symbol table identified by the handle argument. The handle argument is a symbol table handle returned from a call to dlopen() (and which has not since been released by a call to dlclose()), and name is the symbol's name as a character string. The return value from dlsym(), cast to a pointer to the type of the named symbol, can be used to call (in the case of a function) or access the contents of (in the case of a data object) the named symbol.

The dlsym() function shall search for the named symbol in the symbol table referenced by handle. If the symbol table was created with lazy loading (see RTLD_LAZY in dlopen()), load ordering shall be used in dlsym() operations to relocate executable object files needed to resolve the symbol. The symbol resolution algorithm used shall be dependency order as described in dlopen().

The RTLD_DEFAULT and RTLD_NEXT symbolic constants (which may be defined in <dlfcn.h>) are reserved for future use as special values that applications may be allowed to use for handle.

RETURN VALUE

Upon successful completion, if name names a function identifier, dlsym() shall return the address of the function converted from type pointer to function to type pointer to void; otherwise, dlsym() shall return the address of the data object associated with the data object identifier named by name converted from a pointer to the type of the data object to a pointer to void. If handle does not refer to a valid symbol table handle or if the symbol named by name cannot be found in the symbol table associated with handle, dlsym() shall return a null pointer.

More detailed diagnostic information shall be available through dlerror().

ERRORS

No errors are defined.

EXAMPLES

The following example shows how dlopen() and dlsym() can be used to access either a function or a data object. For simplicity, error checking has been omitted.

```
void *handle;
int (*fptr)(int), *iptr, result;
/* open the needed symbol table */
handle = dlopen("/usr/home/me/libfoo.so", RTLD_LOCAL | RTLD_LAZY);
/* find the address of the function my_function */
fptr = (int (*)(int))dlsym(handle, "my_function");
/* find the address of the data object my_object */
iptr = (int *)dlsym(handle, "my_OBJ");
/* invoke my_function, passing the value of my_OBJ as the parameter */
result = (*fptr)(*iptr);
```

APPLICATION USAGE

The following special purpose values for handle are reserved for future use and have the indicated meanings:

RTLD_DEFAULT The identifier lookup happens in the normal global scope; that is, a search for a identifier using handle would find the same definition as a direct use of this identifier in the program code.

RTLD_NEXT Specifies the next executable object file after this one that defines name. This one refers to the executable object file containing the invocation of dlsym(). The next executable object file is the one found upon the application of a load order symbol resolution algorithm (see dlopen()). The next symbol is either one of global scope (because it was introduced as part of the original process image or because it was added with a dlopen() operation including the RTLD_GLOBAL flag), or is in an executable object file that was included in the same dlopen() operation that loaded this one.

The RTLD_NEXT flag is useful to navigate an intentionally created hierarchy of multiply-defined symbols created through interposition. For example, if a program wished to create an implementation of malloc() that embedded some statistics gathering about memory allocations, such an implementation could use the real malloc() definition to perform the memory allocation - and itself only embed the necessary logic to implement the statistics gathering function.

Note that conversion from a void * pointer to a function pointer as in:

```
fptr = (int (*)(int))dlsym(handle, "my_function");
```

is not defined by the ISO C Standard. This standard requires this conversion to work correctly on conforming implementations.

RATIONALE

None.

Rationale: Austin Group Defect Report(s) applied: 74.
See http://austingroupbugs.net/view.php?id=74

Change Number: XSH/TC1/D5/0075 [149,428]

On Page: 742 Line: 24917 Section: dup()

In the APPLICATION USAGE section, change from:

None.

to:

Implementations may use file descriptors that must be inherited into child processes for the child process to remain conforming, such as for message catalog or tracing purposes. Therefore, an application that calls dup2() with an arbitrary integer for fildes2 risks non-conforming behavior, and dup2() can only portably be used to overwrite file descriptor values that the application has obtained through explicit actions, or for the three file descriptors corresponding to the standard file streams. In order to avoid a race condition of leaking an unintended file descriptor into a child process, an application should consider opening all file descriptors with the FD_CLOEXEC bit set unless the file descriptor is intended to be inherited across exec.

Rationale: Austin Group Defect Report(s) applied: 149,428.
See http://austingroupbugs.net/view.php?id=149
See http://austingroupbugs.net/view.php?id=428

Change Number: XSH/TC1/D5/0076 [149]

On Page: 742 Line: 24915 Section: dup()

In the RATIONALE section, change from:

The dup() and dup2() functions are redundant. Their services are also provided by the fcntl() function. They have been included in this volume of POSIX.1-2008 primarily for historical reasons, since many existing applications use them.

to:

The dup() function is redundant. Its services are also provided by the fcntl() function. It has been included in this volume of POSIX.1-2008 primarily for historical reasons, since many existing applications use it. On the other hand, the dup2() function provides unique services, as no other interface is able to atomically replace an existing file descriptor.

Rationale: Austin Group Defect Report(s) applied: 149.
See http://austingroupbugs.net/view.php?id=149

Change Number: XSH/TC1/D5/0077 [283,301]

On Page: 743 Line: 24943 Section: duplocale()

In the DESCRIPTION section, add two new paragraphs to the end of the section:

If the locobj argument is LC_GLOBAL_LOCALE, duplocale() shall create a new locale object containing a copy of the global locale determined by the setlocale() function.

The behavior is undefined if the locobj argument is not a valid locale object handle.

Rationale: Austin Group Defect Report(s) applied: 283,301.
See http://austingroupbugs.net/view.php?id=283
See http://austingroupbugs.net/view.php?id=301

Change Number: XSH/TC1/D5/0078 [283]

On Page: 743 Line: 24951 Section: duplocale()

In the ERRORS section, delete:

The duplocale() function may fail if:

[EINVAL] locobj is not a handle for a locale object.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0079 [301]

On Page: 743 Line: 24978 Section: duplocale()

In the APPLICATION USAGE section, add a new paragraph to the end of the section:

The duplocale() function can also be used in conjunction with uselocale((locale_t)0). This returns the locale in effect for the calling thread, but can have the value LC_GLOBAL_LOCALE. Passing LC_GLOBAL_LOCALE to functions such as isalnum_l() results in undefined behavior, but applications can convert it into a usable locale object by using duplocale().

Rationale: Austin Group Defect Report(s) applied: 301.
See http://austingroupbugs.net/view.php?id=301

Change Number: XSH/TC1/D5/0080 [75]

On Page: 747 Line: 25061 Section: endgrent()

In the RETURN VALUE section, change from:

The return value may point to a static area which is overwritten by a subsequent call to getgrgid(), getgrnam(), or getgrent().

to:

The application shall not modify the structure to which the return value points, nor any storage areas pointed to by pointers within the structure. The returned pointer, and pointers within the structure, might be invalidated or the structure or the storage areas might be overwritten by a subsequent call to getgrgid(), getgrnam(), or getgrent().

Rationale: Austin Group Defect Report(s) applied: 75.
See http://austingroupbugs.net/view.php?id=75

Change Number: XSH/TC1/D5/0081 [75,428]

On Page: 749 Line: 25121 Section: endhostent()

In the RETURN VALUE section, add a new paragraph to the end of the section:

The application shall not modify the structure to which the return value points, nor any storage areas pointed to by pointers within the structure. The returned pointer, and pointers within the structure, might be invalidated or the structure or the storage areas might be overwritten by a subsequent call to gethostent().

Rationale: Austin Group Defect Report(s) applied: 75,428.
See http://austingroupbugs.net/view.php?id=75
See http://austingroupbugs.net/view.php?id=428

Change Number: XSH/TC1/D5/0082 [75]

On Page: 749 Line: 25127 Section: endhostent()

In the APPLICATION USAGE section, change from:

The gethostent() function may return pointers to static data, which may be overwritten by subsequent calls to any of these functions.

to:

None.

Rationale: Austin Group Defect Report(s) applied: 75.
See http://austingroupbugs.net/view.php?id=75

Change Number: XSH/TC1/D5/0083 [75]

On Page: 751 Line: 25175 Section: endnetent()

In the RETURN VALUE section, add a new paragraph to the end of the section:

The application shall not modify the structure to which the return value points, nor any storage areas pointed to by pointers within the structure. The returned pointer, and pointers within the structure, might be invalidated or the structure or the storage areas might be overwritten by a subsequent call to getnetbyaddr(), getnetbyname(), or getnetent().

Rationale: Austin Group Defect Report(s) applied: 75.
See http://austingroupbugs.net/view.php?id=75

Change Number: XSH/TC1/D5/0084 [75]

On Page: 752 Line: 25181 Section: endnetent()

In the APPLICATION USAGE section, change from:

The getnetbyaddr(), getnetbyname(), and getnetent() functions may return pointers to static data, which may be overwritten by subsequent calls to any of these functions.

to:

None.

Rationale: Austin Group Defect Report(s) applied: 75.
See http://austingroupbugs.net/view.php?id=75

Change Number: XSH/TC1/D5/0085 [75]

On Page: 753 Line: 25230 Section: endprotoent()

In the RETURN VALUE section, add a new paragraph to the end of the section:

The application shall not modify the structure to which the return value points, nor any storage areas pointed to by pointers within the structure. The returned pointer, and pointers within the structure, might be invalidated or the structure or the storage areas might be overwritten by a subsequent call to getprotobyname(), getprotobynumber(), or getprotoent().

Rationale: Austin Group Defect Report(s) applied: 75.
See http://austingroupbugs.net/view.php?id=75

Change Number: XSH/TC1/D5/0086 [75]

On Page: 754 Line: 25236 Section: endprotent()

In the APPLICATION USAGE section, change from:

The getprotobyname(), getprotobynumber(), and getprotoent() functions may return pointers to static data, which may be overwritten by subsequent calls to any of these functions.

to:

None.

Rationale: Austin Group Defect Report(s) applied: 75.
See http://austingroupbugs.net/view.php?id=75

Change Number: XSH/TC1/D5/0087 [75]

On Page: 755 Line: 25271 Section: endpwent()

In the RETURN VALUE section, add a new paragraph to the end of the section:

The application shall not modify the structure to which the return value points, nor any storage areas pointed to by pointers within the structure. The returned pointer, and pointers within the structure, might be invalidated or the structure or the storage areas might be overwritten by a subsequent call to getpwuid(), getpwnam(), or getpwent().

On L25278, delete from the ERRORS section:

The return value may point to a static area which is overwritten by a subsequent call to getpwuid(), getpwnam(), or getpwent().

Rationale: Austin Group Defect Report(s) applied: 75.
See http://austingroupbugs.net/view.php?id=75

Change Number: XSH/TC1/D5/0088 [75]

On Page: 758 Line: 25362 Section: endservent()

In the RETURN VALUE section, add a new paragraph to the end of the section:

The application shall not modify the structure to which the return value points, nor any storage areas pointed to by pointers within the structure. The returned pointer, and pointers within the structure, might be invalidated or the structure or the storage areas might be overwritten by a subsequent call to getservbyname(), getservbyport(), or getservent().

Rationale: Austin Group Defect Report(s) applied: 75.
See http://austingroupbugs.net/view.php?id=75

Change Number: XSH/TC1/D5/0089 [75]

On Page: 759 Line: 25370 Section: endservent()

In the APPLICATION USAGE section, delete the paragraph:

The getservbyname(), getservbyport(), and getservent() functions may return pointers to static data, which may be overwritten by subsequent

calls to any of these functions.

Rationale: Austin Group Defect Report(s) applied: 75.
See http://austingroupbugs.net/view.php?id=75

Change Number: XSH/TC1/D5/0090 [213,428]

On Page: 761 Line: 25428 Section: endutxent()

In the DESCRIPTION section, in the table, change from:

LOGIN_PROCESS ut_id, ut_user (implementation-defined name of the login process), ut_pid, ut_tv

to:

LOGIN_PROCESS ut_id, ut_user (implementation-defined name of the login process), ut_line, ut_pid, ut_tv

Rationale: Austin Group Defect Report(s) applied: 213,428.
See http://austingroupbugs.net/view.php?id=213
See http://austingroupbugs.net/view.php?id=428
ut_line is added before ut_pid on L25429.

Change Number: XSH/TC1/D5/0091 [213]

On Page: 761 Line: 25435 Section: endutxent()

In the DESCRIPTION section, change from:

If the process has appropriate privileges, the pututxline() function shall write out the structure into the user accounting database. It shall use getutxid() to search for a record that satisfies the request. If this search succeeds, then the entry shall be replaced. Otherwise, a new entry shall be made at the end of the user accounting database.

to:

If the process has appropriate privileges, the pututxline() function shall write out the structure into the user accounting database. It shall search for a record as if by getutxid() that satisfies the request. If this search succeeds, then the entry shall be replaced. Otherwise, a new entry shall be made at the end of the user accounting database.

Rationale: Austin Group Defect Report(s) applied: 213.
See http://austingroupbugs.net/view.php?id=213

Change Number: XSH/TC1/D5/0092 [68]

On Page: 765 Line: 25514 Section: erf()

In the RETURN VALUE section, change from:

[MX]If x is subnormal, a range error may occur, and 2 * x/sqrt(pi) should be returned.[/MX]

to:

If the correct value would cause underflow, a range error may occur, and erf(), erff(), and erfl() shall return an implementation-defined value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively. [MXX]If IEC 60559 Floating-Point is supported, 2 * x/sqrt(pi) should be returned.[/MXX]

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0093 [68]

On Page: 767 Line: 25578 Section: erfc()

In the RETURN VALUE section, change from:

If the correct value would cause underflow and is not representable, a range error may occur and [MX]either 0.0 (if representable), or[/MX] an implementation-defined value shall be returned.

to:

If the correct value would cause underflow, [MXX]and is not representable[/MXX], a range error may occur, and erfc(), erfcf(), and erfcl() shall return [MXX]0.0, or[/MXX] (if IEC 60559 Floating-Point is not supported) an implementation-defined value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0094 [68]

On Page: 767 Line: 25584 Section: erfc()

Change MX shading to MXX for:

If the correct value would cause underflow and is representable, a range error may occur and the correct value shall be returned.

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0095 [386]

On Page: 772 Line: 25708 Section: exec

In the DESCRIPTION section, change from:

In addition, the following variable:

```
extern char **environ;
```

to:

In addition, the following variable, which must be declared by the user if it is to be used directly:

```
extern char **environ;

Rationale: Austin Group Defect Report(s) applied: 386.
See http://austingroupbugs.net/view.php?id=386


Change Number: XSH/TC1/D5/0096 [167]


On Page: 772 Line: 25712 Section: exec

In the DESCRIPTION section, after L25712, add two new paragraphs:

Applications can change the entire environment in a single operation by
assigning the environ variable to point to an array of character pointers
to the new environment strings. After assigning a new value to environ,
applications should not rely on the new environment strings remaining
part of the environment, as a call to getenv(), [XSI]putenv(),[/XSI]
setenv(), unsetenv() or any function that is dependent on an environment
variable may, on noticing that environ has changed, copy the environment
strings to a new array and assign environ to point to it.

Any application that directly modifies the pointers to which the environ
variable points has undefined behavior.

Rationale: Austin Group Defect Report(s) applied: 167.
See http://austingroupbugs.net/view.php?id=167


Change Number: XSH/TC1/D5/0097 [291]


On Page: 773 Line: 25737,25741 Section: exec

In the DESCRIPTION section, change both instances of "filename" to
"filename string".

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291


Change Number: XSH/TC1/D5/0098 [173]


On Page: 773 Line: 25757 Section: exec

In the DESCRIPTION section, change from:

If file descriptors 0, 1, and 2 would otherwise be closed after a
successful call to one of the exec family of functions, and the new
process image file has the set-user-ID or set-group-ID file mode bits set,
and the ST_NOSUID bit is not set for the file system containing the new
process image file, implementations may open an unspecified file for
each of these file descriptors in the new process image.

to:

If file descriptor 0, 1, or 2 would otherwise be closed after a successful
call to one of the exec family of functions, implementations may open an
unspecified file for the file descriptor in the new process image. If
a standard utility or a conforming application is executed with file
descriptor 0 not open for reading or with file descriptor 1 or 2 not
```

3922 open for writing, the environment in which the utility or application
3923 is executed shall be deemed non-conforming, and consequently the utility
3924 or application might not behave as described in this standard.
3925
3926 *Rationale:* Austin Group Defect Report(s) applied: 173.
3927 See http://austingroupbugs.net/view.php?id=173
3928
3929
3930 **Change Number: XSH/TC1/D5/0099** [296]
3931
3932
3933 On Page: 777 Line: 25899 Section: exec
3934
3935 In the ERRORS section, [EACCES] error, change:
3936
3937 Search permission is denied for a directory listed in the new process
3938 image file's path prefix, or the new process image file denies execution
3939 permission, or the ...
3940
3941 to:
3942
3943 The ...
3944
3945 *Rationale:* Austin Group Defect Report(s) applied: 296.
3946 See http://austingroupbugs.net/view.php?id=296
3947
3948
3949 **Change Number: XSH/TC1/D5/0100** [324]
3950
3951
3952 On Page: 777 Line: 25912 Section: exec
3953
3954 In the ERRORS section, [ENOTDIR] error, change from:
3955
3956 A component of the new process image file's path prefix is not a
3957 directory, ...
3958
3959 to:
3960
3961 A component of the new process image file's path prefix names an existing
3962 file that is neither a directory nor a symbolic link to a directory, ...
3963
3964 *Rationale:* Austin Group Defect Report(s) applied: 324.
3965 See http://austingroupbugs.net/view.php?id=324
3966 This is an editorial issue clarifying the intent of the
3967 standard.
3968
3969
3970 **Change Number: XSH/TC1/D5/0101** [296]
3971
3972
3973 On Page: 777 Line: 25917-25931 Section: exec
3974
3975 In the ERRORS section, at L25917 before:
3976
3977 The exec functions, except for execlp() and execvp(), shall fail if:
3978
3979 insert:
3980
3981 The exec functions, except for fexecve(), shall fail if:
3982
3983 [EACCES] Search permission is denied for a directory listed in the new
3984 process image file's path prefix, or the new process image file denies

execution permission.

Move the following errors from the indicated line numbers to after this inserted [EACCES] error:

L25906 [ELOOP]
L25908 [ENAMETOOLONG]
L25910 [ENOENT]
L25912 [ENOTDIR]

At L25931 (after [ENOMEM]) add:

The exec functions, except for fexecve(), may fail if:

Move the following errors from the indicated line numbers to after this added line:

L25923 [ELOOP]
L25925 [ENAMETOOLONG]

(This leaves just ENOMEM under "The exec functions may fail if:".)

Rationale: Austin Group Defect Report(s) applied: 296.
See http://austingroupbugs.net/view.php?id=296

Change Number: XSH/TC1/D5/0102 [302]

On Page: 779 Line: 25987 Section: exec

In the APPLICATION USAGE section, change from:

Applications that require other than the default POSIX locale should call setlocale() with the appropriate parameters to establish the locale of the new process.

to:

Applications that require other than the default POSIX locale as the global locale in the new process image should call setlocale() with the appropriate parameters.

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when per-thread locales were added.

Change Number: XSH/TC1/D5/0103 [167]

On Page: 779 Line: 25989 Section: exec

In the APPLICATION USAGE section, change from:

The environ array should not be accessed directly by the application.

The new process might be invoked in a non-conforming environment if the envp array does not contain implementation-defined variables required by the implementation to provide a conforming environment. See the _CS_V7_ENV entry in <unistd.h> and confstr() for details.

to:

When assigning a new value to the environ variable, applications should ensure that the environment to which it will point contains at least the following:

a. Any implementation-defined variables required by the implementation to provide a conforming environment. See the _CS_V7_ENV entry in <unistd.h> and confstr() for details.

b. A value for PATH which finds conforming versions of all standard utilities before any other versions.

The same constraint applies to the envp array passed to execle() or execve(), in order to ensure that the new process image is invoked in a conforming environment.

Rationale: Austin Group Defect Report(s) applied: 167.
See http://austingroupbugs.net/view.php?id=167

Change Number: XSH/TC1/D5/0104 [173]

On Page: 779 Line: 25993 Section: exec

In the APPLICATION USAGE section, change from:

Applications should not depend on file descriptors 0, 1, and 2 being closed after an exec. A future version may allow these file descriptors to be automatically opened for any process.

to:

Applications should not execute programs with file descriptor 0 not open for reading or with file descriptor 1 or 2 not open for writing, as this might cause the executed program to misbehave. In order not to pass on these file descriptors to an executed program, applications should not just close them but should reopen them on, for example, /dev/null. Some implementations may reopen them automatically, but applications should not rely on this being done.

Rationale: Austin Group Defect Report(s) applied: 173.
See http://austingroupbugs.net/view.php?id=173

Change Number: XSH/TC1/D5/0105 [291,429]

On Page: 780 Line: 26018 Section: exec

In the RATIONALE section, change from:

The requirement on a Strictly Conforming POSIX Application also states that the value passed as the first argument be a filename associated with the process being started. Although some existing applications pass a pathname rather than a filename in some circumstances, a filename is more generally useful, since the common usage of argv[0] is in printing diagnostics. In some cases the filename passed is not the actual filename of the file; for example, many implementations of the login utility use a convention of prefixing a <hyphen> ('-') to the actual filename, which indicates to the command interpreter being invoked that it is a "login shell".

to:

The requirement on a Strictly Conforming POSIX Application also states that the value passed as the first argument be a filename string associated with the process being started. Although some existing applications pass a pathname rather than a filename string in some circumstances, a filename string is more generally useful, since the common usage of argv[0] is in printing diagnostics. In some cases the filename passed is not the actual filename of the file; for example, many implementations of the login utility use a convention of prefixing a <hyphen> ('-') to the actual filename, which indicates to the command interpreter being invoked that it is a "login shell".

Rationale: Austin Group Defect Report(s) applied: 291,429.
See http://austingroupbugs.net/view.php?id=291
See http://austingroupbugs.net/view.php?id=429

Change Number: XSH/TC1/D5/0106 [68]

On Page: 786 Line: 26265 Section: exp()

In the RETURN VALUE section, change from:

If the correct value would cause underflow, and is not representable, a range error may occur, and [MX]either 0.0 (if supported), or[/MX] an implementation-defined value shall be returned.

to:

If the correct value would cause underflow, [MXX]and is not representable[/MXX], a range error may occur, and exp(), expf(), and expl() shall return [MXX]0.0, or[/MXX] (if IEC 60559 Floating-Point is not supported) an implementation-defined value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0107 [68]

On Page: 786 Line: 26271 Section: exp()

In the RETURN VALUE section, change the MX shading to MXX for:

If the correct value would cause underflow, and is representable, a range error may occur and the correct value shall be returned.

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0108 [68]

On Page: 788 Line: 26342 Section: exp2()

In the RETURN VALUE section, change from:

If the correct value would cause underflow, and is not representable, a range error may occur, and [MX]either 0.0 (if supported), or[/MX] an implementation-defined value shall be returned.

to:

If the correct value would cause underflow, [MXX]and is not representable[/MXX], a range error may occur, and exp2(), exp2f(), and exp2l() shall return [MXX]0.0, or[/MXX] (if IEC 60559 Floating-Point is not supported) an implementation-defined value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0109 [68]

On Page: 788 Line: 26348 Section: exp2()

In the RETURN VALUE section, change the MX shading to MXX for:

If the correct value would cause underflow, and is representable, a range error may occur and the correct value shall be returned.

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0110 [68]

On Page: 790 Line: 26404 Section: expm1()

In the RETURN VALUE section, change from:

[MX]If x is subnormal, a range error may occur and x should be returned.[/MX]

to:

[MX]If x is subnormal, a range error may occur[/MX] [MXX]and x should be returned.[/MXX]

[MX]If x is not returned, expm1(), expm1f(), and expm1l() shall return an implementation-defined value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.[/MX]

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0111 [146,324]

On Page: 795 Line: 26553 Section: fattach()

In the ERRORS section, change from:

[ENOTDIR] A component of the path prefix is not a directory, or the path argument contains at least one non-<slash> character and ends with one or more trailing <slash> characters and the last pathname component

names an existing file that is neither a directory nor a symbolic link to a directory.

to:

[ENOTDIR] A component of the path prefix names an existing file that is neither a directory nor a symbolic link to a directory, or the path argument contains at least one non-<slash> character and ends with one or more trailing <slash> characters.

Rationale: Austin Group Defect Report(s) applied: 146,324.
See http://austingroupbugs.net/view.php?id=146
See http://austingroupbugs.net/view.php?id=324
The [ENOTDIR] error is updated, including an editorial clarification for the component of the path prefix.

Change Number: XSH/TC1/D5/0112 [291]

On Page: 796 Line: 26577-26579 Section: fattach()

In the EXAMPLES section, change both instances of "filename" to "pathname".

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XSH/TC1/D5/0113 [87]

On Page: 805 Line: 26801 Section: fclose()

In the DESCRIPTION section, change from:

... the file offset of the underlying open file description shall be adjusted so that the next operation on the open file description deals with the byte after the last one read from or written to the stream being closed.

to:

... the file offset of the underlying open file description shall be set to the file position of the stream if the stream is the active handle to the underlying file description.

Rationale: Austin Group Defect Report(s) applied: 87.
See http://austingroupbugs.net/view.php?id=87

Change Number: XSH/TC1/D5/0114 [79]

On Page: 805 Line: 26824 Section: fclose()

In the ERRORS section [EIO], change from:

... the process is neither ignoring nor blocking SIGTTOU, and the process group of the process is orphaned.

to:

... the calling thread is not blocking SIGTTOU, the process is not

ignoring SIGTTOU, and the process group of the process is orphaned.

Rationale: Austin Group Defect Report(s) applied: 79.
See http://austingroupbugs.net/view.php?id=79

Change Number: XSH/TC1/D5/0115 [14]

On Page: 806 Line: 26845 Section: fclose()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0116 [141]

On Page: 807 Line: 26910 Section: fcntl()

In the DESCRIPTION section (for F_SETFL), add to the end of the paragraph the additional sentence:

If fildes does not support non-blocking operations, it is unspecified whether the O_NONBLOCK flag will be ignored.

Rationale: Austin Group Defect Report(s) applied: 141.
See http://austingroupbugs.net/view.php?id=141

Change Number: XSH/TC1/D5/0117 [324]

On Page: 816 Line: 27274 Section: fdetach()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of the path prefix is not a directory, ...

to:

A component of the path prefix names an existing file that is neither a directory nor a symbolic link to a directory, ...

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0118 [291]

On Page: 817 Line: L27294-27296 Section: fdetach()

In the EXAMPLES section, change both instances of "filename" to "pathname".

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XSH/TC1/D5/0119 [68,428]

On Page: 818 Line: 27340 Section: fdim()

In the RETURN VALUE section, change from:

If x-y is positive and underflows, a range error may occur, and either (x-y) (if representable), [XSI]or 0.0 (if supported),[/XSI] or an implementation-defined value shall be returned.

to:

If the correct value would cause underflow, a range error may occur, and fdim(), fdimf(), and fdiml() shall return [MXX]the correct value, or[/MXX] (if IEC 60559 Floating-Point is not supported) an implementation-defined value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.

Rationale: Austin Group Defect Report(s) applied: 68,428.
See http://austingroupbugs.net/view.php?id=68
See http://austingroupbugs.net/view.php?id=428

Change Number: XSH/TC1/D5/0120 [68,428]

On Page: 819 Line: 27359 Section: fdim()

In the APPLICATION USAGE section, delete the following text:

On implementations supporting IEEE Std 754-1985, x-y cannot underflow, and hence the 0.0 return value is shaded as an extension for implementations supporting the XSI option rather than an MX extension.

Rationale: Austin Group Defect Report(s) applied: 68,428.
See http://austingroupbugs.net/view.php?id=68
See http://austingroupbugs.net/view.php?id=428

Change Number: XSH/TC1/D5/0121 [409]

On Page: 821 Line: 27421 Section: fdopen()

In the RATIONALE section, change the text style from:

<italics>a b</italics>

to:

a <italics>b</italics>

Rationale: Austin Group Defect Report(s) applied: 409.
See http://austingroupbugs.net/view.php?id=409

Change Number: XSH/TC1/D5/0122 [422]

On Page: 823 Line: 27468 Section: fdopendir()

In the DESCRIPTION section, change from:

Upon successful return from fdopendir(), the file descriptor is under the control of the system, and if any attempt is made to close the file descriptor, or to modify the state of the associated description, other than by means of closedir(), readdir(), readdir_r(), or rewinddir(), ...

to:

Upon successful return from fdopendir(), the file descriptor is under the control of the system, and if any attempt is made to close the file descriptor, or to modify the state of the associated description, other than by means of closedir(), readdir(), readdir_r(), rewinddir(), or [XSI]seekdir()[/XSI], ...

Rationale: Austin Group Defect Report(s) applied: 422.
See http://austingroupbugs.net/view.php?id=422

Change Number: XSH/TC1/D5/0123 [324]

On Page: 823 Line: 27494 Section: fdopendir()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of dirname is not a directory.

to:

A component of dirname names an existing file that is neither a directory nor a symbolic link to a directory.

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0124 [401]

On Page: 833 Line: 27807 Section: feof()

In the DESCRIPTION section, add to the end of the section:

[CX]The feof() function shall not change the setting of errno if stream is valid.[/CX]

Rationale: Austin Group Defect Report(s) applied: 401.
See http://austingroupbugs.net/view.php?id=401

Change Number: XSH/TC1/D5/0125 [401]

On Page: 835 Line: 27871 Section: ferror()

In the DESCRIPTION section, add to the end of the section:

[CX]The ferror() function shall not change the setting of errno if stream is valid.[/CX]

Rationale: Austin Group Defect Report(s) applied: 401.
See http://austingroupbugs.net/view.php?id=401

Change Number: XSH/TC1/D5/0126 [87]

On Page: 844 Line: 28024 Section: fflush()

In the DESCRIPTION section, change from:

... the file offset of the underlying open file description shall be adjusted so that the next operation on the open file description deals with the byte after the last one read from or written to the stream being flushed.

to:

... the file offset of the underlying open file description shall be set to the file position of the stream, and any characters pushed back onto the stream by ungetc() or ungetwc() that have not subsequently been read from the stream shall be discarded (without further changing the file offset).

Rationale: Austin Group Defect Report(s) applied: 87.
See http://austingroupbugs.net/view.php?id=87

Change Number: XSH/TC1/D5/0127 [79]

On Page: 844 Line: 28042 Section: fflush()

In the ERRORS section [EIO], change from:

... the process is neither ignoring nor blocking SIGTTOU, and the process group of the process is orphaned.

to:

... the calling thread is not blocking SIGTTOU, the process is not ignoring SIGTTOU, and the process group of the process is orphaned.

Rationale: Austin Group Defect Report(s) applied: 79.
See http://austingroupbugs.net/view.php?id=79

Change Number: XSH/TC1/D5/0128 [14]

On Page: 846 Line: 28101 Section: fflush()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0129 [79]

On Page: 848 Line: 28184 Section: fgetc()

In the ERRORS section [EIO], change from:

... and either the process is ignoring or blocking the SIGTTIN signal or the process group is orphaned.

to:

... and either the calling thread is blocking SIGTTIN or the process is ignoring SIGTTIN or the process group of the process is orphaned.

Rationale: Austin Group Defect Report(s) applied: 79.
See http://austingroupbugs.net/view.php?id=79

Change Number: XSH/TC1/D5/0130 [14]

On Page: 849 Line: 28206 Section: fgetc()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0131 [105]

On Page: 850 Line: 28241 Section: fgetpos()

In the DESCRIPTION section, add a new paragraph at the end of the DESCRIPTION:

The fgetpos() function shall not change the setting of errno if successful.

Rationale: Austin Group Defect Report(s) applied: 105.
See http://austingroupbugs.net/view.php?id=105

Change Number: XSH/TC1/D5/0132 [122]

On Page: 850 Line: 28249 Section: fgetpos()

In the ERRORS section, change from:

The fgetpos() function shall fail if:

CX [EOVERFLOW] The current value of the file position cannot be represented correctly in an object of type fpos_t.

The fgetpos() function may fail if:

CX [EBADF] The file descriptor underlying stream is not valid.

CX [ESPIPE] The file descriptor underlying stream is associated with a pipe, FIFO, or socket.

to:

The fgetpos() function shall fail if:

CX [EBADF] The file descriptor underlying stream is not valid.

CX [EOVERFLOW] The current value of the file position cannot be represented correctly in an object of type fpos_t.

CX [ESPIPE] The file descriptor underlying stream is associated with a pipe, FIFO, or socket.

Rationale: Austin Group Defect Report(s) applied: 122.
See http://austingroupbugs.net/view.php?id=122
The [EBADF] and [ESPIPE] errors are made mandatory.

Change Number: XSH/TC1/D5/0133 [14]

On Page: 850 Line: 28261 Section: fgetpos()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0134 [182]

On Page: 852 Line: 28298-28308 Section: fgets()

In the EXAMPLES section, change from:

The following example uses fgets() to read each line of input. {LINE_MAX}, which defines the maximum size of the input line, is defined in the <limits.h> header.

```
#include <stdio.h>
...
char line[LINE_MAX];
...
while (fgets(line, LINE_MAX, fp) != NULL) {
    ...
}
...
```

to:

The following example uses fgets() to read lines of input. It assumes that the file it is reading is a text file and that lines in this text file are no longer than 16384 (or {LINE_MAX} if it is less than 16384 on the implementation where it is running) bytes long. (Note that the standard utilities have no line length limit if sysconf(_SC_LINE_MAX) returns -1 without setting errno. This example assumes that sysconf(_SC_LINE_MAX) will not fail.)

```
#include <limits.h>
#include <stdio.h>
#include <unistd.h>
#define MYLIMIT 16384

char    *line;
int     line_max;
```

```
if (LINE_MAX >= MYLIMIT) {
    // Use maximum line size of MYLIMIT. If LINE_MAX is
    // bigger than our limit, sysconf() can't report a
    // smaller limit.
    line_max = MYLIMIT;
} else {
    long limit = sysconf(_SC_LINE_MAX);
    line_max = (limit < 0 || limit > MYLIMIT) ? MYLIMIT : (int)limit;
}

// line_max + 1 leaves room for the nul byte added by fgets().
line = malloc(line_max + 1);
if (line == NULL) {
    // out of space
    ...
    return error;
}

while (fgets(line, line_max + 1, fp) != NULL) {
    // Verify that a full line has been read...
    //    If not, report an error or prepare to treat the
    //    next time through the loop as a read of a
    //    continuation of the current line.
    ...
    // Process line...
    ...
}
free(line);
...
```

Rationale: Austin Group Defect Report(s) applied: 182.
See http://austingroupbugs.net/view.php?id=182

Change Number: XSH/TC1/D5/0135 [14]

On Page: 853 Line: 28316 Section: fgets()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0136 [105]

On Page: 854 Line: 28343 Section: fgetwc()

In the DESCRIPTION section, add a new paragraph at the end of the DESCRIPTION:

The fgetwc() function shall not change the setting of errno if successful.

Rationale: Austin Group Defect Report(s) applied: 105.
See http://austingroupbugs.net/view.php?id=105

Change Number: XSH/TC1/D5/0137 [79]

On Page: 854 Line: 28362 Section: fgetwc()

In the ERRORS section [EIO], change from:

... and either the process is ignoring or blocking the SIGTTIN signal
or the process group is orphaned.

to:

... and either the calling thread is blocking SIGTTIN or the process is
ignoring SIGTTIN or the process group of the process is orphaned.

Rationale: Austin Group Defect Report(s) applied: 79.
See http://austingroupbugs.net/view.php?id=79

Change Number: XSH/TC1/D5/0138 [14]

On Page: 855 Line: 28381 Section: fgetwc()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0139 [14]

On Page: 856 Line: 28437 Section: fgetws()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0140 [118]

On Page: 859 Line: 28511 Section: flockfile()

In the DESCRIPTION section, change from:

All functions that reference (FILE *) objects shall behave ...

to:

All functions that reference (FILE *) objects, except those with names
ending in _unlocked, shall behave ...

Rationale: Austin Group Defect Report(s) applied: 118.
See http://austingroupbugs.net/view.php?id=118

Change Number: XSH/TC1/D5/0141 [346]

On Page: 861 Line: 28567- Section: floor()

For floor(), delete P861 L28567-28570 (DESCRIPTION), L28577-28579 (RETURN VALUE), and replace L28581-28586 (ERRORS) with:

No errors are defined.

At L28572 (RETURN VALUE), add:

[MX]The result shall have the same sign as x.[/MX]

At L28590 (APPLICATION USAGE), change "an int or long" to "an intmax_t", and replace L28593-28596 (APPLICATION USAGE) with:

These functions may raise the inexact floating-point exception if the result differs in value from the argument.

Rationale: Austin Group Defect Report(s) applied: 346.
See http://austingroupbugs.net/view.php?id=346

Change Number: XSH/TC1/D5/0142 [461]

On Page: 866 Line: 28742 Section: fmemopen()

In the DESCRIPTION section, change from:

The mode argument is a character string having one of the following values:

to:

The mode argument points to a string. If the string is one of the following, the stream shall be opened in the indicated mode. Otherwise, the behavior is undefined.

Rationale: Austin Group Defect Report(s) applied: 461.
See http://austingroupbugs.net/view.php?id=461
This change is to match the wording used in fopen().

Change Number: XSH/TC1/D5/0143 [396]

On Page: 866 Line: 28743 Section: fmemopen()

In the DESCRIPTION section, change the first column from:

r or rb
w or wb
a or ab
r+ or rb+ or r+b
w+ or wb+ or w+b
a+ or ab+ or a+b

to:

r
w
a
r+
w+
a+

Rationale: Austin Group Defect Report(s) applied: 396.
See http://austingroupbugs.net/view.php?id=396

Change Number: XSH/TC1/D5/0144 [396]

On Page: 866 Line: 28751 Section: fmemopen()

In the DESCRIPTION section, change from:

The character 'b' shall have no effect.

to:

Implementations shall accept all mode strings allowed by fopen(), but the use of the character 'b' shall produce implementation-defined results, where the resulting FILE * need not behave the same as if 'b' were omitted.

Rationale: Austin Group Defect Report(s) applied: 396.
See http://austingroupbugs.net/view.php?id=396

Change Number: XSH/TC1/D5/0145 [461]

On Page: 866 Line: 28757 Section: fmemopen()

In the DESCRIPTION section, change from:

The stream maintains a current position in the buffer. This position is initially set to either the beginning of the buffer (for r and w modes) or to the first null byte in the buffer (for a modes). If no null byte is found in append mode, the initial position is set to one byte after the end of the buffer.

to:

The stream shall maintain a current position in the buffer. This position shall be initially set to either the beginning of the buffer (for r and w modes) or to the first null byte in the buffer (for a modes). If no null byte is found in append mode, the initial position shall be set to one byte after the end of the buffer.

Rationale: Austin Group Defect Report(s) applied: 461.
See http://austingroupbugs.net/view.php?id=461

Change Number: XSH/TC1/D5/0146 [461]

On Page: 866 Line: 28762 Section: fmemopen()

In the DESCRIPTION section, change from:

The stream also maintains the size of the current buffer contents. For modes r and r+ the size is set to the value given by the size argument. For modes w and w+ the initial size is zero and for modes a and a+ the initial size is either the position of the first null byte in the buffer or the value of the size argument if no null byte is found.

to:

The stream shall also maintain the size of the current buffer contents; use of fseek() or fseeko() on the stream with SEEK_END shall seek relative to this size. For modes r and r+ the size shall be set to the value given by the size argument. For modes w and w+ the initial size shall be zero and for modes a and a+ the initial size shall be either the position of the first null byte in the buffer or the value of the size argument if no null byte is found.

Rationale: Austin Group Defect Report(s) applied: 461.
See http://austingroupbugs.net/view.php?id=461

Change Number: XSH/TC1/D5/0147 [461]

On Page: 866 Line: 28766 Section: fmemopen()

In the DESCRIPTION section, change from:

A read operation on the stream cannot advance the current buffer position beyond the current buffer size. Reaching the buffer size in a read operation counts as "end-of-file". Null bytes in the buffer have no special meaning for reads. The read operation starts at the current buffer position of the stream.

to:

A read operation on the stream shall not advance the current buffer position beyond the current buffer size. Reaching the buffer size in a read operation shall count as "end-of-file". Null bytes in the buffer shall have no special meaning for reads. The read operation shall start at the current buffer position of the stream.

Rationale: Austin Group Defect Report(s) applied: 461.
See http://austingroupbugs.net/view.php?id=461

Change Number: XSH/TC1/D5/0148 [461]

On Page: 866 Line: 28770 Section: fmemopen()

In the DESCRIPTION section, change from:

A write operation starts either at the current position of the stream (if mode has not specified 'a' as the first character) or at the current size of the stream (if mode had 'a' as the first character). If the current position at the end of the write is larger than the current buffer size, the current buffer size is set to the current position. A write operation on the stream cannot advance the current buffer size beyond the size given in the size argument.

to:

A write operation shall start either at the current position of the stream (if mode has not specified 'a' as the first character) or at the current size of the stream (if mode had 'a' as the first character). If the current position at the end of the write is larger than the current buffer size, the current buffer size shall be set to the current position. A write operation on the stream shall not advance the current buffer size beyond the size given in the size argument.

Rationale: Austin Group Defect Report(s) applied: 461.
See http://austingroupbugs.net/view.php?id=461

Change Number: XSH/TC1/D5/0149 [461]

On Page: 867 Line: 28775 Section: fmemopen()

In the DESCRIPTION section, change from:

When a stream open for writing is flushed or closed, a null byte is written at the current position or at the end of the buffer, depending on the size of the contents. If a stream open for update is flushed or closed and the last write has advanced the current buffer size, a null byte is written at the end of the buffer if it fits.

to:

When a stream open for writing is flushed or closed, a null byte shall be written at the current position or at the end of the buffer, depending on the size of the contents. If a stream open for update is flushed or closed and the last write has advanced the current buffer size, a null byte shall be written at the end of the buffer if it fits.

Rationale: Austin Group Defect Report(s) applied: 461.
See http://austingroupbugs.net/view.php?id=461

Change Number: XSH/TC1/D5/0150 [396]

On Page: 868 Line: 28824 Section: fmemopen()

In the FUTURE DIRECTIONS section, change from:

None.

to:

A future revision of this standard may mandate specific behavior when the mode argument includes 'b'.

Rationale: Austin Group Defect Report(s) applied: 396.
See http://austingroupbugs.net/view.php?id=396

Change Number: XSH/TC1/D5/0151 [68]

On Page: 870 Line: 28885 Section: fmod()

In the RETURN VALUE section, change from:

If the correct value would cause underflow, and is not representable, a range error may occur, and [MX]either 0.0 (if supported), or[/MX] an implementation-defined value shall be returned.

to:

If the correct value would cause underflow, [MXX]and is not representable[/MXX], a range error may occur, and fmod(), modf(), and

fmodl() shall return [MXX]0.0, or[/MXX] (if IEC 60559 Floating-Point is not supported) an implementation-defined value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0152 [320]

On Page: 870 Line: 28888,28890 Section: fmod()

In the RETURN VALUE section, on lines 28888 and 28890 change from:

... and either a NaN (if supported), or an implementation-defined value shall be returned.

to:

... and a NaN shall be returned.

Rationale: Austin Group Defect Report(s) applied: 320.
See http://austingroupbugs.net/view.php?id=320

Change Number: XSH/TC1/D5/0153 [68]

On Page: 870 Line: 28894 Section: fmod()

In the RETURN VALUE section, change the MX shading to MXX for:

If the correct value would cause underflow, and is representable, a range error may occur and the correct value shall be returned.

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0154 [291]

On Page: 875 Line: 29045 Section: fnmatch()

In the NAME section, change from:

filename

to:

filename string

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XSH/TC1/D5/0155 [291]

On Page: 875 Line: 29084 Section: fnmatch()

In the APPLICATION USAGE section, change from:

5119
5120 match filenames, rather than pathnames
5121
5122 to:
5123
5124 match filename strings, rather than pathnames
5125
5126 *Rationale:* Austin Group Defect Report(s) applied: 291.
5127 See http://austingroupbugs.net/view.php?id=291
5128
5129
5130 **Change Number: XSH/TC1/D5/0156** [291,433]
5131
5132
5133 On Page: 877 Line: 29108,29113,29143,29184 Section: fopen()
5134
5135 Change all instances of "filename" to "pathname".
5136
5137 *Rationale:* Austin Group Defect Report(s) applied: 291,433.
5138 See http://austingroupbugs.net/view.php?id=291
5139 See http://austingroupbugs.net/view.php?id=433
5140
5141
5142 **Change Number: XSH/TC1/D5/0157** [146,433]
5143
5144
5145 On Page: 878 Line: 29180 Section: fopen()
5146
5147 In the ERRORS section, change from:
5148
5149 [ENOENT] A component of filename does not name an existing file or
5150 filename is an empty string.
5151
5152 to:
5153
5154 [ENOENT] The mode string begins with 'r' and a component of pathname
5155 does not name an existing file, or mode begins with 'w' or 'a' and a
5156 component of the path prefix of pathname does not name an existing file,
5157 or pathname is an empty string.
5158
5159 [ENOENT] or [ENOTDIR] The pathname argument contains at least one
5160 non-<slash> character and ends with one or more trailing <slash>
5161 characters. If pathname names an existing file, an [ENOENT] error shall
5162 not occur.
5163
5164 *Rationale:* Austin Group Defect Report(s) applied: 146,433.
5165 See http://austingroupbugs.net/view.php?id=146
5166 See http://austingroupbugs.net/view.php?id=433
5167
5168
5169 **Change Number: XSH/TC1/D5/0158** [324]
5170
5171
5172 On Page: 878 Line: 29184 Section: fopen()
5173
5174 In the ERRORS section, for the [ENOTDIR] error, change from:
5175
5176 A component of the path prefix is not a directory, ...
5177
5178 to:
5179
5180 A component of the path prefix names an existing file that is neither
5181 a directory nor a symbolic link to a directory, ...

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0159 [14]

On Page: 879 Line: 29227 Section: fopen()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
Editorial improvement.

Change Number: XSH/TC1/D5/0160 [256,428]

On Page: 886 Line: 29481 Section: fpathconf()

In the DESCRIPTION section, change from:

3,4

to:

4,7

Rationale: Austin Group Defect Report(s) applied: 256,428.
See http://austingroupbugs.net/view.php?id=256
See http://austingroupbugs.net/view.php?id=428

Change Number: XSH/TC1/D5/0161 [256,428]

On Page: 887 Line: 29510 Section: fpathconf()

In the DESCRIPTION section, change from:

5. If path or fildes refers to a directory, the value returned shall be the maximum length of a relative pathname when the specified directory is the working directory.

to:

5. If path or fildes refers to a directory, the value returned shall be the maximum length of a relative pathname that would not cross any mount points when the specified directory is the working directory.

Rationale: Austin Group Defect Report(s) applied: 256,428.
See http://austingroupbugs.net/view.php?id=256
See http://austingroupbugs.net/view.php?id=428

Change Number: XSH/TC1/D5/0162 [324]

On Page: 888 Line: 29567 Section: fpathconf()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of the path prefix is not a directory, ...

to:

A component of the path prefix names an existing file that is neither a directory nor a symbolic link to a directory, ...

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0163 [302]

On Page: 894 Line: 29763 Section: fprintf()

In the DESCRIPTION section, change from:

... the process' locale ...

to:

... the current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when per-thread locales were added.

Change Number: XSH/TC1/D5/0164 [316]

On Page: 899 Line: 30004 Section: fprintf()

In the ERRORS section, add after the [EILSEQ] error:

[EOVERFLOW] [CX] The value to be returned is greater than {INT_MAX}.[/CX]

Rationale: Austin Group Defect Report(s) applied: 316.
See http://austingroupbugs.net/view.php?id=316

Change Number: XSH/TC1/D5/0165 [316]

On Page: 899 Line: 30012 Section: fprintf()

In the ERRORS section, for the snprintf() [EOVERFLOW] error, change from:

[EOVERFLOW] [CX]The value of n is greater than {INT_MAX} or the number of bytes needed to hold the output excluding the terminating null is greater than {INT_MAX}.

to:

[EOVERFLOW] [CX]The value of n is greater than {INT_MAX}.

Rationale: Austin Group Defect Report(s) applied: 316.
See http://austingroupbugs.net/view.php?id=316

Change Number: XSH/TC1/D5/0166 [451,291]

On Page: 902 Line: 30140 Section: fprintf()

In the EXAMPLES section, change from:

Creating a Filename

The following example creates a filename using information from a previous getpwnam() function that returned the HOME directory of the user.

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
...
char filename[PATH_MAX+1];
struct passwd *pw;
...
sprintf(filename, "%s/%d.out", pw->pw_dir, getpid());
...
```

to:

Creating a Pathname

The following example creates a pathname using information from a previous getpwnam() function that returned the password database entry of the user.

```
#include <stdint.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <unistd.h>
...
char *pathname;
struct passwd *pw;
size_t len;
...
// digits required for pid_t is number of bits times log2(10) = approx 10/33
len = strlen(pw->pw_dir) + 1 + 1+(sizeof(pid_t)*80+32)/33 + sizeof ".out";
pathname = malloc(len);
if (pathname != NULL)
{
   snprintf(pathname, len, "%s/%jd.out", pw->pw_dir, (intmax_t)getpid());
   ...
}
```

Rationale: Austin Group Defect Report(s) applied: 451,291.
See http://austingroupbugs.net/view.php?id=451
See http://austingroupbugs.net/view.php?id=291

Change Number: XSH/TC1/D5/0167 [14]

On Page: 904 Line: 30229 Section: fprintf()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0168 [79]

On Page: 906 Line: 30300 Section: fputc()

In the ERRORS section [EIO], change from:

... the process is neither ignoring nor blocking SIGTTOU, and the process group of the process is orphaned.

to:

... the calling thread is not blocking SIGTTOU, the process is not ignoring SIGTTOU, and the process group of the process is orphaned.

Rationale: Austin Group Defect Report(s) applied: 79.
See http://austingroupbugs.net/view.php?id=79

Change Number: XSH/TC1/D5/0169 [14]

On Page: 907 Line: 30319 Section: fputc()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement.

Change Number: XSH/TC1/D5/0170 [174,412]

On Page: 908 Line: 30373 Section: fputs()

In the APPLICATION USAGE section, add to the end of the section:

This volume of POSIX.1-2008 requires that successful completion simply return a non-negative integer. There are at least three known different implementation conventions for this requirement:

 * Return a constant value.
 * Return the last character written.
 * Return the number of bytes written. Note that this implementation convention cannot be adhered to for strings longer than {INT_MAX} bytes as the value would not be representable in the return type of the function. For backwards compatibility, implementations can return the number of bytes for strings of up to INT_MAX bytes, and return INT_MAX for all longer strings.

Rationale: Austin Group Defect Report(s) applied: 174,412.
See http://austingroupbugs.net/view.php?id=174
See http://austingroupbugs.net/view.php?id=412

Change Number: XSH/TC1/D5/0171 [412]

On Page: 908 Line: 30375 Section: fputs()

In the RATIONALE section, change from:

None

to:

The fputs() function is one whose source code was specified in _The C Programming Language_ by Brian W. Kernighan and Dennis M. Ritchie. In the original edition, the function had no defined return value, yet many practical implementations would, as a side effect, return the value of the last character written as that was the value remaining in the accumulator used as a return value. In the second edition of the book, either the fixed value 0 or EOF would be returned depending upon the return value of ferror(); however, for compatibility with extant implementations, several implementations would, upon success, return a positive value representing last byte written.

Rationale: Austin Group Defect Report(s) applied: 412.
See http://austingroupbugs.net/view.php?id=412

Change Number: XSH/TC1/D5/0172 [14]

On Page: 909 Line: 30379 Section: fputs()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement.

Change Number: XSH/TC1/D5/0173 [105]

On Page: 910 Line: 30406 Section: fputwc()

In the DESCRIPTION section, add a new paragraph at the end of the DESCRIPTION:

The fputwc() function shall not change the setting of errno if successful.

Rationale: Austin Group Defect Report(s) applied: 105.
See http://austingroupbugs.net/view.php?id=105

Change Number: XSH/TC1/D5/0174 [79]

On Page: 910 Line: 30426 Section: fputwc()

In the ERRORS section [EIO], change from:

... the process is neither ignoring nor blocking SIGTTOU, and the process group of the process is orphaned.

to:

... the calling thread is not blocking SIGTTOU, the process is not ignoring SIGTTOU, and the process group of the process is orphaned.

Rationale: Austin Group Defect Report(s) applied: 79.
See http://austingroupbugs.net/view.php?id=79

Change Number: XSH/TC1/D5/0175 [14]

On Page: 911 Line: 30445 Section: fputwc()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0176 [412]

On Page: 912 Line: 30489 Section: fputws()

In the APPLICATION USAGE section, add to the end of the section:

This volume of POSIX.1-2008 requires that successful completion simply return a non-negative integer. There are at least three known different implementation conventions for this requirement:

* Return a constant value.
* Return the last character written.
* Return the number of bytes written. Note that this implementation convention cannot be adhered to for strings longer than {INT_MAX} bytes as the value would not be representable in the return type of the function. For backwards compatibility, implementations can return the number of bytes for strings of up to INT_MAX bytes, and return INT_MAX for all longer strings.

Rationale: Austin Group Defect Report(s) applied: 412.
See http://austingroupbugs.net/view.php?id=412

Change Number: XSH/TC1/D5/0177 [14]

On Page: 912 Line: 30495 Section: fputws()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0178 [232]

On Page: 913 Line: 30537 Section: fread()

In the EXAMPLES section, change from:

The following example reads a single element from the fp stream into the array pointed to by buf.

```
#include <stdio.h>
...
size_t bytes_read;
char buf[100];
FILE *fp;
...
bytes_read = fread(buf, sizeof(buf), 1, fp);
...
```

to:

The following example reads a single element from the fp stream into the array pointed to by buf.

```
#include <stdio.h>
...
size_t elements_read;
char buf[100];
FILE *fp;
...
elements_read = fread(buf, sizeof(buf), 1, fp);
...
```

If a read error occurs, elements_read will be zero but the number of bytes read from the stream could be anything from zero to sizeof(buf)-1.

The following example reads multiple single-byte elements from the fp stream into the array pointed to by buf.

```
#include <stdio.h>
...
size_t bytes_read;
char buf[100];
FILE *fp;
...
bytes_read = fread(buf, 1, sizeof(buf), fp);
...
```

If a read error occurs, bytes_read will contain the number of bytes read from the stream.

Rationale: Austin Group Defect Report(s) applied: 232.
See http://austingroupbugs.net/view.php?id=232

Change Number: XSH/TC1/D5/0179 [14]

On Page: 914 Line: 30558 Section: fread()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0180 [283]

On Page: 921 Line: 30834 Section: freelocale()

In the DESCRIPTION section, add a new paragraph:

The behavior is undefined if the locobj argument is the special locale object LC_GLOBAL_LOCALE or is not a valid locale object handle.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0181 [291,433]

On Page: 923 Line: 30870 Section: freopen()

On L30870, L30877, L30881, L30885, L30893, L30914, L30927, L30939, L30964, change all instances of "filename" to "pathname".

Rationale: Austin Group Defect Report(s) applied: 291,433.
See http://austingroupbugs.net/view.php?id=291
See http://austingroupbugs.net/view.php?id=433

Change Number: XSH/TC1/D5/0182 [146,433]

On Page: 924 Line: 30923 Section: freopen()

In the ERRORS section, change from:

[ENOENT] A component of filename does not name an existing file or filename is an empty string.

to:

[ENOENT] The mode string begins with 'r' and a component of pathname does not name an existing file, or mode begins with 'w' or 'a' and a component of the path prefix of pathname does not name an existing file, or pathname is an empty string.

[ENOENT] or [ENOTDIR] The pathname argument contains at least one non-<slash> character and ends with one or more trailing <slash> characters. If pathname names an existing file, an [ENOENT] error shall not occur.

Rationale: Austin Group Defect Report(s) applied: 146,433.
See http://austingroupbugs.net/view.php?id=146
See http://austingroupbugs.net/view.php?id=433

Change Number: XSH/TC1/D5/0183 [324]

On Page: 924 Line: 30927 Section: freopen()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of the path prefix is not a directory, ...

to:

A component of the path prefix names an existing file that is neither

a directory nor a symbolic link to a directory, ...

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the
standard.

Change Number: XSH/TC1/D5/0184 [14]

On Page: 925 Line: 30983 Section: freopen()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0185 [302]

On Page: 929 Line: 31097 Section: fscanf()

change from:

... the locale of the process ...

to:

... the current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0186 [90]

On Page: 934 Line: 31304 Section: fscanf()

In the RETURN VALUE section, change from:

If the input ends before the first matching failure or conversion, EOF
shall be returned. If any error occurs, EOF shall be returned, [CX]and
errno shall be set to indicate the error[/CX]. If a read error occurs,
the error indicator for the stream shall be set.

to:

If the input ends before the first conversion (if any) has completed,
and without a matching failure having occurred, EOF shall be returned. If
an error occurs before the first conversion (if any) has completed,
and without a matching failure having occurred, EOF shall be returned
[CX]and errno shall be set to indicate the error[/CX]. If a read error
occurs, the error indicator for the stream shall be set.

Add to the CHANGE HISTORY section:

A change to the second sentence in RETURN VALUE is made to align with

expected wording changes in the next revision of the C standard.

Rationale: Austin Group Defect Report(s) applied: 90.
See http://austingroupbugs.net/view.php?id=90

Change Number: XSH/TC1/D5/0187 [14]

On Page: 935 Line: 31351 Section: fscanf()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0188 [79]

On Page: 938 Line: 31438 Section: fseek()

In the ERRORS section [EIO], change from:

... the process is neither ignoring nor blocking SIGTTOU, and the process group of the process is orphaned.

to:

... the calling thread is not blocking SIGTTOU, the process is not ignoring SIGTTOU, and the process group of the process is orphaned.

Rationale: Austin Group Defect Report(s) applied: 79.
See http://austingroupbugs.net/view.php?id=79

Change Number: XSH/TC1/D5/0189 [122]

On Page: 938 Line: 31450 Section: fseek()

In the ERRORS section, change from:

CX [ESPIPE] The file descriptor underlying stream is associated with a pipe or FIFO.

to:

CX [ESPIPE] The file descriptor underlying stream is associated with a pipe, FIFO, or socket.

Rationale: Austin Group Defect Report(s) applied: 122.
See http://austingroupbugs.net/view.php?id=122
The [ESPIPE] error is updated to also apply to a socket.

Change Number: XSH/TC1/D5/0190 [225]

On Page: 938 Line: 31450 Section: fseek()

In the ERRORS section, at the end of the section, add a new paragraph:

The fseek() [CX]and fseeko()[/CX] functions may fail if:

and then move the existing [ENXIO] condition at L31442-31443 out of the shall fail into this new may fail section, still with CX margin marker and shading.

[ENXIO] A request was made of a nonexistent device, or the request was outside the capabilities of the device.

Rationale: Austin Group Defect Report(s) applied: 225.
See http://austingroupbugs.net/view.php?id=225

Change Number: XSH/TC1/D5/0191 [14]

On Page: 938 Line: 31460 Section: fseek()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0192 [105]

On Page: 940 Line: 31503 Section: fsetpos()

In the DESCRIPTION section, add a new paragraph at the end of the DESCRIPTION:

The fsetpos() function shall not change the setting of errno if successful.

Rationale: Austin Group Defect Report(s) applied: 105.
See http://austingroupbugs.net/view.php?id=105

Change Number: XSH/TC1/D5/0193 [79]

On Page: 940 Line: 31523 Section: fsetpos()

In the ERRORS section [EIO], change from:

... the process is neither ignoring nor blocking SIGTTOU, and the process group of the process is orphaned.

to:

... the calling thread is not blocking SIGTTOU, the process is not ignoring SIGTTOU, and the process group of the process is orphaned.

Rationale: Austin Group Defect Report(s) applied: 79.
See http://austingroupbugs.net/view.php?id=79

Change Number: XSH/TC1/D5/0194 [225]

5875 On Page: 941 Line: 31531 Section: fsetpos()
5876
5877 In the ERRORS section, at the end of the section, add a new paragraph:
5878
5879 The fsetpos() function may fail if:
5880
5881 and then move the existing [ENXIO] condition at L31527-31528 out of the
5882 shall fail into the new may fail section, still with CX margin marker
5883 and shading.
5884
5885 *Rationale:* Austin Group Defect Report(s) applied: 225.
5886 See http://austingroupbugs.net/view.php?id=225
5887
5888
5889 **Change Number: XSH/TC1/D5/0195** [450]
5890
5891
5892 On Page: 941 Line: 31531 Section: fsetpos()
5893
5894 In the ERRORS section, change from:
5895
5896 [ESPIPE] The file descriptor underlying stream is associated with a pipe
5897 or FIFO.
5898
5899 to:
5900
5901 [ESPIPE] The file descriptor underlying stream is associated with a pipe,
5902 FIFO or socket.
5903
5904 *Rationale:* Austin Group Defect Report(s) applied: 450.
5905 See http://austingroupbugs.net/view.php?id=450
5906
5907
5908 **Change Number: XSH/TC1/D5/0196** [450]
5909
5910
5911 On Page: 941 Line: 31547 Section: fsetpos()
5912
5913 In the CHANGE HISTORY section, delete:
5914
5915 An additional [ESPIPE] error condition is added for sockets.
5916
5917 *Rationale:* Austin Group Defect Report(s) applied: 450.
5918 See http://austingroupbugs.net/view.php?id=450
5919
5920
5921 **Change Number: XSH/TC1/D5/0197** [14]
5922
5923
5924 On Page: 941 Line: 31541 Section: fsetpos()
5925
5926 In the SEE ALSO section, add a reference to XSH Section 2.5.
5927
5928 *Rationale:* Austin Group Defect Report(s) applied: 14.
5929 See http://austingroupbugs.net/view.php?id=14
5930 This is an editorial improvement
5931
5932
5933 **Change Number: XSH/TC1/D5/0198** [461]
5934
5935
5936 On Page: 946 Line: 31694 Section: fstatat()
5937

In the DESCRIPTION section, change from:

... the current working directory is used ...

to:

... the current working directory shall be used ...

Rationale: Austin Group Defect Report(s) applied: 461.
See http://austingroupbugs.net/view.php?id=461

Change Number: XSH/TC1/D5/0199 [324]

On Page: 946 Line: 31709 Section: fstatat()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of the path prefix is not a directory, ...

to:

A component of the path prefix names an existing file that is neither a directory nor a symbolic link to a directory, ...

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0200 [278]

On Page: 946 Line: 31720 fstatat()

In the ERRORS section, add (after the [EBADF] error):

[ENOTDIR] The path argument is not an absolute path and fd is a file descriptor associated with a non-directory file.

Rationale: Austin Group Defect Report(s) applied: 278.
See http://austingroupbugs.net/view.php?id=278

Change Number: XSH/TC1/D5/0201 [278]

On Page: 947 Line: 31731 Section: fstatat()

In the ERRORS section, delete:

[ENOTDIR] The path argument is not an absolute path and fd is neither AT_FDCWD nor a file descriptor associated with a directory.

Rationale: Austin Group Defect Report(s) applied: 278.
See http://austingroupbugs.net/view.php?id=278

Change Number: XSH/TC1/D5/0202 [291]

On Page: 948 Line: 31794 Section: fstatat()

In the EXAMPLES section, change from:

filename

to:

pathname

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XSH/TC1/D5/0203 [324]

On Page: 951 Line: 31896 Section: fstatvfs()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of the path prefix is not a directory, ...

to:

A component of the path prefix names an existing file that is neither
a directory nor a symbolic link to a directory, ...

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the
standard.

Change Number: XSH/TC1/D5/0204 [105]

On Page: 956 Line: 32041 Section: ftell()

In the DESCRIPTION, change from:

[CX]The ftello() function shall be equivalent to ftell(), except that
the return value is of type off_t.[/CX]

to:

The ftell() function shall not change the setting of errno if successful.

[CX]The ftello() function shall be equivalent to ftell(), except that
the return value is of type off_t and the ftello() function may change
the setting of errno if successful.[/CX]

Rationale: Austin Group Defect Report(s) applied: 105.
See http://austingroupbugs.net/view.php?id=105

Change Number: XSH/TC1/D5/0205 [421]

On Page: 956 Line: 32045 Section: ftell()

In the RETURN VALUE section, change from:

Otherwise, ftell() and ftello() shall return -1, cast to long
and off_t respectively, ...

to:

Otherwise, ftell() and ftello() shall return -1, ...

Rationale: Austin Group Defect Report(s) applied: 421.
See http://austingroupbugs.net/view.php?id=421

Change Number: XSH/TC1/D5/0206 [122]

On Page: 956 Line: 32054 Section: ftell()

In the ERRORS section, change from:

[ESPIPE] The file descriptor underlying stream is associated with a pipe
or FIFO.

to:

[ESPIPE] The file descriptor underlying stream is associated with a pipe,
FIFO, or socket.

Rationale: Austin Group Defect Report(s) applied: 122.
See http://austingroupbugs.net/view.php?id=122

Change Number: XSH/TC1/D5/0207 [122]

On Page: 956 Line: 32055 Section: ftell()

In the ERRORS section, delete:

The ftell() function may fail if:

CX [ESPIPE] The file descriptor underlying stream is associated with
a socket.

Rationale: Austin Group Defect Report(s) applied: 122.
See http://austingroupbugs.net/view.php?id=122

Change Number: XSH/TC1/D5/0208 [14]

On Page: 956 Line: 32066 Section: ftell()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0209 [343]

On Page: 958 Line: 32087 Section: ftok()

In the DESCRIPTION section, change from:

... existing file that the process is able to stat()

to:

... existing file that the process is able to stat(), with the exception that if stat() would fail with [EOVERFLOW] due to file size, ftok() shall still succeed.

Rationale: Austin Group Defect Report(s) applied: 343.
See http://austingroupbugs.net/view.php?id=343

Change Number: XSH/TC1/D5/0210 [366]

On Page: 958 Line: 32089 Section: ftok()

In the DESCRIPTION section, change from:

... and return different key values when called with different id ...

to:

... and should return different key values when called with different id ...

Rationale: Austin Group Defect Report(s) applied: 366.
See http://austingroupbugs.net/view.php?id=366

Change Number: XSH/TC1/D5/0211 [343]

On Page: 958 Line: 32100 Section: ftok()

In the ERRORS section, after L32100, add a new paragraph:

[EIO] An error occurred while reading from the file system.

Rationale: Austin Group Defect Report(s) applied: 343.
See http://austingroupbugs.net/view.php?id=343

Change Number: XSH/TC1/D5/0212 [324]

On Page: 958 Line: 32106 Section: ftok()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of the path prefix is not a directory, ...

to:

A component of the path prefix names an existing file that is neither a directory nor a symbolic link to a directory, ...

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0213 [366]

On Page: 959 Line: 32119-32128 Section: ftok()

In the EXAMPLES section delete L32119-32128 [EXAMPLES] (the body of the first example, and the title of the second), leaving just the second example under the title "Getting an IPC Key".

Rationale: Austin Group Defect Report(s) applied: 366.
See http://austingroupbugs.net/view.php?id=366

Change Number: XSH/TC1/D5/0214 [366]

On Page: 959 Line: 32129 Section: ftok()

In the EXAMPLES section, on L32129 delete the word "unique".

Rationale: Austin Group Defect Report(s) applied: 366.
See http://austingroupbugs.net/view.php?id=366

Change Number: XSH/TC1/D5/0215 [366]

On Page: 959 Line: 32139 Section: ftok()

In the APPLICATION USAGE section add:

Applications should not assume that the resulting key value is unique.

Rationale: Austin Group Defect Report(s) applied: 366.
See http://austingroupbugs.net/view.php?id=366

Change Number: XSH/TC1/D5/0216 [366]

On Page: 959 Line: 32143 Section: ftok()

In the FUTURE DIRECTIONS section, change from:

None.

to:

Future versions of this standard may add new interfaces to provide unique keys.

Rationale: Austin Group Defect Report(s) applied: 366.
See http://austingroupbugs.net/view.php?id=366
No current implementation is known to provide unique keys.

Change Number: XSH/TC1/D5/0217 [403]

On Page: 964 Line: 32261 Section: ftw()

In the DESCRIPTION section, change from:

FTW_F For a file.

to:

FTW_F For a non-directory file.

Rationale: Austin Group Defect Report(s) applied: 403.
See http://austingroupbugs.net/view.php?id=403

Change Number: XSH/TC1/D5/0218 [324]

On Page: 965 Line: 32299 Section: ftw()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of path is not a directory.

to:

A component of path names an existing file that is neither a directory nor a symbolic link to a directory.

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0219 [361]

On Page: 966 Line: 32351 Section: ftw()

In the CHANGE HISTORY section, change "ia" to "is".

Rationale: Austin Group Defect Report(s) applied: 361.
See http://austingroupbugs.net/view.php?id=361
Editorial correction.

Change Number: XSH/TC1/D5/0220 [63,428]

On Page: 969 Line: 32426-32430 Section: futimens()

In the ERRORS section, change from:

[EPERM] The times argument is not a null pointer, does not have both tv_nsec fields set to UTIME_NOW, does not have both tv_nsec fields set to UTIME_OMIT, the calling process' effective user ID has write access to the file but does not match the owner of the file, and the calling process does not have appropriate privileges.

to:

[EPERM] The times argument is not a null pointer, does not have both tv_nsec fields set to UTIME_NOW, does not have both tv_nsec fields set to UTIME_OMIT, the calling process effective user ID does not match the owner of the file, and the calling process does not have appropriate privileges.

Rationale: Austin Group Defect Report(s) applied: 63,428.
See http://austingroupbugs.net/view.php?id=63
See http://austingroupbugs.net/view.php?id=428

Change Number: XSH/TC1/D5/0221 [278]

On Page: 969 Line: 32438 Section: futimens()

In the ERRORS section, add (after the [EBADF] error):

[ENOTDIR] The path argument is not an absolute path and fd is a file descriptor associated with a non-directory file.

Rationale: Austin Group Defect Report(s) applied: 278.
See http://austingroupbugs.net/view.php?id=278

Change Number: XSH/TC1/D5/0222 [324]

On Page: 970 Line: 32446 Section: futimens()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of the path prefix is not a directory, ...

to:

A component of the path prefix names an existing file that is neither a directory nor a symbolic link to a directory, ...

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0223 [306]

On Page: 970 Line: 32457 Section: futimens()

After:

The utimensat() function may fail if:

add:

[EINVAL] The value of the flag argument is not valid.

Rationale: Austin Group Defect Report(s) applied: 306.
See http://austingroupbugs.net/view.php?id=306

Change Number: XSH/TC1/D5/0224 [278]

On Page: 970 Line: 32457 Section: futimens()

In the ERRORS section, delete:

[ENOTDIR] The path argument is not an absolute path and fd is neither AT_FDCWD nor a file descriptor associated with a directory.

Rationale: Austin Group Defect Report(s) applied: 278.
See http://austingroupbugs.net/view.php?id=278

Change Number: XSH/TC1/D5/0225 [272]

On Page: 972 Line: 32521 Section: fwide()

Insert before L32521 (CX shaded):

[CX]The fwide() function shall not change the setting of errno if successful.[/CX]

Rationale: Austin Group Defect Report(s) applied: 272.
See http://austingroupbugs.net/view.php?id=272

Change Number: XSH/TC1/D5/0226 [302]

On Page: 973 Line: 32590 Section: fwprintf()

change from:

... the locale of the process ...

to:

... the current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when per-thread locales were added.

Change Number: XSH/TC1/D5/0227 [14]

On Page: 979 Line: 32842 Section: fwprintf()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0228 [14]

On Page: 981 Line: 32908 Section: fwrite()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0229 [302]

On Page: 983 Line: 32955 Section: fwscanf()

change from:

... the locale of the process ...

to:

... the current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0230 [14]

On Page: 989 Line: 33199 Section: fwscanf()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0231 [14]

On Page: 992 Line: 33301 Section: getc()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0232 [395]

On Page: 993 Line: 33321 Section: getc_unlocked()

In the DESCRIPTION section, add a new paragraph at the end of the section:

If getc_unlocked or putc_unlocked are implemented as macros they may
evaluate stream more than once, so the stream argument should never be
an expression with side-effects.

Rationale: Austin Group Defect Report(s) applied: 395.
See http://austingroupbugs.net/view.php?id=395

Change Number: XSH/TC1/D5/0233 [395]

On Page: 993 Line: 33331 Section: getc_unlocked()

In the APPLICATION USAGE section, change from:

putc_unlocked(*f++)

to:

putc_unlocked(c,*f++)

Rationale: Austin Group Defect Report(s) applied: 395.
See http://austingroupbugs.net/view.php?id=395

Change Number: XSH/TC1/D5/0234 [395]

On Page: 994 Line: 33375 Section: getc_unlocked()

In the SEE ALSO section, add a reference to flockfile().

Rationale: Austin Group Defect Report(s) applied: 395.
See http://austingroupbugs.net/view.php?id=395

Change Number: XSH/TC1/D5/0235 [14]

On Page: 994 Line: 33375 Section: getc_unlocked()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0236 [14]

On Page: 995 Line: 33412 Section: getchar()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0237 [14]

On Page: 1006 Line: 33803 Section: getdelim()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0238 [75,428]

On Page: 1008 Line: 33853 Section: getenv()

In the DESCRIPTION section, change from:

The string pointed to may be overwritten by a subsequent call to getenv(), [CX]setenv(), unsetenv(),[/CX] [XSI]or putenv()[/XSI] but shall not be overwritten by a call to any other function in this volume of POSIX.1-200x.

to:

[CX]The returned string pointer might be invalidated or[/CX] the string content might be overwritten by a subsequent call to getenv(), [CX]setenv(), unsetenv(),[/CX] [XSI]or putenv()[/XSI] but they shall not be affected by a call to any other function in this volume of POSIX.1-200x.

Rationale: Austin Group Defect Report(s) applied: 75,428.
See http://austingroupbugs.net/view.php?id=75
See http://austingroupbugs.net/view.php?id=428

Change Number: XSH/TC1/D5/0239 [167]

On Page: 1008 Line: 33856 Section: getenv()

In the DESCRIPTION section, remove the text:

If the application modifies environ or the pointers to which it points, the behavior of getenv() is undefined.

Rationale: Austin Group Defect Report(s) applied: 167.
See http://austingroupbugs.net/view.php?id=167
The text relating to undefined behavior on modification of environ is removed as this is now covered by general text added in XBD Section 8.1.

Change Number: XSH/TC1/D5/0240 [167]

On Page: 1008 Line: 33880 Section: getenv()

In the RATIONALE section, change from:

Conforming applications are required not to modify environ directly, but to use only the functions described here to manipulate the process environment as an abstract object. Thus, the implementation of the environment access functions has complete control over the data structure used to represent the environment (subject to the requirement that environ be maintained as a list of strings with embedded <equals-sign> characters for applications that wish to scan the environment). This constraint allows the implementation to properly manage the memory it allocates, either by using allocated storage for all variables (copying them on the first invocation of setenv() or unsetenv()), or keeping track of which strings are currently in allocated space and which are not, via a separate table or some other means. This enables the implementation to free any allocated space used by strings (and perhaps the pointers to them) stored in environ when unsetenv() is called. A C runtime start-up procedure (that which invokes main() and perhaps initializes environ) can also initialize a flag indicating that none of the environment has yet been copied to allocated storage, or that the separate table has

not yet been initialized.

In fact, for higher performance of getenv(), the implementation could also maintain a separate copy of the environment in a data structure that could be searched much more quickly (such as an indexed hash table, or a binary tree), and update both it and the linear list at environ when setenv() or unsetenv() is invoked.

to:

Conforming applications are required not to directly modify the pointers to which environ points, but to use only the setenv(), unsetenv(), and putenv() functions, or assignment to environ itself, to manipulate the process environment. This constraint allows the implementation to properly manage the memory it allocates. This enables the implementation to free any space it has allocated to strings (and perhaps the pointers to them) stored in environ when unsetenv() is called. A C runtime start-up procedure (that which invokes main() and perhaps initializes environ) can also initialize a flag indicating that none of the environment has yet been copied to allocated storage, or that the separate table has not yet been initialized. If the application switches to a complete new environment by assigning a new value to environ, this can be detected by getenv(), setenv(), unsetenv(), or putenv() and the implementation can at that point reinitialize based on the new environment. (This may include copying the environment strings into a new array and assigning environ to point to it.)

In fact, for higher performance of getenv(), implementations that do not provide putenv() could also maintain a separate copy of the environment in a data structure that could be searched much more quickly (such as an indexed hash table, or a binary tree), and update both it and the linear list at environ when setenv() or unsetenv() is invoked. On implementations that do provide putenv(), such a copy might still be worthwhile but would need to allow for the fact that applications can directly modify the content of environment strings added with putenv(). For example, if an environment string found by searching the copy is one that was added using putenv(), the implementation would need to check that the string in environ still has the same name (and value, if the copy includes values), and whenever searching the copy produces no match the implementation would then need to search each environment string in environ that was added using putenv() in case any of them have changed their names and now match. Thus each use of putenv() to add to the environment would reduce the speed advantage of having the copy.

Rationale: Austin Group Defect Report(s) applied: 167.
See http://austingroupbugs.net/view.php?id=167

Change Number: XSH/TC1/D5/0241 [75]

On Page: 1014 Line: 34021 Section: getgrgid()

In the RETURN VALUE section, change from:

The return value may point to a static area which is overwritten by a subsequent call to getgrent(), getgrgid(), or getgrnam().

to:

The application shall not modify the structure to which the return value points, nor any storage areas pointed to by pointers within the structure. The returned pointer, and pointers within the structure,

might be invalidated or the structure or the storage areas might be overwritten by a subsequent call to getgrent(), getgrgid(), or getgrnam().

Rationale: Austin Group Defect Report(s) applied: 75.
See http://austingroupbugs.net/view.php?id=75

Change Number: XSH/TC1/D5/0242 [75]

On Page: 1018 Line: 34149 Section: getgrnam()

In the RETURN VALUE section, change from:

The return value may point to a static area which is overwritten by a subsequent call to getgrent(), getgrgid(), or getgrnam().

to:

The application shall not modify the structure to which the return value points, nor any storage areas pointed to by pointers within the structure. The returned pointer, and pointers within the structure, might be invalidated or the structure or the storage areas might be overwritten by a subsequent call to getgrent(), getgrgid(), or getgrnam().

Rationale: Austin Group Defect Report(s) applied: 75.
See http://austingroupbugs.net/view.php?id=75

Change Number: XSH/TC1/D5/0243 [172]

On Page: 1029 Line: 34464 Section: getlogin()

In the DESCRIPTION section, add a new paragraph to the end of the section:

The getlogin() and getlogin_r() functions may make use of file descriptors 0, 1, and 2 to find the controlling terminal of the current process, examining each in turn until the terminal is found. If in this case none of these three file descriptors is open to the controlling terminal, these functions may fail. The method used to find the terminal associated with a file descriptor may depend on the file descriptor being open to the actual terminal device, not /dev/tty.

Rationale: Austin Group Defect Report(s) applied: 172.
See http://austingroupbugs.net/view.php?id=172

Change Number: XSH/TC1/D5/0244 [75]

On Page: 1029 Line: 34469 Section: getlogin()

In the RETURN VALUE section, change from:

The return value from getlogin() may point to static data whose content is overwritten by each call.

to:

The application shall not modify the string returned. The returned pointer might be invalidated or the string content might be overwritten by a subsequent call to getlogin().

Rationale: Austin Group Defect Report(s) applied: 75.
See http://austingroupbugs.net/view.php?id=75

Change Number: XSH/TC1/D5/0245 [172]

On Page: 1029 Line: 34476 Section: getlogin()

In the ERRORS section, add after L34476 [ENFILE]:

[ENOTTY] None of the file descriptors 0, 1, or 2 is open to the controlling terminal of the current process.

Rationale: Austin Group Defect Report(s) applied: 172.
See http://austingroupbugs.net/view.php?id=172

Change Number: XSH/TC1/D5/0246 [284]

On Page: 1036 Line: 34710 Section: getnameinfo()

After:

The sa argument points to a socket address structure to be translated.

add:

The salen argument contains the length of the address pointed to by sa.

Rationale: Austin Group Defect Report(s) applied: 284.
See http://austingroupbugs.net/view.php?id=284

Change Number: XSH/TC1/D5/0247 [285]

On Page: 1036 Line: 34717 Section: getnameinfo()

Change "up to nodelen characters" to "up to nodelen bytes".

Rationale: Austin Group Defect Report(s) applied: 285.
See http://austingroupbugs.net/view.php?id=285

Change Number: XSH/TC1/D5/0248 [318]

On Page: 1040 Line: 34825 Section: getopt()

After "... and getopt() shall update it when it finishes with each element of argv[].", add a new sentence:

If the application sets optind to zero before calling getopt(), the behavior is unspecified.

Rationale: Austin Group Defect Report(s) applied: 318.
See http://austingroupbugs.net/view.php?id=318

Change Number: XSH/TC1/D5/0249 [460]

On Page: 1041 Line: 34860 Section: getopt()

In the ERRORS section, change from:

No errors are defined.

to:

If the application has not set the variable opterr to 0, the first character of optstring is not a <colon>, and a write error occurs while getopt() is printing a diagnostic message to stderr, then the error indicator for stderr shall be set; but getopt() shall still succeed and the value of errno after getopt() is unspecified.

Rationale: Austin Group Defect Report(s) applied: 460.
See http://austingroupbugs.net/view.php?id=460

Change Number: XSH/TC1/D5/0250 [189]

On Page: 1041 Line: 34866 Section: getopt()

In the EXAMPLES section, at L34866 (the start of the example code), insert:

```
#include <stdio.h>
#include <stdlib.h>
```

Change L34871 from:

```
int bflg, aflg, errflg;
```

to:

```
int bflg = 0, aflg = 0, errflg = 0;
```

Delete L34874 and 34875:

```
extern char *optarg;
extern int optind, optopt;
```

Change L34888-34891 from:

```
else {
    bflg++;
    bproc();
}
```

to:

```
else
    bflg++;
```

Rationale: Austin Group Defect Report(s) applied: 189.
See http://austingroupbugs.net/view.php?id=189

Change Number: XSH/TC1/D5/0251 [189]

On Page: 1042 Line: 34899 Section: getopt()

In the EXAMPLES section, reduce the indentation of L34899 by four spaces, and of L34900-34903 and L34905-34906 by eight spaces, to be consistent with the rest of the case statement.

Change L34906 from:

```
"Unrecognized option: -%c\n", optopt);
```

to:

```
"Unrecognized option: '-%c'\n", optopt);
```

Delete L34925-34955 (the entire "Checking Options and Arguments" example).

Change L34962 from:

```
char *Options = "hdbtl";
```

to:

```
const char *Options = "hdbtl";
```

Change L34964 from:

```
int dbtype, i;
char c;
```

to:

```
int dbtype, c;
```

Rationale: Austin Group Defect Report(s) applied: 189.
See http://austingroupbugs.net/view.php?id=189

Change Number: XSH/TC1/D5/0252 [189]

On Page: 1044 Line: 34997 Section: getopt()

At L34997, insert a cross-reference to XCU getopts.

Rationale: Austin Group Defect Report(s) applied: 189.
See http://austingroupbugs.net/view.php?id=189

Change Number: XSH/TC1/D5/0253 [460]

On Page: 1043 Line: 34980 Section: getopt()

In the APPLICATION USAGE section, add a new paragraph to the end of the section:

While ferror(stderr) may be used to detect failures to write a diagnostic to stderr when getopt() returns '?', the value of errno is unspecified in such a condition. Applications desiring more control over handling write failures should set opterr to 0 and independently perform output to stderr, rather than relying on getopt() to do the output.

Rationale: Austin Group Defect Report(s) applied: 460.

See http://austingroupbugs.net/view.php?id=460

Change Number: XSH/TC1/D5/0254 [464]

On Page: 1045 Line: 35015 Section: getpeername()

In the DESCRIPTION section, insert the following sentence at the beginning of the paragraph:

The address_len argument points to a socklen_t object which on input specifies the length of the supplied sockaddr structure, and on output specifies the length of the stored address.

Rationale: Austin Group Defect Report(s) applied: 464.
See http://austingroupbugs.net/view.php?id=464

Change Number: XSH/TC1/D5/0255 [75,428]

On Page: 1057 Line: 35319 Section: getpwnam()

In the RETURN VALUE section, change from:

The return value may point to a static area which is overwritten by a subsequent call to getpwent(), getpwnam(), or getpwuid().

to:

The application shall not modify the structure to which the return value points, nor any storage areas pointed to by pointers within the structure. The returned pointer, and pointers within the structure, might be invalidated or the structure or the storage areas might be overwritten by a subsequent call to getpwent(), getpwnam(), or getpwuid().

Rationale: Austin Group Defect Report(s) applied: 75,428.
See http://austingroupbugs.net/view.php?id=75
See http://austingroupbugs.net/view.php?id=428

Change Number: XSH/TC1/D5/0256 [75]

On Page: 1061 Line: 35452 Section: getpwuid()

In the RETURN VALUE section, change from:

The return value may point to a static area which is overwritten by a subsequent call to getpwent(), getpwnam(), or getpwuid().

to:

The application shall not modify the structure to which the return value points, nor any storage areas pointed to by pointers within the structure. The returned pointer, and pointers within the structure, might be invalidated or the structure or the storage areas might be overwritten by a subsequent call to getpwent(), getpwnam(), or getpwuid().

Rationale: Austin Group Defect Report(s) applied: 75.
See http://austingroupbugs.net/view.php?id=75

Change Number: XSH/TC1/D5/0257 [14]

On Page: 1070 Line: 35775 Section: gets()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0258 [421]

On Page: 1073 Line: 35804 Section: getsid()

In the RETURN VALUE section, change from:

Otherwise, it shall return (pid_t)-1 and ...

to:

Otherwise, it shall return -1 and ...

Rationale: Austin Group Defect Report(s) applied: 421.
See http://austingroupbugs.net/view.php?id=421

Change Number: XSH/TC1/D5/0259 [464]

On Page: 1074 Line: 35838 Section: getsockname()

In the DESCRIPTION section, insert the following sentence at the beginning of the paragraph:

The address_len argument points to a socklen_t object which on input specifies the length of the supplied sockaddr structure, and on output specifies the length of the stored address.

Rationale: Austin Group Defect Report(s) applied: 464.
See http://austingroupbugs.net/view.php?id=464

Change Number: XSH/TC1/D5/0260 [196]

On Page: 1078 Line: 35941 Section: getsubopt()

In the DESCRIPTION section, add to the end of the paragraph:

The getsubopt() function shall not modify the keylistp vector.

Rationale: Austin Group Defect Report(s) applied: 196.
See http://austingroupbugs.net/view.php?id=196

Change Number: XSH/TC1/D5/0261 [196]

On Page: 1079 Line: 35966 Section: getsubopt()

In the EXAMPLES section, after L35966, insert another line:

```
#include <unistd.h>
```

Rationale: Austin Group Defect Report(s) applied: 196.
See http://austingroupbugs.net/view.php?id=196
This is part of consolidation of the example into one code fragment.

Change Number: XSH/TC1/D5/0262 [196]

On Page: 1079 Line: 36003 Section: getsubopt()

In the EXAMPLES section, change from:

```
while (*subopts != '\0')
    switch(getsubopt(&subopts, mount_opts, &value))
```

to:

```
while (*subopts != '\0')
{
    char *saved = subopts;
    switch(getsubopt(&subopts, (char **)mount_opts, &value))
```

Rationale: Austin Group Defect Report(s) applied: 196.
See http://austingroupbugs.net/view.php?id=196
This is part of consolidation of the example into one code fragment.

Change Number: XSH/TC1/D5/0263 [196]

On Page: 1079 Line: 36004-36026 Section: getsubopt()

In the EXAMPLES section, adjust the indentation of L36004-36026 to reflect the new code block.

Rationale: Austin Group Defect Report(s) applied: 196.
See http://austingroupbugs.net/view.php?id=196
This is part of consolidation of the example into one code fragment.

Change Number: XSH/TC1/D5/0264 [196]

On Page: 1080 Line: 36024 Section: getsubopt()

In the EXAMPLES section, change from:

```
printf("Unknown suboption `%s'\n", value);
break;
```

to:

```
printf("Unknown suboption `%s'\n", saved);
abort();
```

Rationale: Austin Group Defect Report(s) applied: 196.
See http://austingroupbugs.net/view.php?id=196
This is part of consolidation of the example into one code
fragment.

Change Number: XSH/TC1/D5/0265 [196]

On Page: 1080 Line: 36026 Section: getsubopt()

In the EXAMPLES section, after L36026, insert a new line:

}

Rationale: Austin Group Defect Report(s) applied: 196.
See http://austingroupbugs.net/view.php?id=196
This is part of consolidation of the example into one code
fragment.

Change Number: XSH/TC1/D5/0266 [196]

On Page: 1080 Line: 36033 Section: getsubopt()

In the EXAMPLES section, after L36033, insert a new paragraph describing
the previous example:

If the above example is invoked with:

program -o ro,rsize=512

then after option parsing, the variable do_all will be 0, type will be a
null pointer, read_size will be 512, write_size will be 0, and read_only
will be 1. If it is invoked with:

program -o oops

it will print "Unknown suboption `oops'" before aborting.

Rationale: Austin Group Defect Report(s) applied: 196.
See http://austingroupbugs.net/view.php?id=196
This is part of consolidation of the example into one code
fragment.

Change Number: XSH/TC1/D5/0267 [196]

On Page: 1080 Line: 36055 Section: getsubopt()

In the APPLICATION USAGE section, change from:

None.

to:

The value of *valuep when getsubopt() returns -1 is unspecified.
Historical implementations provide various incompatible extensions to
allow an application to access the suboption text that was not found in
the keylistp array.

Rationale: Austin Group Defect Report(s) applied: 196.
See http://austingroupbugs.net/view.php?id=196

Change Number: XSH/TC1/D5/0268 [196]

On Page: 1080 Line: 36034- Section: getsubopt()

In the EXAMPLES section, move L36034-36036 (the subheading Parsing Suboptions and following paragraph) prior to the code example at P1079 L35965; then delete L36037-36053 (the second code fragment).

Rationale: Austin Group Defect Report(s) applied: 196.
See http://austingroupbugs.net/view.php?id=196
This is part of consolidation of the example into one code fragment.

Change Number: XSH/TC1/D5/0269 [196]

On Page: 1081 Line: 36057 Section: getsubopt()

In the RATIONALE section, change from:

None.

to:

The keylistp argument of getsubopt() is typed as char * const * to match historical practice. However, the standard is clear that implementations will not modify either the array or the strings contained in the array, as if the argument had been typed const char * const *.

Rationale: Austin Group Defect Report(s) applied: 196.
See http://austingroupbugs.net/view.php?id=196

Change Number: XSH/TC1/D5/0270 [14]

On Page: 1085 Line: 36186 Section: getwc()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0271 [14]

On Page: 1086 Line: 36218 Section: getwchar()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0272 [96]

On Page: 1093 Line: 36478 Section: grantpt()

In the RATIONALE section, change from:

None.

to:

See RATIONALE for posix_openpt().

On L36482, in the SEE ALSO section, add posix_openpt().

Rationale: Austin Group Defect Report(s) applied: 96.
See http://austingroupbugs.net/view.php?id=96
Add reference to posix_openpt() to the RATIONALE and SEE
ALSO sections.

Change Number: XSH/TC1/D5/0273 [68]

On Page: 1098 Line: 36646 Section: hypot()

In the RETURN VALUE section, change the MX shading to MXX for:

If both arguments are subnormal and the correct result is subnormal,
a range error may occur and the correct result is returned.

and change from "is returned" to "shall be returned".

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0274 [302]

On Page: 1135 Line: 37918 Section: isalnum()

In the DESCRIPTION section, change from:

... current locale of the process ...

to:

... current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0275 [283]

On Page: 1135 Line: 37922 Section: isalnum()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to isalnum_l() is the special locale object LC_GLOBAL_LOCALE or is not a valid locale object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0276 [283]

On Page: 1135 Line: 37927 Section: isalnum()

In the ERRORS section, change from:

The isalnum_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0277 [302]

On Page: 1137 Line: 37961 Section: isalpha()

In the DESCRIPTION section, change from:

... current locale of the process ...

to:

... current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when per-thread locales were added.

Change Number: XSH/TC1/D5/0278 [283]

On Page: 1137 Line: 37965 Section: isalpha()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to isalpha_l() is the special locale object LC_GLOBAL_LOCALE or is not a valid locale object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0279 [283]

On Page: 1137 Line: 37970 Section: isalpha()

In the ERRORS section, change from:

The isalpha_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0280 [302]

On Page: 1142 Line: 38097 Section: isblank()

In the DESCRIPTION section, change from:

... current locale of the process ...

to:

... current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0281 [283]

On Page: 1142 Line: 38101 Section: isblank()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to isblank_l()
is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0282 [283]

On Page: 1142 Line: 38106 Section: isblank()

In the ERRORS section, change from:

The isblank_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0283 [302]

On Page: 1143 Line: 38138 Section: iscntrl()

In the DESCRIPTION section, change from:

... current locale of the process ...

to:

... current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0284 [283]

On Page: 1143 Line: 38142 Section: iscntrl()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to iscntrl_l()
is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0285 [283]

On Page: 1143 Line: 38147 Section: iscntrl()

In the ERRORS section, change from:

The iscntrl_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0286 [302]

On Page: 1145 Line: 38181 Section: isdigit()

In the DESCRIPTION section, change from:

... current locale of the process ...

to:

... current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0287 [283]

On Page: 1145 Line: 38185 Section: isdigit()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to isdigit_l()
is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0288 [283]

On Page: 1145 Line: 38190 Section: isdigit()

In the ERRORS section, change from:

The isdigit_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0289 [302]

On Page: 1148 Line: 38254 Section: isgraph()

In the DESCRIPTION section, change from:

... current locale of the process ...

to:

... current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0290 [283]

On Page: 1148 Line: 38258 Section: isgraph()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to isgraph_l()
is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0291 [283]

On Page: 1148 Line: 38263 Section: isgraph()

In the ERRORS section, change from:

The isgraph_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0292 [302]

On Page: 1156 Line: 38515 Section: islower()

In the DESCRIPTION section, change from:

... current locale of the process ...

to:

... current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0293 [283]

On Page: 1156 Line: 38519 Section: islower()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to islower_l() is the special locale object LC_GLOBAL_LOCALE or is not a valid locale object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0294 [283]

On Page: 1156 Line: 38524 Section: islower()

In the ERRORS section, change from:

The islower_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0295 [302]

On Page: 1156 Line: 38530 Section islower()

In the EXAMPLES section, change from:

... the locale of the user ...

to:

... the current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when per-thread locales were added.

Change Number: XSH/TC1/D5/0296 [304]

On Page: 1156 Line: 38539,38545,38558,38564 Section: islower()

In the EXAMPLES section, on L38539 and L38558 change from:

```
char c;
```

to:

```
unsigned char c;
```

In the EXAMPLES section, on L38545 and L38564 change from:

(char)

to

(unsigned char)

Rationale: Austin Group Defect Report(s) applied: 304.
See http://austingroupbugs.net/view.php?id=304

Change Number: XSH/TC1/D5/0297 [302]

On Page: 1160 Line: 38666 Section: isprint()

In the DESCRIPTION section, change from:

... current locale of the process ...

to:

... current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0298 [283]

On Page: 1160 Line: 38670 Section: isprint()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to isprint_l()
is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0299 [283]

On Page: 1160 Line: 38675 Section: isprint()

In the ERRORS section, change from:

The isprint_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0300 [302]

On Page: 1162 Line: 38709 Section: ispunct()

In the DESCRIPTION section, change from:

... current locale of the process ...

to:

... current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0301 [283]

On Page: 1162 Line: 38713 Section: ispunct()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to ispunct_l()
is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0302 [283]

On Page: 1162 Line: 38718 Section: ispunct()

In the ERRORS section, change from:

The ispunct_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0303 [302]

On Page: 1164 Line: 38752 Section: isspace()

In the DESCRIPTION section, change from:

... current locale of the process ...

to:

... current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0304 [283]

On Page: 1164 Line: 38756 Section: isspace()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to isspace_l()
is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0305 [283]

On Page: 1164 Line: 38761 Section: isspace()

In the ERRORS section, change from:

The isspace_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0306 [302]

On Page: 1167 Line: 38833 Section: isupper()

In the DESCRIPTION section, change from:

... current locale of the process ...

to:

... current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0307 [283]

On Page: 1167 Line: 38837 Section: isupper()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to isupper_l()
is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0308 [283]

On Page: 1167 Line: 38842 Section: isupper()

In the ERRORS section, change from:

The isupper_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0309 [302]

On Page: 1169 Line: 38876 Section: iswalnum()

In the DESCRIPTION section, change from:

... current locale of the process ...

to:

... current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0310 [283]

On Page: 1169 Line: 38880 Section: iswalnum()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to iswalnum_l()
is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0311 [283]

On Page: 1169 Line: 38885 Section: iswalnum()

In the ERRORS section, change from:

The iswalnum_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0312 [302]

On Page: 1171 Line: 38924 Section: iswalpha()

In the DESCRIPTION section, change from:

... current locale of the process ...

to:

... current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0313 [283]

On Page: 1171 Line: 38928 Section: iswalpha()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to iswalpha_l()
is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0314 [283]

On Page: 1171 Line: 38933 Section: iswalpha()

In the ERRORS section, change from:

The iswalpha_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0315 [302]

On Page: 1173 Line: 38972 Section: iswblank()

In the DESCRIPTION section, change from:

... current locale of the process ...

to:

... current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0316 [283]

On Page: 1173 Line: 38976 Section: iswblank()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to iswblank_l()
is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0317 [283]

On Page: 1173 Line: 38981 Section: iswblank()

In the ERRORS section, change from:

The iswblank_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0318 [302]

On Page: 1174 Line: 39013 Section: iswcntrl()

In the DESCRIPTION section, change from:

... current locale of the process ...

to:

... current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0319 [283]

On Page: 1174 Line: 39017 Section: iswcntrl()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to iswcntrl_l()
is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0320 [283]

On Page: 1174 Line: 39022 Section: iswcntrl()

In the ERRORS section, change from:

The iswcntrl_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0321 [283]

On Page: 1176 Line: 39067 Section: iswctype()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to iswctype_l() is the special locale object LC_GLOBAL_LOCALE or is not a valid locale object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0322 [283]

On Page: 1176 Line: 39072 Section: iswctype()

In the ERRORS section, change from:

The iswctype_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0323 [302]

On Page: 1178 Line: 39147 Section: iswdigit()

In the DESCRIPTION section, change from:

... current locale of the process ...

to:

... current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when per-thread locales were added.

Change Number: XSH/TC1/D5/0324 [283]

On Page: 1178 Line: 39151 Section: iswdigit()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to iswdigit_l() is the special locale object LC_GLOBAL_LOCALE or is not a valid locale object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0325 [283]

On Page: 1178 Line: 39156 Section: iswdigit()

In the ERRORS section, change from:

The iswdigit_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0326 [302]

On Page: 1180 Line: 39195 Section: iswgraph()

In the DESCRIPTION section, change from:

... current locale of the process ...

to:

... current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0327 [283]

On Page: 1180 Line: 39199 Section: iswgraph()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to iswgraph_l()
is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0328 [283]

On Page: 1180 Line: 39204 Section: iswgraph()

In the ERRORS section, change from:

The iswgraph_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0329 [302]

On Page: 1182 Line: 39243 Section: iswlower()

In the DESCRIPTION section, change from:

... current locale of the process ...

to:

... current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0330 [283]

On Page: 1182 Line: 39247 Section: iswlower()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to iswlower_l()
is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0331 [283]

On Page: 1182 Line: 39252 Section: iswlower()

In the ERRORS section, change from:

The iswlower_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0332 [302]

On Page: 1184 Line: 39291 Section: iswprint()

In the DESCRIPTION section, change from:

... current locale of the process ...

to:

... current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0333 [283]

On Page: 1184 Line: 39295 Section: iswprint()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to iswprint_l()
is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0334 [283]

On Page: 1184 Line: 39300 Section: iswprint()

In the ERRORS section, change from:

The iswprint_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0335 [302]

On Page: 1186 Line: 39339 Section: iswpunct()

In the DESCRIPTION section, change from:

... current locale of the process ...

to:

... current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0336 [283]

On Page: 1186 Line: 39343 Section: iswpunct()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to iswpunct_l()
is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0337 [283]

On Page: 1186 Line: 39348 Section: iswpunct()

In the ERRORS section, change from:

The iswpunct_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0338 [302]

On Page: 1188 Line: 39387 Section: iswspace()

In the DESCRIPTION section, change from:

... current locale of the process ...

to:

... current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0339 [283]

On Page: 1188 Line: 39391 Section: iswspace()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to iswspace_l() is the special locale object LC_GLOBAL_LOCALE or is not a valid locale object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0340 [283]

On Page: 1188 Line: 39396 Section: iswspace()

In the ERRORS section, change from:

The iswspace_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0341 [302]

On Page: 1190 Line: 39435 Section: iswupper()

In the DESCRIPTION section, change from:

... current locale of the process ...

to:

... current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when per-thread locales were added.

Change Number: XSH/TC1/D5/0342 [283]

On Page: 1190 Line: 39439 Section: iswupper()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to iswupper_l() is the special locale object LC_GLOBAL_LOCALE or is not a valid locale object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0343 [283]

On Page: 1190 Line: 39444 Section: iswupper()

In the ERRORS section, change from:

The iswupper_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0344 [302]

On Page: 1192 Line: 39483 Section: iswxdigit()

In the DESCRIPTION section, change from:

... current locale of the process ...

to:

... current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0345 [283]

On Page: 1192 Line: 39487 Section: iswxdigit()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to iswxdigit_l()
is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0346 [283]

On Page: 1192 Line: 39492 Section: iswxdigit()

In the ERRORS section, change from:

The iswxdigit_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0347 [302]

On Page: 1194 Line: 39531 Section: isxdigit()

In the DESCRIPTION section, change from:

... current locale of the process ...

to:

... current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0348 [283]

On Page: 1194 Line: 39535 Section: isxdigit()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to isxdigit_l()
is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0349 [283]

On Page: 1194 Line: 39540 Section: isxdigit()

In the ERRORS section, change from:

The isxdigit_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0350 [68]

On Page: 1196 Line: 39582 Section: j0()

In the RETURN VALUE section, add MXX shading to:

If x is NaN, a NaN shall be returned.

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0351 [324]

On Page: 1206 Line: 39847 Section: lchown()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of the path prefix is not a directory, ...

to:

A component of the path prefix names an existing file that is neither a directory nor a symbolic link to a directory, ...

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0352 [68]

On Page: 1210 Line: 39934 Section: ldexp()

In the RETURN VALUE section, change from:

If the correct value would cause underflow, and is not representable, a range error may occur, and [MX]either 0.0 (if supported), or[/MX] an implementation-defined value shall be returned.

to:

If the correct value would cause underflow, [MXX]and is not representable[/MXX], a range error may occur, and ldexp(), ldexpf(), and ldexpl() shall return [MXX]0.0, or[/MXX] (if IEC 60559 Floating-Point is not supported) an implementation-defined value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0353 [68]

On Page: 1210 Line: 39939 Section: ldexp()

In the RETURN VALUE section, change the MX shading to MXX for:

If the correct value would cause underflow, and is representable,
a range error may occur and the correct value shall be returned.

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0354 [326]

On Page: 1216 Line: 40123 Section: link()

Change:

The linkat() function shall be equivalent to the link() function except
in the case where either path1 or path2 or both are relative paths.

to:

The linkat() function shall be equivalent to the link() function except
that symbolic links shall be handled as specified by the value of flag
(see below) and except in the case where either path1 or path2 or both
are relative paths.

Rationale: Austin Group Defect Report(s) applied: 326.
See http://austingroupbugs.net/view.php?id=326

Change Number: XSH/TC1/D5/0355 [461]

On Page: 1216 Line: 40134 Section: link()

In the DESCRIPTION section, change from:

... the current working directory is used ...

to:

... the current working directory shall be used ...

Rationale: Austin Group Defect Report(s) applied: 461.
See http://austingroupbugs.net/view.php?id=461

Change Number: XSH/TC1/D5/0356 [326]

On Page: 1216 Line: 40136 Section link()

Change:

... the behavior shall be identical to a call to link().

to:

... the behavior shall be identical to a call to link(),
except that symbolic links shall be handled as specified
by the value of flag.

Rationale: Austin Group Defect Report(s) applied: 326.
See http://austingroupbugs.net/view.php?id=326

Change Number: XSH/TC1/D5/0357 [324]

On Page: 1217 Line: 40158 Section: link()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of either path prefix is not a directory, ...

to:

A component of either path prefix names an existing file that is neither a directory nor a symbolic link to a directory, ...

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0358 [147,429]

On Page: 1217 Line: 40161 Section: link()

In the ERRORS section, add to the end of the [ENOTDIR] error (replacing the full stop):

..., or the path1 argument names an existing non-directory file and the path2 argument names a nonexistent file, contains at least one non-<slash> character, and ends with one or more trailing <slash> characters.

Rationale: Austin Group Defect Report(s) applied: 147,429.
See http://austingroupbugs.net/view.php?id=147
See http://austingroupbugs.net/view.php?id=429

Change Number: XSH/TC1/D5/0359 [277]

On Page: 1217 Line: 40172 Section: link()

In the ERRORS section, change from:

... nor a valid file descriptor open for reading.

to:

... nor a valid file descriptor open for reading or searching.

Rationale: Austin Group Defect Report(s) applied: 277.
See http://austingroupbugs.net/view.php?id=277

Change Number: XSH/TC1/D5/0360 [278]

On Page: 1217 Line: 40172 Section: link()

In the ERRORS section, add (after the [EBADF] error):

[ENOTDIR] The path1 or path2 argument is not an absolute path and fd1 or fd2, respectively, is a file descriptor associated with a non-directory file.

Rationale: Austin Group Defect Report(s) applied: 278.
See http://austingroupbugs.net/view.php?id=278

Change Number: XSH/TC1/D5/0361 [278]

On Page: 1218 Line: 40182 Section: link()

In the ERRORS section, delete:

[ENOTDIR] The path1 or path2 argument is not an absolute path and fd1 or fd2, respectively, is neither AT_FDCWD nor a file descriptor associated with a directory.

Rationale: Austin Group Defect Report(s) applied: 278.
See http://austingroupbugs.net/view.php?id=278

Change Number: XSH/TC1/D5/0362 [75]

On Page: 1235 Line: 40719 Section: localeconv()

In the RETURN VALUE section, change from:

The application shall not modify the structure pointed to by the return value which may be overwritten by a subsequent call to localeconv(). In addition, calls to setlocale() with the categories LC_ALL, LC_MONETARY, or LC_NUMERIC or calls to uselocale() which change the categories LC_MONETARY or LC_NUMERIC may overwrite the contents of the structure.

to:

The application shall not modify the structure to which the return value points, [CX]nor any storage areas pointed to by pointers within the structure. The returned pointer, and pointers within the structure, might be invalidated or[/CX] the structure [CX]or the storage areas[/CX] might be overwritten by a subsequent call to localeconv(). In addition, [CX]the returned pointer, and pointers within the structure, might be invalidated or[/CX] the structure [CX]or the storage areas[/CX] might be overwritten by subsequent calls to setlocale() with the categories LC_ALL, LC_MONETARY, or LC_NUMERIC, [CX]or by calls to uselocale() which change the categories LC_MONETARY or LC_NUMERIC[/CX].

Rationale: Austin Group Defect Report(s) applied: 75.
See http://austingroupbugs.net/view.php?id=75

Change Number: XSH/TC1/D5/0363 [291]

On Page: 1239 Line: 40849-40860 Section: localtime()

In the EXAMPLES section, change all instances of "filename" to "pathname".

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XSH/TC1/D5/0364 [68]

On Page: 1249 Line: 41185 Section: log1p()

In the RETURN VALUE section, change from:

[MX]If x is subnormal, a range error may occur and x should be returned.[/MX]

to:

[MX]If x is subnormal, a range error may occur[/MX] [MXX]and x should be returned.[/MXX]

[MX]If x is not returned, log1p(), log1pf(), and log1pl() shall return an implementation-defined value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.[/MX]

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0365 [394]

On Page: 1256 Line: 41363 Section: longjmp()

In the DESCRIPTION section, change from:

The longjmp() function shall restore the environment saved by the most recent invocation of setjmp() in the same thread, with the corresponding jmp_buf argument. If there is no such invocation, or if the function containing the invocation of setjmp() has terminated execution in the interim, or if the invocation of setjmp() was within the scope of an identifier with variably modified type and execution has left that scope in the interim, the behavior is undefined

to:

The longjmp() function shall restore the environment saved by the most recent invocation of setjmp() in the same process, with the corresponding jmp_buf argument. If the most recent invocation of setjmp() with the corresponding jmp_buf occurred in another thread, or if there is no such invocation, or if the function containing the invocation of setjmp() has terminated execution in the interim, or if the invocation of setjmp() was within the scope of an identifier with variably modified type and execution has left that scope in the interim, the behavior is undefined

Rationale: Austin Group Defect Report(s) applied: 394.
See http://austingroupbugs.net/view.php?id=394

Change Number: XSH/TC1/D5/0366 [421]

On Page: 1265 Line: 41639 Section: lseek()

In the RETURN VALUE section, change from:

Otherwise, (off_t)-1 shall be returned, ...

to:

Otherwise, -1 shall be returned, ...

Rationale: Austin Group Defect Report(s) applied: 421.
See http://austingroupbugs.net/view.php?id=421

Change Number: XSH/TC1/D5/0367 [109]

On Page: 1270 Line: 41765 Section: mblen()

In the DESCRIPTION section, add a new paragraph to the end of the section, with CX shading:

The mblen() function need not be thread-safe.

Rationale: Austin Group Defect Report(s) applied: 109.
See http://austingroupbugs.net/view.php?id=109

Change Number: XSH/TC1/D5/0368 [109,105]

On Page: 1272 Line: 41808 Section: mbrlen()

In the DESCRIPTION section, add two new paragraphs to the end of the section:

[CX]The mbrlen() function need not be thread-safe if called with a NULL ps argument.[/CX]

The mbrlen() function shall not change the setting of errno if successful.

Rationale: Austin Group Defect Report(s) applied: 109,105.
See http://austingroupbugs.net/view.php?id=109
See http://austingroupbugs.net/view.php?id=105

Change Number: XSH/TC1/D5/0369 [109,105]

On Page: 1274 Line: 41871 Section: mbrtowc()

In the DESCRIPTION section, add two new paragraphs to the end of the section:

[CX]The mbrtowc() function need not be thread-safe if called with a NULL ps argument.[/CX]

The mbrtowc() function shall not change the setting of errno if successful.

Rationale: Austin Group Defect Report(s) applied: 109,105.
See http://austingroupbugs.net/view.php?id=109
See http://austingroupbugs.net/view.php?id=105

Change Number: XSH/TC1/D5/0370 [109,105]

On Page: 1277 Line: 41989 Section: mbsnrtowcs()

In the DESCRIPTION section, add two new paragraphs to the end of the section:

[CX]The mbsnrtowcs() and mbsrtowcs() functions need not be thread-safe if called with a NULL ps argument.[/CX]

The mbsrtowcs() function shall not change the setting of errno if successful.

Rationale: Austin Group Defect Report(s) applied: 109,105.
See http://austingroupbugs.net/view.php?id=109
See http://austingroupbugs.net/view.php?id=105

Change Number: XSH/TC1/D5/0371 [195]

On Page: 1279 Line: 42044 Section: mbstowcs()

In the RETURN VALUE section, change from:

... and may set errno ...

to:

... and shall set errno ...

Rationale: Austin Group Defect Report(s) applied: 195.
See http://austingroupbugs.net/view.php?id=195

Change Number: XSH/TC1/D5/0372 [109]

On Page: 1281 Line: 42094 Section: mbtowc()

In the DESCRIPTION section, add a new paragraph to the end of the section, with CX shading:

The mbtowc() function need not be thread-safe.

Rationale: Austin Group Defect Report(s) applied: 109.
See http://austingroupbugs.net/view.php?id=109

Change Number: XSH/TC1/D5/0373 [195]

On Page: 1281 Line: 42099 Section: mbtowc()

In the RETURN VALUE section, change from:

... and may set errno ...

to:

... and shall set errno ...

Rationale: Austin Group Defect Report(s) applied: 195.
See http://austingroupbugs.net/view.php?id=195

Change Number: XSH/TC1/D5/0374 [110]

On Page: 1284 Line: 42163 Section: memchr()

In the DESCRIPTION section, change from:

The memchr() function shall locate the first occurrence of c (converted to an unsigned char) in the initial n bytes (each interpreted as unsigned char) of the object pointed to by s.

to:

The memchr() function shall locate the first occurrence of c (converted to an unsigned char) in the initial n bytes (each interpreted as unsigned char) pointed to by s.

Add to DESCRIPTION as a new paragraph:

Implementations shall behave as if they read the memory byte by byte from the beginning of the bytes pointed to by s and stop at the first occurrence of c (if it is found in the initial n bytes).

In the RETURN VALUE section change from:

The memchr() function shall return a pointer to the located byte, or a null pointer if the byte does not occur in the object.

to:

The memchr() function shall return a pointer to the located byte, or a null pointer if the byte is not found.

Rationale: Austin Group Defect Report(s) applied: 110.
See http://austingroupbugs.net/view.php?id=110

Change Number: XSH/TC1/D5/0375 [461]

On Page: 1289 Line: 42327 Section: mkdir()

In the DESCRIPTION section, change from:

... the current working directory is used ...

to:

... the current working directory shall be used ...

Rationale: Austin Group Defect Report(s) applied: 461.
See http://austingroupbugs.net/view.php?id=461

Change Number: XSH/TC1/D5/0376 [324]

On Page: 1290 Line: 42346 Section: mkdir()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of the path prefix is not a directory.

to:

A component of the path prefix names an existing file that is neither a directory nor a symbolic link to a directory.

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0377 [277]

On Page: 1290 Line: 42350 Section: mkdir()

In the ERRORS section, [EBADF] error, change from:

... nor a valid file descriptor open for reading.

to:

... nor a valid file descriptor open for reading or searching.

Rationale: Austin Group Defect Report(s) applied: 277.
See http://austingroupbugs.net/view.php?id=277

Change Number: XSH/TC1/D5/0378 [278]

On Page: 1290 Line: 42350 mkdir()

In the ERRORS section, add (after the [EBADF] error):

[ENOTDIR] The path argument is not an absolute path and fd is a file descriptor associated with a non-directory file.

Rationale: Austin Group Defect Report(s) applied: 278.
See http://austingroupbugs.net/view.php?id=278

Change Number: XSH/TC1/D5/0379 [278]

On Page: 1290 Line: 42358 Section: mkdir()

In the ERRORS section, delete:

The mkdirat() function may fail if:

[ENOTDIR] The path argument is not an absolute path and fd is neither AT_FDCWD nor a file descriptor associated with a directory.

Rationale: Austin Group Defect Report(s) applied: 278.
See http://austingroupbugs.net/view.php?id=278

Change Number: XSH/TC1/D5/0380 [291]

On Page: 1292 Line: 42420- Section: mkdtemp()

In the DESCRIPTION section, on L42420, L42426, L42428, and L42430
change "filename" to "pathname".

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XSH/TC1/D5/0381 [324]

On Page: 1293 Line: 42454 Section: mkdtemp()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of the path prefix is not a directory.

to:

A component of the path prefix names an existing file that is neither
a directory nor a symbolic link to a directory.

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the
standard.

Change Number: XSH/TC1/D5/0382 [291]

On Page: 1293 Line: 42465 Section: mkdtemp()

In the EXAMPLES section, change "Filename" to "Pathname".

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XSH/TC1/D5/0383 [461]

On Page: 1295 Line: 42523 Section: mkfifo()

In the DESCRIPTION section, change from:

... the current working directory is used ...

to:

... the current working directory shall be used ...

Rationale: Austin Group Defect Report(s) applied: 461.
See http://austingroupbugs.net/view.php?id=461

Change Number: XSH/TC1/D5/0384 [146,435]

On Page: 1296 Line: 42537 Section: mkfifo()

In the ERRORS section, change from:

[ENOENT] A component of the path prefix specified by path does not name an existing directory or path is an empty string.

to:

[ENOENT] A component of the path prefix of path does not name an existing file or path is an empty string.

[ENOENT] or [ENOTDIR]
The path argument contains at least one non-<slash> character and ends with one or more trailing <slash> characters. If path names an existing file, an [ENOENT] error shall not occur.

Rationale: Austin Group Defect Report(s) applied: 146,435.
See http://austingroupbugs.net/view.php?id=146
See http://austingroupbugs.net/view.php?id=435

Change Number: XSH/TC1/D5/0385 [324]

On Page: 1296 Line: 42541 Section: mkfifo()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of the path prefix is not a directory.

to:

A component of the path prefix names an existing file that is neither a directory nor a symbolic link to a directory.

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0386 [278]

On Page: 1296 Line: 42547 mkfifo()

In the ERRORS section, add (after the [EBADF] error):

[ENOTDIR] The path argument is not an absolute path and fd is a file descriptor associated with a non-directory file.

Rationale: Austin Group Defect Report(s) applied: 278.
See http://austingroupbugs.net/view.php?id=278

Change Number: XSH/TC1/D5/0387 [278]

On Page: 1296 Line: 42555 Section: mkfifo()

In the ERRORS section, delete:

The mkfifoat() function may fail if:

[ENOTDIR] The path argument is not an absolute path and fd is neither AT_FDCWD nor a file descriptor associated with a directory.

Rationale: Austin Group Defect Report(s) applied: 278.
See http://austingroupbugs.net/view.php?id=278

Change Number: XSH/TC1/D5/0388 [324]

On Page: 1299 Line: 42693 Section: mknod()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of the path prefix is not a directory.

to:

A component of the path prefix names an existing file that is neither a directory nor a symbolic link to a directory.

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0389 [461]

On Page: 1299 Line: 42673 Section: mknod()

In the DESCRIPTION section, change from:

... the current working directory is used ...

to:

... the current working directory shall be used ...

Rationale: Austin Group Defect Report(s) applied: 461.
See http://austingroupbugs.net/view.php?id=461

Change Number: XSH/TC1/D5/0390 [146,435]

On Page: 1299 Line: 42689 Section: mknod()

In the ERRORS section, change from:

[ENOENT] A component of the path prefix specified by path does not name an existing directory or path is an empty string.

to:

[ENOENT] A component of the path prefix of path does not name an existing file or path is an empty string.

[ENOENT] or [ENOTDIR]
The path argument contains at least one non-<slash> character and ends with one or more trailing <slash> characters. If path names an existing file, an [ENOENT] error shall not occur.

Rationale: Austin Group Defect Report(s) applied: 146,435.

See http://austingroupbugs.net/view.php?id=146
See http://austingroupbugs.net/view.php?id=435

Change Number: XSH/TC1/D5/0391 [278]

On Page: 1300 Line: 42702 mknod()

In the ERRORS section, add (after the [EBADF] error):

[ENOTDIR] The path argument is not an absolute path and fd is a file descriptor associated with a non-directory file.

Rationale: Austin Group Defect Report(s) applied: 278.
See http://austingroupbugs.net/view.php?id=278

Change Number: XSH/TC1/D5/0392 [278]

On Page: 1300 Line: 42710 Section: mknod()

In the ERRORS section, delete:

The mknodat() function may fail if:

[ENOTDIR] The path argument is not an absolute path and fd is neither AT_FDCWD nor a file descriptor associated with a directory.

Rationale: Austin Group Defect Report(s) applied: 278.
See http://austingroupbugs.net/view.php?id=278

Change Number: XSH/TC1/D5/0393 [104]

On Page: 1303 Line: 42798 Section: mktime()

In the RETURN VALUE section, change from:

... shall return the value (time_t)-1 [CX]and may set errno to indicate the error[/CX].

to:

... shall return the value (time_t)-1 [CX]and set errno to indicate the error[/CX].

On L42800, in the ERRORS section, change from:

The mktime() function may fail if ...

to:

The mktime() function shall fail if ...

Rationale: Austin Group Defect Report(s) applied: 104.
See http://austingroupbugs.net/view.php?id=104
The [EOVERFLOW] error is changed from a may fail to a shall fail.

Change Number: XSH/TC1/D5/0394 [259]

On Page: 1328 Line: 43672 Section: mq_open()

In the DESCRIPTION section (O_CREAT), after:

... the message queue mq_maxmsg and mq_msgsize attributes shall be set to the values of the corresponding members in the mq_attr structure referred to by attr.

Add:

The values of the mq_flags and mq_curmsgs members of the mq_attr structure shall be ignored.

Rationale: Austin Group Defect Report(s) applied: 259.
See http://austingroupbugs.net/view.php?id=259

Change Number: XSH/TC1/D5/0395 [345]

On Page: 1342 Line: 44077 Section: msgctl()

In the DESCRIPTION section (for IPC_SET), add:

Also, the msg_ctime timestamp shall be set to the current time, as described in XSH Section 2.7.1.

Rationale: Austin Group Defect Report(s) applied: 345.
See http://austingroupbugs.net/view.php?id=345

Change Number: XSH/TC1/D5/0396 [345]

On Page: 1344 Line: 44145 Section: msgget()

In the DESCRIPTION section, change from:

... set equal to the current time

to:

... set to the current time, as described in XSH Section 2.7.1

Rationale: Austin Group Defect Report(s) applied: 345.
See http://austingroupbugs.net/view.php?id=345

Change Number: XSH/TC1/D5/0397 [344]

On Page: 1345 Line: 44174 Section: msgget()

In the SEE ALSO section, add ftok() prior to mq_close().

Rationale: Austin Group Defect Report(s) applied: 344.
See http://austingroupbugs.net/view.php?id=344

Change Number: XSH/TC1/D5/0398 [345]

On Page: 1347 Line: 44230 Section: msgrcv()

In the DESCRIPTION section, change from:

... set equal to the current time

to:

... set to the current time, as described in XSH Section 2.7.1

Rationale: Austin Group Defect Report(s) applied: 345.
See http://austingroupbugs.net/view.php?id=345

Change Number: XSH/TC1/D5/0399 [421]

On Page: 1347 Line: 44234 Section: msgrcv()

In the RETURN VALUE section, change from:

... msgrcv() shall return (ssize_t)-1, and ...

to:

... msgrcv() shall return -1, and ...

Rationale: Austin Group Defect Report(s) applied: 421.
See http://austingroupbugs.net/view.php?id=421

Change Number: XSH/TC1/D5/0400 [345]

On Page: 1350 Line: 44329 Section: msgsnd()

In the DESCRIPTION section, change from:

... set equal to the current time

to:

... set to the current time, as described in XSH Section 2.7.1

Rationale: Austin Group Defect Report(s) applied: 345.
See http://austingroupbugs.net/view.php?id=345

Change Number: XSH/TC1/D5/0401 [359]

On Page: 1350 Line: 44342 Section: msgsnd() ERRORS

Remove the words "less than 0 or" from the description of [EINVAL].

Rationale: Austin Group Defect Report(s) applied: 359.
See http://austingroupbugs.net/view.php?id=359

Change Number: XSH/TC1/D5/0402 [346,428]

On Page: 1362 Line: 44682 Section: nearbyint()

For nearbyint(), delete P1362 L44682-44685 (DESCRIPTION), L44691-44693 (RETURN VALUE), and replace L44695-44700 (ERRORS) with:

No errors are defined.

In the RETURN VALUE section, at L44687 add:

[MX]The result shall have the same sign as x.[/MX]

In the APPLICATION USAGE section, change from:

On error, the expressions (math_errhandling & MATH_ERRNO) and (math_errhandling & MATH_ERREXCEPT) are independent of each other, but at least one of them must be non-zero.

to:

The integral value returned by these functions need not be expressible as an intmax_t. The return value should be tested before assigning it to an integer type to avoid the undefined results of an integer overflow.

Rationale: Austin Group Defect Report(s) applied: 346,428.
See http://austingroupbugs.net/view.php?id=346
See http://austingroupbugs.net/view.php?id=428

Change Number: XSH/TC1/D5/0403 [227]

On Page: 1364 Line: 44724 Section: newlocale()

In the DESCRIPTION section, change from:

It is unspecified whether the locale object pointed to by base shall be modified or freed and a new locale object created.

to:

It is unspecified whether the locale object pointed to by base shall be modified, or freed and a new locale object created.

Rationale: Austin Group Defect Report(s) applied: 227.
See http://austingroupbugs.net/view.php?id=227

Change Number: XSH/TC1/D5/0404 [283]

On Page: 1364 Line: 44746 Section: newlocale()

In the DESCRIPTION section, change from:

The results are undefined if the base argument is the special locale object LC_GLOBAL_LOCALE.

to:

The behavior is undefined if the base argument is the special locale object LC_GLOBAL_LOCALE, or is not a valid locale object handle and is not (locale_t)0.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0405 [295]

On Page: 1365 Line: 44768, 44787 Section: newlocale()

In the EXAMPLES section, change both occurrences of "NULL" to "(locale_t)0" (without the quotes).

Rationale: Austin Group Defect Report(s) applied: 295.
See http://austingroupbugs.net/view.php?id=295

Change Number: XSH/TC1/D5/0406 [227]

On Page: 1366 Line: 44793 Section: newlocale()

In the APPLICATION USAGE section, change from:

Handles for locale objects created by the newlocale() function should be released by a corresponding call to freelocale().

to:

Handles for locale objects returned by the newlocale() function should either be released by a corresponding call to freelocale(), or be used as a base locale to another newlocale() call.

Rationale: Austin Group Defect Report(s) applied: 227.
See http://austingroupbugs.net/view.php?id=227

Change Number: XSH/TC1/D5/0407 [68]

On Page: 1367 Line: 44840 Section: nextafter()

In the RETURN VALUE section, change from:

[MX]If x!=y and the correct function value is subnormal, zero, or underflows, a range error shall occur, and either the correct function value (if representable) or 0.0 shall be returned.[/MX]

to:

[MX]If x!=y and the correct function value is subnormal, zero, or underflows, a range error shall occur, and[/MX] [MXX]the correct function value (if representable) or[/MXX] [MX]0.0 shall be returned.[/MX]

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0408 [357]

On Page: 1368 Line: 44858 Section: nextafter()

In the APPLICATION USAGE section, add the following after L44858:

When <tgmath.h> is included, note that the return type of nextafter() depends on the generic typing deduced from both arguments, while the return type of nexttoward() depends only on the generic typing of the first argument.

Rationale: Austin Group Defect Report(s) applied: 357.
See http://austingroupbugs.net/view.php?id=357

Change Number: XSH/TC1/D5/0409 [403]

On Page: 1369 Line: 44912 Section: nftw()

In the DESCRIPTION section, change from:

FTW_F The object is a file.

to:

FTW_F The object is a non-directory file.

Rationale: Austin Group Defect Report(s) applied: 403.
See http://austingroupbugs.net/view.php?id=403

Change Number: XSH/TC1/D5/0410 [324]

On Page: 1370 Line: 44944 Section: nftw()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of path is not a directory.

to:

A component of path names an existing file that is neither a directory nor a symbolic link to a directory.

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0411 [403]

On Page: 1371 Line: 44972 Section: nftw()

In the EXAMPLES section, change from:

```
(tflag == FTW_D) ? "d" : (tflag == FTW_DNR) ? "dnr" :
(tflag == FTW_DP) ? "dp" : (tflag == FTW_F) ? "f" :
(tflag == FTW_NS) ? "ns" : (tflag == FTW_SL) ? "sl" :
(tflag == FTW_SLN) ? "sln" : "???",
```

to:

```
(tflag == FTW_D) ? "d" : (tflag == FTW_DNR) ? "dnr" :
```

```
(tflag == FTW_DP) ? "dp" : (tflag == FTW_F) ?
    (S_ISBLK(sb->st_mode) ? "f b" :
     S_ISCHR(sb->st_mode) ? "f c" :
     S_ISFIFO(sb->st_mode) ? "f p" :
     S_ISREG(sb->st_mode) ? "f r" :
     S_ISSOCK(sb->st_mode) ? "f s" : "f ?") :
(tflag == FTW_NS) ? "ns" : (tflag == FTW_SL) ? "sl" :
(tflag == FTW_SLN) ? "sln" : "?",
```

Rationale: Austin Group Defect Report(s) applied: 403.
See http://austingroupbugs.net/view.php?id=403

Change Number: XSH/TC1/D5/0412 [302]

On Page: 1375 Line: 45090 Section: nl_langinfo()

change from:

... the locale of the process ...

to:

... the current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0413 [75]

On Page: 1375 Line: 45100 Section: nl_langinfo()

In the DESCRIPTION section, delete:

Calls to setlocale() with a category corresponding to the category of item (see <langinfo.h>), or to the category LC_ALL, may overwrite the array pointed to by the return value. Calls to uselocale() which change the category corresponding to the category of item may overwrite the array pointed to by the return value.

Rationale: Austin Group Defect Report(s) applied: 75.
See http://austingroupbugs.net/view.php?id=75

Change Number: XSH/TC1/D5/0414 [283]

On Page: 1375 Line: 45104 Section: nl_langinfo()

In the DESCRIPTION section, add a new paragraph to the end of the section:

The behavior is undefined if the locale argument to nl_langinfo_l() is the special locale object LC_GLOBAL_LOCALE or is not a valid locale object handle.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0415 [75,402]

On Page: 1375 Line: 45109 Section: nl_langinfo()

In the RETURN VALUE section, change from:

This pointer may point to static data that may be overwritten on the next call to either function.

to:

The application shall not modify the string returned. The pointer returned by nl_langinfo() might be invalidated or the string content might be overwritten by a subsequent call to nl_langinfo() in any thread or to nl_langinfo_l() in the same thread or the initial thread, by subsequent calls to setlocale() with a category corresponding to the category of item (see <langinfo.h>) or the category LC_ALL, or by subsequent calls to uselocale() which change the category corresponding to the category of item. The pointer returned by nl_langinfo_l() might be invalidated or the string content might be overwritten by a subsequent call to nl_langinfo_l() in the same thread or to nl_langinfo() in any thread, or by subsequent calls to freelocale() or newlocale() which free or modify the locale object that was passed to nl_langinfo_l().

Rationale: Austin Group Defect Report(s) applied: 75,402.
See http://austingroupbugs.net/view.php?id=75
See http://austingroupbugs.net/view.php?id=402

Change Number: XSH/TC1/D5/0416 [283]

On Page: 1375 Line: 45111 Section: nl_langinfo()

In the ERRORS section, change from:

The nl_langinfo_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0417 [402]

On Page: 1376 Line: 45126 Section: nl_langinfo()

In the RATIONALE section, change from:

None.

to:

The possible interactions between internal data used by nl_langinfo() and nl_langinfo_l() are complicated by the fact that nl_langinfo_l()

must be thread-safe but nl_langinfo() need not be. The various implementation choices are:

1. nl_langinfo_l() and nl_langinfo() use separate buffers, or at least one of them does not use an internal string buffer. In this case there are no interactions.

2. nl_langinfo_l() and nl_langinfo() share an internal per-thread buffer. There can be interactions, but only in the same thread.

3. nl_langinfo_l() uses an internal per-thread buffer, and nl_langinfo() uses (in all threads) the same buffer that nl_langinfo_l() uses in the initial thread. There can be interactions, but only when nl_langinfo_l() is called in the initial thread.

Rationale: Austin Group Defect Report(s) applied: 402.
See http://austingroupbugs.net/view.php?id=402

Change Number: XSH/TC1/D5/0418 [292]

On Page: 1380 Line: 45209 Section: open()

In the DESCRIPTION section, change from:

O_DIRECTORY If path does not name a directory, fail and set errno to [ENOTDIR].

to:

O_DIRECTORY If path resolves to a non-directory file, fail and set errno to [ENOTDIR].

Rationale: Austin Group Defect Report(s) applied: 292.
See http://austingroupbugs.net/view.php?id=292
The standard is clear regarding symbolic links; however, there is an overlap between the [ENOENT] and [ENOTDIR] error conditions. The intention is that an [ENOENT] error should result when the pathname does not resolve.

Change Number: XSH/TC1/D5/0419 [141]

On Page: 1380 Line: 45239 Section: open()

In the DESCRIPTION section (for O_NONBLOCK), change from:

Otherwise, the behavior of O_NONBLOCK is unspecified.

to:

Otherwise, the O_NONBLOCK flag shall not cause an error, but it is unspecified whether the file status flags will include the O_NONBLOCK flag.

Rationale: Austin Group Defect Report(s) applied: 141.
See http://austingroupbugs.net/view.php?id=141

Change Number: XSH/TC1/D5/0420 [461]

On Page: 1382 Line: 45296 Section: open()

In the DESCRIPTION section, change from:

... the current working directory is used ...

to:

... the current working directory shall be used ...

Rationale: Austin Group Defect Report(s) applied: 461.
See http://austingroupbugs.net/view.php?id=461

Change Number: XSH/TC1/D5/0421 [390]

On Page: 1382 Line: 45301 Section: open()

In the RETURN VALUE section, change from:

If - is returned,

to:

If -1 is returned,

Rationale: Austin Group Defect Report(s) applied: 390.
See http://austingroupbugs.net/view.php?id=390

Change Number: XSH/TC1/D5/0422 [146]

On Page: 1382 Line: 45322 Section: open()

In the ERRORS section, change from:

[ENOENT] O_CREAT is not set and the named file does not exist; or O_CREAT is set and either the path prefix does not exist or the path argument points to an empty string.

to:

[ENOENT] O_CREAT is not set and a component of path does not name an existing file, or O_CREAT is set and a component of the path prefix of path does not name an existing file, or path points to an empty string.

[ENOENT] or [ENOTDIR]
O_CREAT is set, and the path argument contains at least one non-<slash> character and ends with one or more trailing <slash> characters. If path names an existing file, an [ENOENT] error shall not occur.

Rationale: Austin Group Defect Report(s) applied: 146.
See http://austingroupbugs.net/view.php?id=146

Change Number: XSH/TC1/D5/0423 [324]

On Page: 1382 Line: 45329 Section: open()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of the path prefix is not a directory; ...

to:

A component of the path prefix names an existing file that is neither a directory nor a symbolic link to a directory; ...

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0424 [292]

On Page: 1382 Line: 45333 Section: open()

In the ERRORS section for the [ENOTDIR] error, change from:

... or O_DIRECTORY was specified and the path argument does not name a directory.

to:

... or O_DIRECTORY was specified and the path argument resolves to a non-directory file.

Rationale: Austin Group Defect Report(s) applied: 292.
See http://austingroupbugs.net/view.php?id=292
The standard is clear regarding symbolic links; however, there is an overlap between the [ENOENT] and [ENOTDIR] error conditions. The intention is that an [ENOENT] error should result when the pathname does not resolve.

Change Number: XSH/TC1/D5/0425 [278]

On Page: 1383 Line: 45348 open()

In the ERRORS section, add (after the [EBADF] error):

[ENOTDIR] The path argument is not an absolute path and fd is a file descriptor associated with a non-directory file.

Rationale: Austin Group Defect Report(s) applied: 278.
See http://austingroupbugs.net/view.php?id=278

Change Number: XSH/TC1/D5/0426 [278]

On Page: 1383 Line: 45364 Section: open()

In the ERRORS section, delete:

The openat() function may fail if:

[ENOTDIR] The path argument is not an absolute path and fd is neither

AT_FDCWD nor a file descriptor associated with a directory.

Rationale: Austin Group Defect Report(s) applied: 278.
See http://austingroupbugs.net/view.php?id=278

Change Number: XSH/TC1/D5/0427 [291]

On Page: 1384 Line: 45377-45379,45409-45411 Section: open()

In the EXAMPLES section, change all instances of "filename" to "pathname".

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XSH/TC1/D5/0428 [307]

On Page: 1385 Line: 45427 Section: open()

In the RATIONALE section, add the following text to the end of the first paragraph (at the end of P1385 L45427):

Not all combinations of flags make sense. For example, using O_SEARCH | O_CREAT will successfully open a pre-existing directory for searching, but if there is no existing file by that name, then it is unspecified whether a regular file will be created. Likewise, if a non-directory file descriptor is successfully returned, it is unspecified whether that descriptor will have execute permissions as if by O_EXEC (note that it is unspecified whether O_EXEC and O_SEARCH have the same value).

And then move the entire paragraph from RATIONALE to APPLICATION USAGE after P1385 L45424.

Rationale: Austin Group Defect Report(s) applied: 307.
See http://austingroupbugs.net/view.php?id=307

Change Number: XSH/TC1/D5/0429 [389,401]

On Page: 1398 Line: 45791 Section: perror()

In the DESCRIPTION section, after L45791 add two new paragraphs, extending the CX shading to include them:

On error, perror() shall set the error indicator for the stream to which stderr points, and shall set errno to indicate the error.

Since no value is returned, an application wishing to check for error situations should call clearerr(stderr) before calling perror(), then if ferror(stderr) returns non-zero, the value of errno indicates which error occurred.

Rationale: Austin Group Defect Report(s) applied: 389,401.
See http://austingroupbugs.net/view.php?id=389
See http://austingroupbugs.net/view.php?id=401

Change Number: XSH/TC1/D5/0430 [389]

On Page: 1398 Line: 45795 Section: perror()

In the ERRORS section, change from:

No errors are defined.

to:

[CX]Refer to fputc().[/CX]

Rationale: Austin Group Defect Report(s) applied: 389.
See http://austingroupbugs.net/view.php?id=389

Change Number: XSH/TC1/D5/0431 [389,401]

On Page: 1398 Line: 45810 Section: perror()

In the APPLICATION USAGE section, change from:

None.

to:

Application writers may prefer to use alternative interfaces instead of perror(), such as strerror_r() in combination with fprintf().

Rationale: Austin Group Defect Report(s) applied: 389,401.
See http://austingroupbugs.net/view.php?id=389
See http://austingroupbugs.net/view.php?id=401

Change Number: XSH/TC1/D5/0432 [14]

On Page: 1409 Line: 46171 Section: popen()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0433 [291]

On Page: 1422 Line: 46551 Section: posix_spawn()

In the DESCRIPTION section, change "filename" to "filename string".

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XSH/TC1/D5/0434 [173]

On Page: 1423 Line: 46584 Section: posix_spawn()

In the DESCRIPTION section, after L46584 add a new paragraph:

10222
10223 If file descriptor 0, 1, or 2 would otherwise be closed in the new process
10224 image created by posix_spawn() or posix_spawnp(), implementations may
10225 open an unspecified file for the file descriptor in the new process
10226 image. If a standard utility or a conforming application is executed
10227 with file descriptor 0 not open for reading or with file descriptor
10228 1 or 2 not open for writing, the environment in which the utility or
10229 application is executed shall be deemed non-conforming, and consequently
10230 the utility or application might not behave as described in this standard.
10231
10232 *Rationale:* Austin Group Defect Report(s) applied: 173.
10233 See http://austingroupbugs.net/view.php?id=173
10234
10235
10236 **Change Number: XSH/TC1/D5/0435** [173]
10237
10238
10239 On Page: 1425 Line: 46681 Section: posix_spawn()
10240
10241 In the APPLICATION USAGE section, add at the end of the section:
10242
10243 See also the APPLICATION USAGE section for exec().
10244
10245 *Rationale:* Austin Group Defect Report(s) applied: 173.
10246 See http://austingroupbugs.net/view.php?id=173
10247
10248
10249 **Change Number: XSH/TC1/D5/0436** [418]
10250
10251
10252 On Page: 1430 Line: 46893-46895 Section: posix_spawn_file_actions_addclose()
10253
10254 In the ERRORS section, change from:
10255
10256 These functions shall fail if:
10257
10258 [EBADF] The value specified by fildes is negative or greater than or
10259 equal to {OPEN_MAX}.
10260
10261 to:
10262
10263 The posix_spawn_file_actions_addopen() function shall fail if:
10264
10265 [EBADF] The value specified by fildes is negative or greater than or
10266 equal to {OPEN_MAX}.
10267
10268 The posix_spawn_file_actions_addclose() function shall fail if:
10269
10270 [EBADF] The value specified by fildes is negative.
10271
10272 *Rationale:* Austin Group Defect Report(s) applied: 418.
10273 See http://austingroupbugs.net/view.php?id=418
10274
10275
10276 **Change Number: XSH/TC1/D5/0437** [149]
10277
10278
10279 On Page: 1431 Line: 46906 Section: posix_spawn_file_actions_addclose()
10280
10281 In the APPLICATION USAGE section, add a new paragraph at the end of
10282 the section:
10283
10284 Implementations may use file descriptors that must be inherited into

child processes for the child process to remain conforming, such as for message catalog or tracing purposes. Therefore, an application that calls posix_spawn_file_actions_addclose() with an arbitrary integer risks non-conforming behavior, and this function can only portably be used to close file descriptor values that the application has obtained through explicit actions, or for the three file descriptors corresponding to the standard file streams. In order to avoid a race condition of leaking an unintended file descriptor into a child process, an application should consider opening all file descriptors with the FD_CLOEXEC bit set unless the file descriptor is intended to be inherited across exec.

Rationale: Austin Group Defect Report(s) applied: 149.
See http://austingroupbugs.net/view.php?id=149

Change Number: XSH/TC1/D5/0438 [291]

On Page: 1431 Line: 46915 Section: posix_spawn_file_actions_addclose()

In the RATIONALE section, change "filenames" to "pathnames".

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XSH/TC1/D5/0439 [418]

On Page: 1432 Line: 46954 Section: posix_spawn_file_actions_addclose()

In the RATIONALE section, add a new paragraph at the end of the section:

The posix_spawn_file_actions_addclose() function is not required to check whether the file descriptor is less than {OPEN_MAX} because on some implementations {OPEN_MAX} reflects the RLIMIT_NOFILE soft limit and therefore calling setrlimit() to reduce this limit can result in an {OPEN_MAX} value less than or equal to an already open file descriptor. Applications need to be able to close such file descriptors on spawn. On implementations where {OPEN_MAX} does not change, it is recommended that posix_spawn_file_actions_addclose() should return EBADF if fildes is greater than or equal to {OPEN_MAX}.

Rationale: Austin Group Defect Report(s) applied: 418.
See http://austingroupbugs.net/view.php?id=418

Change Number: XSH/TC1/D5/0440 [149]

On Page: 1433 Line: 46997 Section: posix_spawn_file_actions_adddup2()

In the APPLICATION USAGE section, add a new paragraph at the end of the section:

Implementations may use file descriptors that must be inherited into child processes for the child process to remain conforming, such as for message catalog or tracing purposes. Therefore, an application that calls posix_spawn_file_actions_adddup2() with an arbitrary integer for newfildes risks non-conforming behavior, and this function can only portably be used to overwrite file descriptor values that the application has obtained through explicit actions, or for the three file descriptors corresponding to the standard file streams. In order to avoid a race

condition of leaking an unintended file descriptor into a child process, an application should consider opening all file descriptors with the FD_CLOEXEC bit set unless the file descriptor is intended to be inherited across exec.

Rationale: Austin Group Defect Report(s) applied: 149.
See http://austingroupbugs.net/view.php?id=149

Change Number: XSH/TC1/D5/0441 [358]

On Page: 1483 Line: 48027 Section: posix_trace_shutdown()

In the SYNOPSIS section, shade the declaration for posix_trace_shutdown() and mark with margin markers OB TRC.

Rationale: Austin Group Defect Report(s) applied: 358.
See http://austingroupbugs.net/view.php?id=358
Shading has stopped early in the synopsis. Shading is correct on P1507 (the pointer page).

Change Number: XSH/TC1/D5/0442 [119,428]

On Page: 1516 Line: 48952 Section: posix_typed_mem_open()

In the ERRORS section, move the [ENAMETOOLONG] error from L48952 to become a may fail error after inserting at the end of the section:

The posix_typed_mem_open() function may fail if:

Rationale: Austin Group Defect Report(s) applied: 119,428.
See http://austingroupbugs.net/view.php?id=119
See http://austingroupbugs.net/view.php?id=428

Change Number: XSH/TC1/D5/0443 [68]

On Page: 1518 Line: 49002 Section: pow()

In the RETURN VALUE section, change from:

If the correct value would cause underflow, and is not representable, a range error may occur, and [MX]either 0.0 (if supported), or[/MX] an implementation-defined value shall be returned.

to:

If the correct value would cause underflow, [MXX]and is not representable[/MXX], a range error may occur, and pow(), powf(), and powl() shall return [MXX]0.0, or[/MXX] (if IEC 60559 Floating-Point is not supported) an implementation-defined value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0444 [148]

On Page: 1518 Line: 49005-49008 Section: pow()

In the RETURN VALUE section, change from:

On systems that support the IEC 60559 Floating-Point option, a pole error shall occur ...

to:

On systems that support the IEC 60559 Floating-Point option, if x is +/-0, a pole error shall occur ...

Rationale: Austin Group Defect Report(s) applied: 148.
See http://austingroupbugs.net/view.php?id=148

Change Number: XSH/TC1/D5/0445 [68]

On Page: 1519 Line: 49025 Section: pow()

In the RETURN VALUE section, change the MX shading to MXX for:

If the correct value would cause underflow, and is representable, a range error may occur and the correct value shall be returned.

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0446 [372]

On Page: 1526 Line: 49241 Section: pselect()

In the ERRORS section, [EINTR] error, remove the XSI shading from:

If SA_RESTART has been set for the interrupting signal, it is implementation-defined whether the function restarts or returns with [EINTR].

Rationale: Austin Group Defect Report(s) applied: 372.
See http://austingroupbugs.net/view.php?id=372

Change Number: XSH/TC1/D5/0447 [399,428]

On Page: 1528 Line: 49331 Section: psiginfo()

In the DESCRIPTION section, add three new paragraphs to the end of the section:

The psiginfo() and psignal() functions shall not change the setting of errno if successful.

On error, the psiginfo() and psignal() functions shall set the error indicator for the stream to which stderr points, and shall set errno to indicate the error.

Since no value is returned, an application wishing to check for error situations should set errno to 0, then call psiginfo() or psignal(),

then check errno.

Rationale: Austin Group Defect Report(s) applied: 399,428.
See http://austingroupbugs.net/view.php?id=399
See http://austingroupbugs.net/view.php?id=428

Change Number: XSH/TC1/D5/0448 [399]

On Page: 1528 Line: 49335 Section: psiginfo()

In the ERRORS section, change from:

No errors are defined.

to:

Refer to fputc().

Rationale: Austin Group Defect Report(s) applied: 399.
See http://austingroupbugs.net/view.php?id=399

Change Number: XSH/TC1/D5/0449 [399,401]

On Page: 1528 Line: 49339 Section: psiginfo()

In the APPLICATION USAGE section, change from:

None.

to:

As an alternative to setting errno to zero before the call and checking if it is non-zero afterwards, applications can use ferror() to detect whether psiginfo() or psignal() encountered an error.

An application wishing to use this method to check for error situations should call clearerr(stderr) before calling psiginfo() or psignal(), then if ferror(stderr) returns non-zero, the value of errno indicates which error occurred.

Rationale: Austin Group Defect Report(s) applied: 399,401.
See http://austingroupbugs.net/view.php?id=399
See http://austingroupbugs.net/view.php?id=401

Change Number: XSH/TC1/D5/0450 [314]

On Page: 1540 Line: 49793 Section: pthread_attr_getinheritsched()

In the ERRORS section, insert before L49793:

The pthread_attr_setinheritsched() function shall fail if:

Move the [ENOTSUP] error from L49795 to after this new line.

Rationale: Austin Group Defect Report(s) applied: 314.
See http://austingroupbugs.net/view.php?id=314

Change Number: XSH/TC1/D5/0451 [314]

On Page: 1542 Line: 49864 Section: pthread_attr_getschedparam()

In the ERRORS section, insert before L49864:

The pthread_attr_setschedparam() function shall fail if:

Move the [ENOTSUP] error from L49866 to after this new line.

Rationale: Austin Group Defect Report(s) applied: 314.
See http://austingroupbugs.net/view.php?id=314

Change Number: XSH/TC1/D5/0452 [314]

On Page: 1544 Line: 49923 Section: pthread_attr_getschedpolicy()

In the ERRORS section, insert before L49923:

The pthread_attr_setschedpolicy() function shall fail if:

Move the [ENOTSUP] error from L49925 to after this new line.

Rationale: Austin Group Defect Report(s) applied: 314.
See http://austingroupbugs.net/view.php?id=314

Change Number: XSH/TC1/D5/0453 [314]

On Page: 1546 Line: 49986 Section: pthread_attr_getscope()

In the ERRORS section, insert before L49986:

The pthread_attr_setscope() function shall fail if:

Move the [ENOTSUP] error from L49988 to after this new line.

Rationale: Austin Group Defect Report(s) applied: 314.
See http://austingroupbugs.net/view.php?id=314

Change Number: XSH/TC1/D5/0454 [229]

On Page: 1575-1576 Line: 50593 Section: pthread_cleanup_pop()

On L50593, in the EXAMPLES section, change from:

```
while ((l->lock_count < 0) && (l->waiting_writers != 0))
```

to:

```
while ((l->lock_count < 0) || (l->waiting_writers != 0))
```

On L50608 change:

```
pthread_mutex_unlock(l);
```

to:

```
10600
10601   pthread_mutex_unlock(&l->lock);
10602
10603   On L50617 change from:
10604
10605   * This only happens if we have been canceled.
10606
10607   to:
10608
10609   * This only happens if we have been canceled. If the
10610   * lock is not held by a writer, there may be readers who
10611   * were blocked because waiting_writers was positive; they
10612   * can now be unblocked.
10613
10614   On L50619 change from:
10615
10616   pthread_cond_broadcast(&l->wcond);
10617
10618   to:
10619
10620   pthread_cond_broadcast(&l->rcond);
10621
10622   On L50620 increase the indentation of the closing brace from zero <space>
10623   characters to four <space> characters.
10624
10625   On L50644 change from:
10626
10627   pthread_cond_broadcast(&l->rcond)
10628
10629   to:
10630
10631   pthread_cond_broadcast(&l->rcond);
10632
10633   Rationale: Austin Group Defect Report(s) applied: 229.
10634   See http://austingroupbugs.net/view.php?id=229
10635
10636
10637   Change Number: XSH/TC1/D5/0455 [70]
10638
10639
10640   On Page: 1582 Line: 50877 Section: pthread_cond_destroy()
10641
10642   In the DESCRIPTION section, change from:
10643
10644   In cases where default condition variable attributes are appropriate,
10645   the macro PTHREAD_COND_INITIALIZER can be used to initialize condition
10646   variables that are statically allocated.
10647
10648   to:
10649
10650   In cases where default condition variable attributes are appropriate,
10651   the macro PTHREAD_COND_INITIALIZER can be used to initialize condition
10652   variables.
10653
10654   Rationale: Austin Group Defect Report(s) applied: 70.
10655   See http://austingroupbugs.net/view.php?id=70
10656
10657
10658   Change Number: XSH/TC1/D5/0456 [91,286,437]
10659
10660
10661   On Page: 1587 Line: 51027-51034 Section: pthread_cond_timedwait()
10662
```

In the DESCRIPTION section, change from:

The pthread_cond_timedwait() function shall be equivalent to pthread_cond_wait(), except that an error is returned if the absolute time specified by abstime passes (that is, system time equals or exceeds abstime) before the condition cond is signaled or broadcasted, or if the absolute time specified by abstime has already been passed at the time of the call.

The condition variable shall have a clock attribute which specifies the clock that shall be used to measure the time specified by the abstime argument. When such timeouts occur, pthread_cond_timedwait() shall nonetheless release and re-acquire the mutex referenced by mutex. The pthread_cond_timedwait() function is also a cancellation point.

to:

The pthread_cond_timedwait() function shall be equivalent to pthread_cond_wait(), except that an error is returned if the absolute time specified by abstime passes (that is, system time equals or exceeds abstime) before the condition cond is signaled or broadcasted, or if the absolute time specified by abstime has already been passed at the time of the call. When such timeouts occur, pthread_cond_timedwait() shall nonetheless release and re-acquire the mutex referenced by mutex, and may consume a condition signal directed concurrently at the condition variable.

The condition variable shall have a clock attribute which specifies the clock that shall be used to measure the time specified by the abstime argument. The pthread_cond_timedwait() function is also a cancellation point.

Rationale: Austin Group Defect Report(s) applied: 91,286,437.
See http://austingroupbugs.net/view.php?id=91
See http://austingroupbugs.net/view.php?id=286
See http://austingroupbugs.net/view.php?id=437

Change Number: XSH/TC1/D5/0457 [239]

On Page: 1590 Line: 51186 Section: pthread_cond_timedwait()

In the RATIONALE section, change from:

```
if (rc == 0) setmystate(&t);
```

to:

```
if (rc == 0 || mypredicate(&t))
    setmystate(&t);
```

Rationale: Austin Group Defect Report(s) applied: 239.
See http://austingroupbugs.net/view.php?id=239

Change Number: XSH/TC1/D5/0458 [302]

On Page: 1601 Line: 51456 Section: pthread_create()

In the DESCRIPTION section, change from:

[XSI]The alternate stack shall not be inherited.[/XSI]

to:

The thread-local current locale [XSI]and the alternate stack[/XSI]
shall not be inherited.

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0459 [314]

On Page: 1613 Line: 51845 Section: pthread_getschedparam()

In the ERRORS section, insert before L51845:

The pthread_setschedparam() function shall fail if:

Move the two [ENOTSUP] errors from L51848-51851 to after this new line.

Rationale: Austin Group Defect Report(s) applied: 314.
See http://austingroupbugs.net/view.php?id=314

Change Number: XSH/TC1/D5/0460 [70,428]

On Page: 1628 Line: 52373 Section: pthread_mutex_init()

In the DESCRIPTION section, change from:

In cases where default mutex attributes are appropriate, the macro
PTHREAD_MUTEX_INITIALIZER can be used to initialize mutexes that are
statically allocated.

to:

In cases where default mutex attributes are appropriate, the macro
PTHREAD_MUTEX_INITIALIZER can be used to initialize mutexes.

Rationale: Austin Group Defect Report(s) applied: 70,428.
See http://austingroupbugs.net/view.php?id=70
See http://austingroupbugs.net/view.php?id=428

Change Number: XSH/TC1/D5/0461 [121]

On Page: 1638 Line: 52690 Section: pthread_mutex_lock()

In the DESCRIPTION section, change from:

The mutex object referenced by mutex shall be locked by calling
pthread_mutex_lock(). If the mutex is already locked, the calling
thread shall block until the mutex becomes available. This operation
shall return with the mutex object referenced by mutex in the locked
state with the calling thread as its owner.

If the mutex type is PTHREAD_MUTEX_NORMAL, deadlock detection shall

not be provided. Attempting to relock the mutex causes deadlock. If a thread attempts to unlock a mutex that it has not locked or a mutex which is unlocked, undefined behavior results.

If the mutex type is PTHREAD_MUTEX_ERRORCHECK, then error checking shall be provided. If a thread attempts to relock a mutex that it has already locked, an error shall be returned. If a thread attempts to unlock a mutex that it has not locked or a mutex which is unlocked, an error shall be returned.

If the mutex type is PTHREAD_MUTEX_RECURSIVE, then the mutex shall maintain the concept of a lock count. When a thread successfully acquires a mutex for the first time, the lock count shall be set to one. Every time a thread relocks this mutex, the lock count shall be incremented by one. Each time the thread unlocks the mutex, the lock count shall be decremented by one. When the lock count reaches zero, the mutex shall become available for other threads to acquire. If a thread attempts to unlock a mutex that it has not locked or a mutex which is unlocked, an error shall be returned.

If the mutex type is PTHREAD_MUTEX_DEFAULT, attempting to recursively lock the mutex results in undefined behavior. Attempting to unlock the mutex if it was not locked by the calling thread results in undefined behavior. Attempting to unlock the mutex if it is not locked results in undefined behavior.

to:

The mutex object referenced by mutex shall be locked by a call to pthread_mutex_lock() that returns zero or EOWNERDEAD. If the mutex is already locked by another thread, the calling thread shall block until the mutex becomes available. This operation shall return with the mutex object referenced by mutex in the locked state with the calling thread as its owner. If a thread attempts to relock a mutex that it has already locked, pthread_mutex_lock() shall behave as described in the Relock column of the following table. If a thread attempts to unlock a mutex that it has not locked or a mutex which is unlocked, pthread_mutex_unlock() shall behave as described in the Unlock When Not Owner column of the following table.

Mutex Type	Robustness	Relock	Unlock When Not Owner
normal	non-robust	deadlock	undefined behavior
normal	robust	deadlock	error returned
errorcheck	either	error returned	error returned
recursive	either	recursive (see below)	error returned
default	non-robust	undefined behavior[1]	undefined behavior[1]
default	robust	undefined behavior[1]	error returned

1. If the mutex type is PTHREAD_MUTEX_DEFAULT, the behavior of pthread_mutex_lock() may correspond to one of the three other standard mutex types as described in the table above. If it does not correspond to one of those three, the behavior is undefined for the cases marked [1].

Where the table indicates recursive behavior, the mutex shall maintain the concept of a lock count. When a thread successfully acquires a mutex for the first time, the lock count shall be set to one. Every time a thread relocks this mutex, the lock count shall be incremented by one. Each time the thread unlocks the mutex, the lock count shall be decremented by one. When the lock count reaches zero, the mutex shall become available for other threads to acquire.

Rationale: Austin Group Defect Report(s) applied: 121.
See http://austingroupbugs.net/view.php?id=121

Change Number: XSH/TC1/D5/0462 [92,428]

On Page: 1639 Line: 52739 Section: pthread_mutex_lock()

In the RETURN VALUE section, change from:

pthread_mutex_lock() and pthread_mutex_unlock()

to

pthread_mutex_lock(), pthread_mutex_trylock(), and pthread_mutex_unlock()

Delete L52741-52742 (2nd para of RETURN VALUE).

Rationale: Austin Group Defect Report(s) applied: 92,428.
See http://austingroupbugs.net/view.php?id=92
See http://austingroupbugs.net/view.php?id=428

Change Number: XSH/TC1/D5/0463 [121]

On Page: 1639 Line: 52763 Section: pthread_mutex_lock()

In the ERRORS section, change the [EPERM] error from:

The mutex type is PTHREAD_MUTEX_ERRORCHECK or the mutex is a robust mutex, and the current thread does not own the mutex.

to:

The mutex type is PTHREAD_MUTEX_ERRORCHECK or PTHREAD_MUTEX_RECURSIVE, or the mutex is a robust mutex, and the current thread does not own the mutex.

Rationale: Austin Group Defect Report(s) applied: 121.
See http://austingroupbugs.net/view.php?id=121

Change Number: XSH/TC1/D5/0464 [121]

On Page: 1661 Line: 53481 Section: pthread_mutexattr_gettype()

In the DESCRIPTION section, change from:

The type of mutex is contained in the type attribute of the mutex attributes. Valid mutex types include:

PTHREAD_MUTEX_NORMAL
This type of mutex does not detect deadlock. A thread attempting to relock this mutex without first unlocking it shall deadlock. Attempting to unlock a mutex locked by a different thread results in undefined behavior. Attempting to unlock an unlocked mutex results in undefined behavior.

PTHREAD_MUTEX_ERRORCHECK
This type of mutex provides error checking. A thread attempting to relock this mutex without first unlocking it shall return with an error. A thread attempting to unlock a mutex which another thread has locked shall return with an error. A thread attempting to unlock an unlocked mutex shall return with an error.

PTHREAD_MUTEX_RECURSIVE
A thread attempting to relock this mutex without first unlocking it shall succeed in locking the mutex. The relocking deadlock which can occur with mutexes of type PTHREAD_MUTEX_NORMAL cannot occur with this type of mutex. Multiple locks of this mutex shall require the same number of unlocks to release the mutex before another thread can acquire the mutex. A thread attempting to unlock a mutex which another thread has locked shall return with an error. A thread attempting to unlock an unlocked mutex shall return with an error.

PTHREAD_MUTEX_DEFAULT
Attempting to recursively lock a mutex of this type results in undefined behavior. Attempting to unlock a mutex of this type which was not locked by the calling thread results in undefined behavior. Attempting to unlock a mutex of this type which is not locked results in undefined behavior. An implementation may map this mutex to one of the other mutex types.

to:

The type of mutex is contained in the type attribute of the mutex attributes. Valid mutex types include:

PTHREAD_MUTEX_NORMAL
PTHREAD_MUTEX_ERRORCHECK
PTHREAD_MUTEX_RECURSIVE
PTHREAD_MUTEX_DEFAULT

The mutex type affects the behavior of calls which lock and unlock the mutex. See pthread_mutex_lock() for details. An implementation may map PTHREAD_MUTEX_DEFAULT to one of the other mutex types.

Rationale: Austin Group Defect Report(s) applied: 121.
See http://austingroupbugs.net/view.php?id=121

Change Number: XSH/TC1/D5/0465 [70]

On Page: 1671 Line: 53703 Section: pthread_rwlock_destroy()

In the DESCRIPTION section, change from:

In cases where default read-write lock attributes are appropriate, the macro PTHREAD_RWLOCK_INITIALIZER can be used to initialize read-write locks that are statically allocated.

to:

In cases where default read-write lock attributes are appropriate, the

macro PTHREAD_RWLOCK_INITIALIZER can be used to initialize read-write locks.

Rationale: Austin Group Defect Report(s) applied: 70.
See http://austingroupbugs.net/view.php?id=70

Change Number: XSH/TC1/D5/0466 [314]

On Page: 1698 Line: 54396 Section: pthread_setschedprio()

In the ERRORS section, delete the [ENOTSUP] error.

Rationale: Austin Group Defect Report(s) applied: 314.
See http://austingroupbugs.net/view.php?id=314

Change Number: XSH/TC1/D5/0467 [319]

On Page: 1701 Line: 54469 Section: pthread_sigmask()

Change:

... unless the signal was generated by the kill() function, the sigqueue() function, or the raise() function.

to:

... unless the signal was generated by the action of another process, or by one of the functions kill(), pthread_kill(), raise(), or sigqueue().

Rationale: Austin Group Defect Report(s) applied: 319.
See http://austingroupbugs.net/view.php?id=319

Change Number: XSH/TC1/D5/0468 [75]

On Page: 1712 Line: 54752 Section: ptsname()

In the RETURN VALUE section, add a new paragraph at the end of the section:

The application shall not modify the string returned. The returned pointer might be invalidated or the string content might be overwritten by a subsequent call to ptsname().

On L54758, in the APPLICATION USAGE section, change from:

The value returned may point to a static data area that is overwritten by each call to ptsname().

to:

None.

Rationale: Austin Group Defect Report(s) applied: 75.
See http://austingroupbugs.net/view.php?id=75

Change Number: XSH/TC1/D5/0469 [96]

On Page: 1712 Line: 54760 Section: ptsname()

In the RATIONALE section, change from:

None.

to:

See RATIONALE for posix_openpt().

On L54764, in the SEE ALSO section, add posix_openpt().

Rationale: Austin Group Defect Report(s) applied: 96.
See http://austingroupbugs.net/view.php?id=96
Add reference to posix_openpt() to the RATIONALE and SEE
ALSO sections.

Change Number: XSH/TC1/D5/0470 [14]

On Page: 1713 Line: 54800 Section: putc()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0471 [14]

On Page: 1715 Line: 54834 Section: putchar()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0472 [167]

On Page: 1717 Line: 54880 Section: putenv()

In the APPLICATION USAGE section, change from:

The setenv() function is preferred over this function.

to:

The setenv() function is preferred over this function. One reason is
that putenv() is optional and therefore less portable. Another is that
using putenv() can slow down environment searches, as explained in the
RATIONALE for getenv().

Rationale: Austin Group Defect Report(s) applied: 167.
See http://austingroupbugs.net/view.php?id=167

Change Number: XSH/TC1/D5/0473 [167]

On Page: 1717 Line: 54882 Section: putenv()

In the RATIONALE section, change from:

The standard developers noted that putenv() is the only function available to add to the environment without permitting memory leaks.

to:

Refer to the RATIONALE section in setenv().

Rationale: Austin Group Defect Report(s) applied: 167.
See http://austingroupbugs.net/view.php?id=167

Change Number: XSH/TC1/D5/0474 [273,438]

On Page: 1717 Line: 54855 Section: putenv()

In the DESCRIPTION section, remove the last sentence of the section (L54855-54856):

The space used by string is no longer used once a new string which defines name is passed to putenv().

Rationale: Austin Group Defect Report(s) applied: 273,438.
See http://austingroupbugs.net/view.php?id=273
See http://austingroupbugs.net/view.php?id=438

Change Number: XSH/TC1/D5/0475 [273]

On Page: 1717 Line: 54879 Section: putenv()

In the APPLICATION USAGE section, add after L54879:

Although the space used by string is no longer used once a new string which defines name is passed to putenv(), if any thread in the application has used getenv() to retrieve a pointer to this variable, it should not be freed by calling free(). If the changed environment variable is one known by the system (such as the locale environment variables) the application should never free the buffer used by earlier calls to putenv() for the same variable.

Rationale: Austin Group Defect Report(s) applied: 273.
See http://austingroupbugs.net/view.php?id=273

Change Number: XSH/TC1/D5/0476 [174,412]

On Page: 1723 Line: 55077 Section: puts()

In the APPLICATION USAGE section, add to the end of the section:

This volume of POSIX.1-2008 requires that successful completion simply

return a non-negative integer. There are at least three known different implementation conventions for this requirement:

* Return a constant value.
* Return the last character written.
* Return the number of bytes written. Note that this implementation convention cannot be adhered to for strings longer than {INT_MAX} bytes as the value would not be representable in the return type of the function. For backwards compatibility, implementations can return the number of bytes for strings of up to INT_MAX bytes, and return INT_MAX for all longer strings.

Rationale: Austin Group Defect Report(s) applied: 174,412.
See http://austingroupbugs.net/view.php?id=174
See http://austingroupbugs.net/view.php?id=412

Change Number: XSH/TC1/D5/0477 [14]

On Page: 1724 Line: 55083 Section: puts()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0478 [14]

On Page: 1726 Line: 55126 Section: putwc()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0479 [14]

On Page: 1727 Line: 55157 Section: putwchar()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0480 [218]

On Page: 1739 Line: 55487 Section: read()

In the ERRORS section, change from:

[EAGAIN] The O_NONBLOCK flag is set for the file descriptor and the thread would be delayed.

11230 to:
11231
11232 [EAGAIN] The file is neither a pipe, nor a FIFO, nor a socket, the
11233 O_NONBLOCK flag is set for the file descriptor, and the thread would be
11234 delayed in the read operation.
11235
11236 *Rationale:* Austin Group Defect Report(s) applied: 218.
11237 See http://austingroupbugs.net/view.php?id=218
11238
11239
11240 **Change Number: XSH/TC1/D5/0481** [79]
11241
11242
11243 On Page: 1739 Line: 55497 Section: read()
11244
11245 In the ERRORS section [EIO], change from:
11246
11247 ... the process is ignoring or blocking the SIGTTIN signal, or the
11248 process group is orphaned.
11249
11250 to:
11251
11252 ... and either the calling thread is blocking SIGTTIN or the process is
11253 ignoring SIGTTIN or the process group of the process is orphaned.
11254
11255 *Rationale:* Austin Group Defect Report(s) applied: 79.
11256 See http://austingroupbugs.net/view.php?id=79
11257
11258
11259 **Change Number: XSH/TC1/D5/0482** [218]
11260
11261
11262 On Page: 1739 Line: 55506 Section: read()
11263
11264 In the ERRORS section, before L55506 insert:
11265
11266 The pread() function shall fail if:
11267
11268 [EINVAL] The file is a regular file or block special file, and the offset
11269 argument is negative. The file pointer shall remain unchanged.
11270
11271 [ESPIPE] The file is a pipe, FIFO, or socket.
11272
11273 *Rationale:* Austin Group Defect Report(s) applied: 218.
11274 See http://austingroupbugs.net/view.php?id=218
11275
11276
11277 **Change Number: XSH/TC1/D5/0483** [218]
11278
11279
11280 On Page: 1739 Line: 55507 Section: read()
11281
11282 In the ERRORS section, change from:
11283
11284 [EAGAIN] or [EWOULDBLOCK]
11285 The file descriptor is for a socket, is marked O_NONBLOCK, and no data
11286 is waiting to be received.
11287
11288 to:
11289
11290 [EAGAIN] The file is a pipe or FIFO, the O_NONBLOCK flag is set for the
11291 file descriptor, and the thread would be delayed in the read operation.
11292

[EAGAIN] or [EWOULDBLOCK]
The file is a socket, the O_NONBLOCK flag is set for the file descriptor, and the thread would be delayed in the read operation.

Rationale: Austin Group Defect Report(s) applied: 218.
See http://austingroupbugs.net/view.php?id=218

Change Number: XSH/TC1/D5/0484 [218]

On Page: 1740 Line: 55520 Section: read()

In the ERRORS section, delete L55520-55525 (the pread() "shall fail" errors).

Rationale: Austin Group Defect Report(s) applied: 218.
See http://austingroupbugs.net/view.php?id=218

Change Number: XSH/TC1/D5/0485 [218,428]

On Page: 1742 Line: 55612 Section: read()

In the RATIONALE section, add a new paragraph to the end of the section:

This volume of POSIX.1-2008 intentionally does not specify any pread() errors related to pipes, FIFOs, and sockets other than [ESPIPE].

Rationale: Austin Group Defect Report(s) applied: 218,428.
See http://austingroupbugs.net/view.php?id=218
See http://austingroupbugs.net/view.php?id=428

Change Number: XSH/TC1/D5/0486 [75]

On Page: 1744 Line: 55685 Section: readdir()

In the DESCRIPTION section, change from:

The pointer returned by readdir() points to data which may be overwritten by another call to readdir() on the same directory stream. This data is not overwritten by another call to readdir() on a different directory stream.

to:

The application shall not modify the structure to which the return value of readdir() points, nor any storage areas pointed to by pointers within the structure. The returned pointer, and pointers within the structure, might be invalidated or the structure or the storage areas might be overwritten by a subsequent call to readdir() on the same directory stream. They shall not be affected by a call to readdir() on a different directory stream.

Rationale: Austin Group Defect Report(s) applied: 75.
See http://austingroupbugs.net/view.php?id=75

Change Number: XSH/TC1/D5/0487 [120]

On Page: 1749 Line: 55841 Section: readlink()

In the NAME section, change from:

readlink, readlinkat - read the contents of a symbolic link relative to a directory file descriptor

to:

readlink, readlinkat - read the contents of a symbolic link

Rationale: Austin Group Defect Report(s) applied: 120.
See http://austingroupbugs.net/view.php?id=120
A correction is made to the NAME section.

Change Number: XSH/TC1/D5/0488 [461]

On Page: 1749 Line: 55862 Section: readlink()

In the DESCRIPTION section, change from:

... the current working directory is used ...

to:

... the current working directory shall be used ...

Rationale: Austin Group Defect Report(s) applied: 461.
See http://austingroupbugs.net/view.php?id=461

Change Number: XSH/TC1/D5/0489 [143]

On Page: 1749 Line: 55865 Section: readlink()

In the RETURN VALUE section, change from:

Upon successful completion, readlink() shall return the count of bytes placed in the buffer. Otherwise, it shall return a value of -1, leave the buffer unchanged, and set errno to indicate the error.

Upon successful completion, the readlinkat() function shall return 0. Otherwise, it shall return -1 and set errno to indicate the error.

to:

Upon successful completion, these functions shall return the count of bytes placed in the buffer. Otherwise, these functions shall return a value of -1, leave the buffer unchanged, and set errno to indicate the error.

Rationale: Austin Group Defect Report(s) applied: 143.
See http://austingroupbugs.net/view.php?id=143

Change Number: XSH/TC1/D5/0490 [324]

On Page: 1749 Line: 55880 Section: readlink()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of the path prefix is not a directory, ...

to:

A component of the path prefix names an existing file that is neither a directory nor a symbolic link to a directory, ...

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0491 [278]

On Page: 1750 Line: 55888 readlink()

In the ERRORS section, add (after the [EBADF] error):

[ENOTDIR] The path argument is not an absolute path and fd is a file descriptor associated with a non-directory file.

Rationale: Austin Group Defect Report(s) applied: 278.
See http://austingroupbugs.net/view.php?id=278

Change Number: XSH/TC1/D5/0492 [278]

On Page: 1750 Line: 55896 Section: readlink()

In the ERRORS section, delete:

The readlinkat() function may fail if:

[ENOTDIR] The path argument is not an absolute path and fd is neither AT_FDCWD nor a file descriptor associated with a directory.

Rationale: Austin Group Defect Report(s) applied: 278.
See http://austingroupbugs.net/view.php?id=278

Change Number: XSH/TC1/D5/0493 [455]

On Page: 1750 Line: 55904 Section: readlink()

In the EXAMPLES section, change from:

```
ssizet_t len;
```

to:

```
ssize_t len;
```

Rationale: Austin Group Defect Report(s) applied: 455.
See http://austingroupbugs.net/view.php?id=455

Change Number: XSH/TC1/D5/0494 [151,231]

On Page: 1750 Line: 55912 Section: readlink()

In the RATIONALE section, delete the first sentence:

Since POSIX.1-2008 does not require any association of file times with symbolic links, there is no requirement that file times be updated by readlink().

Rationale: Austin Group Defect Report(s) applied: 151,231.
See http://austingroupbugs.net/view.php?id=151
See http://austingroupbugs.net/view.php?id=231

Change Number: XSH/TC1/D5/0495 [400]

On Page: 1754 Line: 56030 Section: realloc()

In the DESCRIPTION section, change from:

The realloc() function shall change the size of the memory object pointed to by ptr to the size specified by size. The contents of the object shall remain unchanged up to the lesser of the new and old sizes. If the new size of the memory object would require movement of the object, the space for the previous instantiation of the object is freed. If the new size is larger, the contents of the newly allocated portion of the object are unspecified. If size is 0 and ptr is not a null pointer, the object pointed to is freed.

to:

The realloc() function shall deallocate the old object pointed to by ptr and return a pointer to a new object that has the size specified by size. The contents of the new object shall be the same as that of the old object prior to deallocation, up to the lesser of the new and old sizes. Any bytes in the new object beyond the size of the old object have indeterminate values. If the size of the space requested is zero, the behavior shall be implementation-defined: either a null pointer is returned, or the behavior shall be as if the size were some nonzero value, except that the returned pointer shall not be used to access an object.

Rationale: Austin Group Defect Report(s) applied: 400.
See http://austingroupbugs.net/view.php?id=400

Change Number: XSH/TC1/D5/0496 [400]

On Page: 1754 Line: 56046 Section: realloc()

In the RETURN VALUE section, change from:

Upon successful completion with a size not equal to 0, realloc() shall return a pointer to the (possibly moved) allocated space. If size is 0, either a null pointer or a unique pointer that can be successfully passed to free() shall be returned. If there is not enough available memory, realloc() shall return a null pointer [CX]and set errno to [ENOMEM].[/CX]

to:

Upon successful completion, realloc() shall return a pointer to the (possibly moved) allocated space. If size is 0, either:

* A null pointer shall be returned [CX]and errno set to an implementation defined value[/CX].
* A unique pointer that can be successfully passed to free() shall be returned, and the memory object pointed to by ptr shall be freed. The application shall ensure that the pointer is not used to access an object.

If there is not enough available memory, realloc() shall return a null pointer [CX]and set errno to [ENOMEM][/CX]. If realloc() returns a null pointer [CX]and errno has been set to a ENOMEM[/CX], the memory referenced by ptr shall not be changed.

Rationale: Austin Group Defect Report(s) applied: 400.
See http://austingroupbugs.net/view.php?id=400

Change Number: XSH/TC1/D5/0497 [400]

On Page: 1754 Line: 56056 Section: realloc()

In the APPLICATION USAGE section, change from:

None.

to

The description of realloc() has been modified from previous versions of this standard to align with C99. Previous versions explicitly permitted a call to realloc(p, 0) to free the space pointed to by p and return NULL. While this behavior could be interpreted as permitted by this version of the standard, the C language committee have indicated that this interpretation is incorrect. Applications should assume that if realloc returns a null pointer, the space pointed to be p has not been freed. Since this could lead to double-frees, implementations should also set errno if a null pointer actually indicates a failure, and applications should only free the space if errno was changed.

Rationale: Austin Group Defect Report(s) applied: 400.
See http://austingroupbugs.net/view.php?id=400

Change Number: XSH/TC1/D5/0498 [400]

On Page: 1754 Line: 56060 Section: realloc()

In the FUTURE DIRECTIONS section, change from:

None.

to:

This standard defers to the C standard. While that standard currently has language that might permit realloc(p, 0), where p is not a null pointer, to free p while still returning a null pointer, the committee responsible for that standard is considering clarifying the language to explicitly prohibit that alternative.

Rationale: Austin Group Defect Report(s) applied: 400.
See http://austingroupbugs.net/view.php?id=400

Change Number: XSH/TC1/D5/0499 [353]

On Page: 1756 Line: 56099 Section: realpath()

In the ERRORS section, [EACCES] error, change from:

Read or search permission was denied for a component of file_name.

to:

Search permission was denied for a component of the path prefix of file_name.

Rationale: Austin Group Defect Report(s) applied: 353.
See http://austingroupbugs.net/view.php?id=353

Change Number: XSH/TC1/D5/0500 [324]

On Page: 1756 Line: 56108 Section: realpath()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of the path prefix is not a directory, ...

to:

A component of the path prefix names an existing file that is neither a directory nor a symbolic link to a directory, ...

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0501 [353]

On Page: 1756 Line: 56112 Section: realpath()

After L56112 add a new "may fail" error:

[EACCES] The file_name argument does not begin with a <slash> and none of the symbolic links (if any) processed during pathname resolution of file_name had contents that began with a <slash>, and either search permission was denied for the current directory or read or search permission was denied for a directory above the current directory in the file hierarchy.

Rationale: Austin Group Defect Report(s) applied: 353.
See http://austingroupbugs.net/view.php?id=353

Change Number: XSH/TC1/D5/0502 [462]

On Page: 1760 Line: 56238 Section: recv()

11671 In the APPLICATION USAGE section, change from:
11672
11673 The recv() function is equivalent to recvfrom() with a zero address_len
11674 argument, and to read() if no flags are used.
11675
11676 to:
11677
11678 The recv() function is equivalent to recvfrom() with null pointer address
11679 and address_len arguments, and to read() if the socket argument refers
11680 to a socket and the flags argument is 0.
11681
11682 *Rationale:* Austin Group Defect Report(s) applied: 462.
11683 See http://austingroupbugs.net/view.php?id=462
11684
11685
11686 **Change Number: XSH/TC1/D5/0503** [464]
11687
11688
11689 On Page: 1761 Line: 56282 Section: recvfrom()
11690
11691 In the DESCRIPTION section, change from:
11692
11693 address_len Specifies the length of the sockaddr structure pointed to
11694 by the address argument.
11695
11696 to:
11697
11698 address_len Either a null pointer, if address is a null pointer, or a
11699 pointer to a socklen_t object which on input specifies the length of
11700 the supplied sockaddr structure, and on output specifies the length of
11701 the stored address.
11702
11703 *Rationale:* Austin Group Defect Report(s) applied: 464.
11704 See http://austingroupbugs.net/view.php?id=464
11705
11706
11707 **Change Number: XSH/TC1/D5/0504** [464]
11708
11709
11710 On Page: 1764 Line: 56385 Section: recvmsg()
11711
11712 In the DESCRIPTION section, change from:
11713
11714 In the msghdr structure, the msg_name and msg_namelen members specify
11715 the source address if the socket is unconnected. If the socket is
11716 connected, the msg_name and msg_namelen members shall be ignored.
11717 The msg_name member may be a null pointer if no names are desired or
11718 required.
11719
11720 to:
11721
11722 In the msghdr structure, the msg_name member may be a null pointer if the
11723 source address is not required. Otherwise, if the socket is unconnected,
11724 the msg_name member points to a sockaddr structure in which the source
11725 address is to be stored, and the msg_namelen member on input specifies
11726 the length of the supplied sockaddr structure and on output specifies
11727 the length of the stored address. If the actual length of the address is
11728 greater than the length of the supplied sockaddr structure, the stored
11729 address shall be truncated. If the socket is connected, the msg_name
11730 and msg_namelen members shall be ignored.
11731
11732 *Rationale:* Austin Group Defect Report(s) applied: 464.
11733 See http://austingroupbugs.net/view.php?id=464

Change Number: XSH/TC1/D5/0505 [305]

On Page: 1769 Line: 56541 Section: regcomp()

In the DESCRIPTION section, change from:

The following constants are defined as error return values:

to:

The following constants are defined as the minimum set of error return values, although other errors listed as implementation extensions in <regex.h> are possible:

Rationale: Austin Group Defect Report(s) applied: 305.
See http://austingroupbugs.net/view.php?id=305

Change Number: XSH/TC1/D5/0506 [320]

On Page: 1774 Line: 56764 Section: remainder()

In the RETURN VALUE section, change from:

... and either a NaN (if supported), or an implementation-defined value shall be returned.

to:

... and a NaN shall be returned.

Rationale: Austin Group Defect Report(s) applied: 320.
See http://austingroupbugs.net/view.php?id=320

Change Number: XSH/TC1/D5/0507 [320]

On Page: 1779 Line: 56883 Section: remquo()

In the RETURN VALUE section, change from:

... and either a NaN (if supported), or an implementation-defined value shall be returned.

to:

... and a NaN shall be returned.

Rationale: Austin Group Defect Report(s) applied: 320.
See http://austingroupbugs.net/view.php?id=320

Change Number: XSH/TC1/D5/0508 [324]

On Page: 1783 Line: 57005 Section: rename()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of either path prefix is not a directory; ...

to:

A component of either path prefix names an existing file that is neither a directory nor a symbolic link to a directory; ...

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0509 [147]

On Page: 1783 Line: 57009 Section: rename()

In the ERRORS section, change the [ENOTDIR] error from:

... or the new argument names a nonexistent file, contains at least one non-<slash> character, and ends with one or more trailing <slash> characters.

to:

... or the old argument names an existing non-directory file and the new argument names a nonexistent file, contains at least one non-<slash> character, and ends with one or more trailing <slash> characters; or the new argument names an existing non-directory file, contains at least one non-<slash> character, and ends with one or more trailing <slash> characters.

Rationale: Austin Group Defect Report(s) applied: 147.
See http://austingroupbugs.net/view.php?id=147

Change Number: XSH/TC1/D5/0510 [379]

On Page: 1783 Line: 57039 Section: rename()

In the ERRORS section, [ETXTBSY] error, change from:

The file to be renamed is a pure procedure (shared text) file that is being executed.

to:

The file named by new exists and is the last directory entry to a pure procedure (shared text) file that is being executed.

Rationale: Austin Group Defect Report(s) applied: 379.
See http://austingroupbugs.net/view.php?id=379

Change Number: XSH/TC1/D5/0511 [278]

On Page: 1783 Line: 57030 Section: rename()

In the ERRORS section, add (after the [EBADF] error):

[ENOTDIR] The old or new argument is not an absolute path and oldfd or newfd, respectively, is a file descriptor associated with a non-directory file.

Rationale: Austin Group Defect Report(s) applied: 278.
See http://austingroupbugs.net/view.php?id=278

Change Number: XSH/TC1/D5/0512 [278]

On Page: 1783 Line: 57041 Section: rename()

In the ERRORS section, delete:

The renameat() function may fail if:

[ENOTDIR] The old argument is not an absolute path and oldfd is neither AT_FDCWD nor a file descriptor associated with a directory, or the new argument is not an absolute path and newfd is neither AT_FDCWD nor a file descriptor associated with a directory.

Rationale: Austin Group Defect Report(s) applied: 278.
See http://austingroupbugs.net/view.php?id=278

Change Number: XSH/TC1/D5/0513 [14]

On Page: 1786 Line: 57142 Section: rewind()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement.

Change Number: XSH/TC1/D5/0514 [346]

On Page: 1788 Line: 57201 Section: rint()

In the DESCRIPTION section, add to the paragraph at P1788 L57201:

If the current rounding mode rounds towards zero, then rint() shall be equivalent to trunc(). [MX]If the current rounding mode rounds towards nearest, then rint() differs from round() in that halfway cases are rounded to even rather than away from zero.[/MX]

Rationale: Austin Group Defect Report(s) applied: 346.
See http://austingroupbugs.net/view.php?id=346

Change Number: XSH/TC1/D5/0515 [346]

On Page: 1788 Line: 57211 Section: rint()

In the RETURN VALUE section, at L57211, add:

[MX]The result shall have the same sign as x.[/MX]

Rationale: Austin Group Defect Report(s) applied: 346.
See http://austingroupbugs.net/view.php?id=346

Change Number: XSH/TC1/D5/0516 [346]

On Page: 1788 Line: 57214-57216 Section: rint()

In the RETURN VALUE section, delete L57214-57216.

Rationale: Austin Group Defect Report(s) applied: 346.
See http://austingroupbugs.net/view.php?id=346

Change Number: XSH/TC1/D5/0517 [346]

On Page: 1788 Line: 57218-57223 Section: rint()

In the ERRORS section, replace the Range Error on L57218-57223 with:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 346.
See http://austingroupbugs.net/view.php?id=346

Change Number: XSH/TC1/D5/0518 [346]

On Page: 1789 Line: 57227-57228 Section: rint()

In the APPLICATION USAGE section, change from:

On error, the expressions (math_errhandling & MATH_ERRNO) and
(math_errhandling & MATH_ERREXCEPT) are independent of each other,
but at least one of them must be non-zero.

to:

The integral value returned by these functions need not be expressible
as an intmax_t. The return value should be tested before assigning it
to an integer type to avoid the undefined results of an integer overflow.

Rationale: Austin Group Defect Report(s) applied: 346.
See http://austingroupbugs.net/view.php?id=346

Change Number: XSH/TC1/D5/0519 [324]

On Page: 1791 Line: 57290 Section: rmdir()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of path is not a directory.

to:

A component of path names an existing file that is neither
a directory nor a symbolic link to a directory.

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the
standard.

Change Number: XSH/TC1/D5/0520 [346]

On Page: 1793 Line: 57367 Section: round()

For round(), delete P1793 L57367-57370 (DESCRIPTION), L57375-57377
(RETURN VALUE), and replace L57379-57384 (ERRORS) with:

No errors are defined.

At L57372 (RETURN VALUE), add:

[MX]The result shall have the same sign as x.[/MX]

Replace L57388-57389 (APPLICATION USAGE) with:

The integral value returned by these functions need not be expressible
as an intmax_t. The return value should be tested before assigning it
to an integer type to avoid the undefined results of an integer overflow.

These functions may raise the inexact floating-point exception if the
result differs in value from the argument.

Rationale: Austin Group Defect Report(s) applied: 346.
See http://austingroupbugs.net/view.php?id=346

Change Number: XSH/TC1/D5/0521 [68]

On Page: 1795 Line: 57424 Section: scalbln()

In the RETURN VALUE section, change from:

If the correct value would cause underflow, and is not representable,
a range error may occur, and [MX]either 0.0 (if supported), or[/MX]
an implementation-defined value shall be returned.

to:

If the correct value would cause underflow, [MXX]and is not
representable[/MXX], a range error may occur, and scalbln(),
scalblnf(), scalblnl(), scalbn(), scalbnf(), and scalbnl() shall return
[MXX]0.0, or[/MXX] (if IEC 60559 Floating-Point is not supported)
an implementation-defined value no greater in magnitude than DBL_MIN,
FLT_MIN, LDBL_MIN, DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0522 [68]

On Page: 1795 Line: 57429 Section: scalbln()

In the RETURN VALUE section, change the MX shading to MXX for:

If the correct value would cause underflow, and is representable,
a range error may occur and the correct value shall be returned.

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0523 [37]

On Page: 1812 Line: 57909 Section: sem_close()

In the APPLICATION USAGE section, change from:

The sem_close() function is part of the Semaphores option and need not
be provided on all implementations.

to:

None.

Rationale: Austin Group Defect Report(s) applied: 37.
See http://austingroupbugs.net/view.php?id=37

Change Number: XSH/TC1/D5/0524 [37]

On Page: 1814 Line: 57950 Section: sem_destroy()

In the APPLICATION USAGE section, change from:

The sem_destroy() function is part of the Semaphores option and need
not be provided on all implementations.

to:

None.

Rationale: Austin Group Defect Report(s) applied: 37.
See http://austingroupbugs.net/view.php?id=37

Change Number: XSH/TC1/D5/0525 [37]

On Page: 1816 Line: 57992 Section: sem_getvalue()

In the APPLICATION USAGE section, change from:

The sem_getvalue() function is part of the Semaphores option and need
not be provided on all implementations.

to:

None.

Rationale: Austin Group Defect Report(s) applied: 37.
See http://austingroupbugs.net/view.php?id=37

Change Number: XSH/TC1/D5/0526 [37]

On Page: 1818 Line: 58049 Section: sem_init()

In the APPLICATION USAGE section, change from:

The sem_init() function is part of the Semaphores option and need not be provided on all implementations.

to:

None.

Rationale: Austin Group Defect Report(s) applied: 37.
See http://austingroupbugs.net/view.php?id=37

Change Number: XSH/TC1/D5/0527 [37]

On Page: 1822 Line: 58156 Section: sem_open()

In the APPLICATION USAGE section, change from:

The sem_open() function is part of the Semaphores option and need not be provided on all implementations.

to:

None.

Rationale: Austin Group Defect Report(s) applied: 37.
See http://austingroupbugs.net/view.php?id=37

Change Number: XSH/TC1/D5/0528 [37]

On Page: 1823 Line: 58223 Section: sem_post()

In the APPLICATION USAGE section, change from:

The sem_post() function is part of the Semaphores option and need not be provided on all implementations.

to:

None.

Rationale: Austin Group Defect Report(s) applied: 37.
See http://austingroupbugs.net/view.php?id=37

Change Number: XSH/TC1/D5/0529 [138]

On Page: 1826 Line: 58299 Section: sem_timedwait()

In the EXAMPLES section, change the example code for the handler() function from:

static void

```
handler(int sig)
{
    write(STDOUT_FILENO, "sem_post() from handler\n", 24);
    if (sem_post(&sem) == -1) {
        write(STDERR_FILENO, "sem_post() failed\n", 18);
    _exit(EXIT_FAILURE);
    }
}

to:

static void
handler(int sig)
{
    int sav_errno = errno;
    static const char info_msg[] = "sem_post() from handler\n";
    write(STDOUT_FILENO, info_msg, sizeof info_msg - 1);
    if (sem_post(&sem) == -1) {
        static const char err_msg[] = "sem_post() failed\n";
        write(STDERR_FILENO, err_msg, sizeof err_msg - 1);
        _exit(EXIT_FAILURE);
    }
    errno = sav_errno;
}
```

Rationale: Austin Group Defect Report(s) applied: 138.
See http://austingroupbugs.net/view.php?id=138

Change Number: XSH/TC1/D5/0530 [37]

On Page: 1828 Line: 58405 Section: sem_trywait()

In the APPLICATION USAGE section, delete

The sem_trywait() and sem_wait() functions are part of the Semaphores option and need not be provided on all implementations.

Rationale: Austin Group Defect Report(s) applied: 37.
See http://austingroupbugs.net/view.php?id=37

Change Number: XSH/TC1/D5/0531 [37]

On Page: 1830 Line: 58462 Section: sem_unlink()

In the APPLICATION USAGE section, change from:

The sem_unlink() function is part of the Semaphores option and need not be provided on all implementations.

to:

None.

Rationale: Austin Group Defect Report(s) applied: 37.
See http://austingroupbugs.net/view.php?id=37

Change Number: XSH/TC1/D5/0532 [345]

On Page: 1833 Line: 58514 Section: semctl()

In the DESCRIPTION section, at L58514 [SETVAL], add a sentence before "Requires alter permission":

Also, the sem_ctime timestamp shall be set to the current time, as described in XSH Section 2.7.1.

Rationale: Austin Group Defect Report(s) applied: 345.
See http://austingroupbugs.net/view.php?id=345

Change Number: XSH/TC1/D5/0533 [345]

On Page: 1833 Line: 58526 Section: semctl()

In the DESCRIPTION section,at L58526 [SETALL], add a sentence before "Requires alter permission":

Also, the sem_ctime timestamp shall be set to the current time, as described in XSH Section 2.7.1.

Rationale: Austin Group Defect Report(s) applied: 345.
See http://austingroupbugs.net/view.php?id=345

Change Number: XSH/TC1/D5/0534 [345]

On Page: 1834 Line: 58541 Section: semctl()

In the DESCRIPTION section, after L58541 [IPC_SET], add a sentence:

The sem_ctime timestamp shall be set to the current time, as described in XSH Section 2.7.1.

Rationale: Austin Group Defect Report(s) applied: 345.
See http://austingroupbugs.net/view.php?id=345

Change Number: XSH/TC1/D5/0535 [335]

On Page: 1835 Line: 58574 Section: semctl()

In the EXAMPLES section, change from:

None.

to:

Refer to semop() (on page 1841).

Rationale: Austin Group Defect Report(s) applied: 335.
See http://austingroupbugs.net/view.php?id=335

Change Number: XSH/TC1/D5/0536 [335,439]

On Page: 1837 Line: 58642-58701 Section: semget()

12301
12302 In the EXAMPLES section, replace the entire example on L58642-58701 with:
12303
12304 Refer to semop() (on page 1841).
12305
12306 On Page: 1838 Line: 58710 Section: semget()
12307
12308 In the FUTURE DIRECTIONS section, change from:
12309
12310 None.
12311
12312 to:
12313
12314 A future version may require that the value of the semval, sempid, semncnt, and
12315 semzcnt members of all semaphores in a semaphore set be initialized to zero
12316 when a call to semget() creates a semaphore set. Many semaphore implementations
12317 already do this and it greatly simplifies what an application must do to
12318 initialize a semaphore set.
12319
12320 *Rationale:* Austin Group Defect Report(s) applied: 335,439.
12321 See http://austingroupbugs.net/view.php?id=335
12322 See http://austingroupbugs.net/view.php?id=439
12323
12324
12325 **Change Number: XSH/TC1/D5/0537** [344]
12326
12327
12328 On Page: 1838 Line: 58712 Section: semget()
12329
12330 In the SEE ALSO section, add ftok() prior to semctl().
12331
12332 *Rationale:* Austin Group Defect Report(s) applied: 344.
12333 See http://austingroupbugs.net/view.php?id=344
12334
12335
12336 **Change Number: XSH/TC1/D5/0538** [329,429]
12337
12338
12339 On Page: 1839 Line: 58742 Section: semop()
12340
12341 In the DESCRIPTION section, change the type of the sem_num member of
12342 the sembuf structure in the description of semop() from:
12343
12344 short
12345
12346 to:
12347
12348 unsigned short
12349
12350 *Rationale:* Austin Group Defect Report(s) applied: 329,429.
12351 See http://austingroupbugs.net/view.php?id=329
12352 See http://austingroupbugs.net/view.php?id=429
12353
12354
12355 **Change Number: XSH/TC1/D5/0539** [345,428]
12356
12357
12358 On Page: 1840 Line: 58793 Section: semop()
12359
12360 In the DESCRIPTION section, after L58793, add a sentence:
12361
12362 Also, the sem_otime timestamp shall be set to the current time, as
12363 described in XSH Section 2.7.1.

Rationale: Austin Group Defect Report(s) applied: 345,428.
See http://austingroupbugs.net/view.php?id=345
See http://austingroupbugs.net/view.php?id=428

Change Number: XSH/TC1/D5/0540 [329,429]

On Page: 1840 Line: 58804 Section: semop()

In the ERRORS section, for the [EFBIG] error, delete the text:

less than 0 or.

Rationale: Austin Group Defect Report(s) applied: 329,429.
See http://austingroupbugs.net/view.php?id=329
See http://austingroupbugs.net/view.php?id=429

Change Number: XSH/TC1/D5/0541 [335]

On Page: 1841-1842 Line: 58825 Section: semop()

In the EXAMPLES section, after L58825 insert:

```
/* Code to initialize semid. */
...
```

At L58841, change:

uses a call to semop() to initialize it

to:

uses calls to semctl() and semop() to initialize it

After L58844, add a paragraph:

Processes that obtain semid without creating it check that sem_otime is non-zero, to ensure that the creating process has completed the semop() initialization.

Delete L58848 (<sys/types.h>), L58850 (<sys/ipc.h>), L58854 (<unistd.h>), and L58856-58858 (<pwd.h>, <fcntl.h>, and <limits.h>).

At L58861, delete ", pfd, fv", leaving just "int semid;".

Delete L58863-58866 (lgn, filename, outstat, pw), and replace them with:

```
union semun {
        int val;
        struct semid_ds *buf;
        unsigned short *array;
} arg;
struct semid_ds ds;
```

After L58878, insert:

```
arg.val = 0;
```

Replace L58882 with:

```
12427
12428   if (semctl(semid, 0, SETVAL, arg) == -1
12429       || semop(semid, &sbuf, 1) == -1) {
12430
12431   After L58889, insert:
12432
12433   goto check_init;
12434
12435   After L58894, insert:
12436
12437   else
12438   {
12439       /* Check that semid has completed initialization. */
12440       /* An application may use a retry loop at this point rather than exiting.
12441   */
12442       check_init:
12443       arg.buf = &ds;
12444       if (semctl(semid, 0, IPC_STAT, arg) < 0) {
12445       perror("IPC error 3: semctl"); exit(1);
12446       }
12447       if (ds.sem_otime == 0) {
12448       perror("IPC error 4: semctl"); exit(1);
12449       }
12450   }
12451
12452   Rationale: Austin Group Defect Report(s) applied: 335.
12453   See http://austingroupbugs.net/view.php?id=335
12454
12455
12456   Change Number: XSH/TC1/D5/0542 [291,429]
12457
12458
12459   On Page: 1842 Line: 58864 Section: semop()
12460
12461   In the EXAMPLES section, delete "char filename[PATH_MAX+1]".
12462
12463   Rationale: Austin Group Defect Report(s) applied: 291,429.
12464   See http://austingroupbugs.net/view.php?id=291
12465   See http://austingroupbugs.net/view.php?id=429
12466   This declaration is unused in this example.
12467
12468
12469   Change Number: XSH/TC1/D5/0543 [463]
12470
12471
12472   On Page: 1845 Line: 58990 Section: send()
12473
12474   In the APPLICATION USAGE section, change from:
12475
12476   The send() function is equivalent to sendto() with a null pointer dest_len
12477   argument, and to write() if no flags are used.
12478
12479   to:
12480
12481   If the socket argument refers to a connection-mode socket, the send()
12482   function is equivalent to sendto() (with any value for the dest_addr and
12483   dest_len arguments, as they are ignored in this case). If the socket
12484   argument refers to a socket and the flags argument is 0, the send()
12485   function is equivalent to write().
12486
12487   Rationale: Austin Group Defect Report(s) applied: 463.
12488   See http://austingroupbugs.net/view.php?id=463
12489
```

Change Number: XSH/TC1/D5/0544 [324]

On Page: 1848 Line: 59085 Section: sendmsg()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of the path prefix of the pathname in the socket address is not a directory, ...

to:

A component of the path prefix of the pathname in the socket address names an existing file that is neither a directory nor a symbolic link to a directory, ...

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0545 [324]

On Page: 1852 Line: 59210 Section: sendto()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of the path prefix of the pathname in the socket address is not a directory, ...

to:

A component of the path prefix of the pathname in the socket address names an existing file that is neither a directory nor a symbolic link to a directory, ...

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0546 [397]

On Page: 1855 Line: 59282 Section: setbuf()

In the ERRORS section, change from:

No errors are defined.

to:

Although the setvbuf() interface may set errno in defined ways, the value of errno after a call to setbuf() is unspecified.

Rationale: Austin Group Defect Report(s) applied: 397.
See http://austingroupbugs.net/view.php?id=397
Existing implementations do not work to preserve errno values in the case of no error.

Change Number: XSH/TC1/D5/0547 [397]

On Page: 1855 Line: 59289 Section: setbuf()

In the APPLICATION USAGE section, add a new paragraph at the end of the section:

Since errno is not required to be unchanged on success, in order to correctly detect and possibly recover from errors, applications should use setvbuf() instead of setbuf().

Rationale: Austin Group Defect Report(s) applied: 397.
See http://austingroupbugs.net/view.php?id=397

Change Number: XSH/TC1/D5/0548 [14]

On Page: 1855 Line: 59295 Section: setbuf()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0549 [167]

On Page: 1857 Line: 59347 Section: setenv()

In the DESCRIPTION section, remove the text:

If the application modifies environ or the pointers to which it points, the behavior of setenv() is undefined.

Rationale: Austin Group Defect Report(s) applied: 167.
See http://austingroupbugs.net/view.php?id=167
The text relating to undefined behavior on modification of environ is removed as this is now covered by general text added in XBD 8.1.

Change Number: XSH/TC1/D5/0550 [185]

On Page: 1857 Line: 59356 Section: setenv()

In the ERRORS section, change from:

[EINVAL] The name argument is a null pointer, points to an empty string, or points to a string containing an '=' character.

to:

[EINVAL] The envname argument points to an empty string or points to a string containing an '=' character.

Rationale: Austin Group Defect Report(s) applied: 185.

12616 See http://austingroupbugs.net/view.php?id=185
12617
12618
12619 **Change Number: XSH/TC1/D5/0551** [167]
12620
12621
12622 On Page: 1858 Line: 59381 Section: setenv()
12623
12624 In the RATIONALE section, add to the end of the section:
12625
12626 See also the RATIONALE section in getenv().
12627
12628 *Rationale:* Austin Group Defect Report(s) applied: 167.
12629 See http://austingroupbugs.net/view.php?id=167
12630
12631
12632 **Change Number: XSH/TC1/D5/0552** [38]
12633
12634
12635 On Page: 1858 Line: 59385 Section: setenv()
12636
12637 In the SEE ALSO section, add "putenv()".
12638
12639 *Rationale:* Austin Group Defect Report(s) applied: 38.
12640 See http://austingroupbugs.net/view.php?id=38
12641
12642
12643 **Change Number: XSH/TC1/D5/0553** [302]
12644
12645
12646 On Page: 1868 Line: 59590 Section: setlocale()
12647
12648 In the DESCRIPTION section, change from:
12649
12650 The setlocale() function selects the appropriate piece of the locale of
12651 the process, as specified by the category and locale arguments, and may
12652 be used to change or query the entire locale of the process or portions
12653 thereof. The value LC_ALL for category names the entire locale of the
12654 process; other values for category name only a part of the locale of
12655 the process:
12656
12657 to:
12658
12659 The setlocale() function selects the appropriate piece of the global
12660 locale, as specified by the category and locale arguments, and can be
12661 used to change or query the entire global locale or portions thereof. The
12662 value LC_ALL for category names the entire global locale; other values
12663 for category name only a part of the global locale:
12664
12665 *Rationale:* Austin Group Defect Report(s) applied: 302.
12666 See http://austingroupbugs.net/view.php?id=302
12667 These changes were overlooked during the revision when
12668 per-thread locales were added.
12669
12670
12671 **Change Number: XSH/TC1/D5/0554** [303]
12672
12673
12674 On Page: 1868 Line: 59597 Section: setlocale()
12675
12676 In the DESCRIPTION section, replace the description of LC_MESSAGES with:
12677
12678 LC_MESSAGES Affects the affirmative and negative response expressions

12679 returned by nl_langinfo() and the way message catalogs are located. It
12680 may also affect the behavior of functions that return or write message
12681 strings.
12682
12683 (All within the CX shading; no XSI shading anywhere.)
12684
12685 *Rationale:* Austin Group Defect Report(s) applied: 303.
12686 See http://austingroupbugs.net/view.php?id=303
12687
12688
12689 **Change Number: XSH/TC1/D5/0555** [302]
12690
12691
12692 On Page: 1868 Line: 59608 Section: setlocale()
12693
12694 In the DESCRIPTION section, change from:
12695
12696 If setlocale() is not invoked, the POSIX locale is the default at entry
12697 to main().
12698
12699 to:
12700
12701 The POSIX locale is the default global locale at entry to main().
12702
12703 *Rationale:* Austin Group Defect Report(s) applied: 302.
12704 See http://austingroupbugs.net/view.php?id=302
12705 These changes were overlooked during the revision when
12706 per-thread locales were added.
12707
12708
12709 **Change Number: XSH/TC1/D5/0556** [302]
12710
12711
12712 On Page: 1868 Line: 59615 Section: setlocale()
12713
12714 In the DESCRIPTION section, change from:
12715
12716 Used to direct setlocale() to query the current internationalized
12717 environment and return the name of the locale.
12718
12719 to:
12720
12721 Directs setlocale() to query the current global locale setting and
12722 return the name of the locale if category is not LC_ALL, or a string
12723 which encodes the locale name(s) for all of the individual categories
12724 if category is LC_ALL.
12725
12726 *Rationale:* Austin Group Defect Report(s) applied: 302.
12727 See http://austingroupbugs.net/view.php?id=302
12728 These changes were overlooked during the revision when
12729 per-thread locales were added.
12730
12731
12732 **Change Number: XSH/TC1/D5/0557** [302]
12733
12734
12735 On Page: 1868 Line: 59617 Section: setlocale()
12736
12737 In the DESCRIPTION section, change from:
12738
12739 Setting all of the categories of the locale of the process is similar
12740 to successively setting each individual category of the locale of the
12741 process, except that all error checking is done before any actions

are performed. To set all the categories of the locale of the process, setlocale() is invoked as:

to:

Setting all of the categories of the global locale is similar to successively setting each individual category of the global locale, except that all error checking is done before any actions are performed. To set all the categories of the global locale, setlocale() can be invoked as:

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when per-thread locales were added.

Change Number: XSH/TC1/D5/0558 [302]

On Page: 1869 Line: 59625 Section: setlocale()

In the DESCRIPTION section, change from:

... the locale of the process ...

to:

... the global locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when per-thread locales were added.

Change Number: XSH/TC1/D5/0559 [302]

On Page: 1869 Line: 59629 Section: setlocale()

In the DESCRIPTION section, change from:

The locale state is common to all threads within a process.

to:

[CX]The global locale established using setlocale() shall only be used in threads for which no current locale has been set using uselocale() or whose current locale has been set to the global locale using uselocale(LC_GLOBAL_LOCALE).[/CX]

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when per-thread locales were added.

Change Number: XSH/TC1/D5/0560 [288]

On Page: 1869 Line: 59629 Section: setlocale()

In the DESCRIPTION section, add a new paragraph to the end of the section:

The implementation shall behave as if no function defined in this volume of POSIX.1-200x calls setlocale().

Rationale: Austin Group Defect Report(s) applied: 288.
See http://austingroupbugs.net/view.php?id=288

Change Number: XSH/TC1/D5/0561 [302]

On Page: 1869 Line: 59632 Section: setlocale()

In the RETURN VALUE section, change from:

... the locale of the process is not changed.

to:

... the global locale shall not be changed.

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when per-thread locales were added.

Change Number: XSH/TC1/D5/0562 [302]

On Page: 1869 Line: 59634 Section: setlocale()

In the RETURN VALUE section, change from:

A null pointer for locale causes setlocale() to return a pointer to the string associated with the category for the current locale of the process. The locale of the process shall not be changed.

to:

A null pointer for locale shall cause setlocale() to return a pointer to the string associated with the specified category for the current global locale. The global locale shall not be changed.

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when per-thread locales were added.

Change Number: XSH/TC1/D5/0563 [302]

On Page: 1869 Line: 59637 Section: setlocale()

In the RETURN VALUE section, change from:

... the locale of the process ...

to:

... the global locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0564 [302]

On Page: 1869 Line: 59645 Section: setlocale()

In the APPLICATION USAGE section, change from:

... the locale of the process ...

to:

... the global locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0565 [302]

On Page: 1869 Line: 59649 Section: setlocale()

In the APPLICATION USAGE section, change from:

Internationalized programs must call setlocale() to initiate a specific
language operation. This can be done by calling setlocale() as follows:

to:

Internationalized programs can initiate language operation according
to environment variable settings (see XBD Section 8.2) by calling
setlocale() as follows:

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0566 [302]

On Page: 1869 Line: 59655 Section: setlocale()

In the RATIONALE section, insert a new paragraph at the beginning of the
section:

References to the international environment or locale in the following
text relate to the global locale for the process. This can be overridden
for individual threads using uselocale().

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0567 [288]

On Page: 1869 Line: 59637 Section: setlocale()

In the RETURN VALUE section, change to:

The application shall not modify the string returned. [CX]The returned string pointer might be invalidated or[/CX] the string content might be overwritten by a subsequent call to setlocale().

Rationale: Austin Group Defect Report(s) applied: 288.
See http://austingroupbugs.net/view.php?id=288

Change Number: XSH/TC1/D5/0568 [288]

On Page: 1869 Line: 59653 Section: setlocale()

In the APPLICATION USAGE section, add a new paragraph to the end of the section:

In order to make use of different locale settings while multiple threads are running, applications should use uselocale() in preference to setlocale().

Rationale: Austin Group Defect Report(s) applied: 288.
See http://austingroupbugs.net/view.php?id=288

Change Number: XSH/TC1/D5/0569 [303]

On Page: 1871 Line: 59717 Section: setlocale()

In the SEE ALSO section, add catopen(), perror(), and psignal().

Rationale: Austin Group Defect Report(s) applied: 303.
See http://austingroupbugs.net/view.php?id=303

Change Number: XSH/TC1/D5/0570 [421]

On Page: 1887 Line: 60039 Section: setsid()

In the RETURN VALUE section, change from:

Otherwise, it shall return (pid_t)-1 and ...

to:

Otherwise, it shall return -1 and ...

Rationale: Austin Group Defect Report(s) applied: 421.
See http://austingroupbugs.net/view.php?id=421

Change Number: XSH/TC1/D5/0571 [369]

On Page: 1889 Line: 60092 Section: setsockopt()

Insert a new sentence between the two sentences in the last paragraph of the DESCRIPTION:

If option_name is equal to SO_RCVTIMEO or SO_SNDTIMEO and the implementation supports setting the option, it is unspecified whether the struct timeval pointed to by option_value is stored as provided by this function or is rounded up to align with the resolution of the clock being used.

Rationale: Austin Group Defect Report(s) applied: 369.
See http://austingroupbugs.net/view.php?id=369

Change Number: XSH/TC1/D5/0572 [345]

On Page: 1905 Line: 60590 Section: shmat()

In the RETURN VALUE section, after L60590, add a sentence:

Also, the shm_atime timestamp shall be set to the current time, as described in XSH Section 2.7.1.

Rationale: Austin Group Defect Report(s) applied: 345.
See http://austingroupbugs.net/view.php?id=345

Change Number: XSH/TC1/D5/0573 [345]

On Page: 1907 Line: 60648 Section: shmctl()

In the DESCRIPTION section [IPC_SET], after L60648, add a sentence:

Also, the shm_ctime timestamp shall be set to the current time, as described in XSH Section 2.7.1.

Rationale: Austin Group Defect Report(s) applied: 345.
See http://austingroupbugs.net/view.php?id=345

Change Number: XSH/TC1/D5/0574 [345]

On Page: 1909 Line: 60710 Section: shmdt()

In the RETURN VALUE section, after L60710, add a sentence:

Also, the shm_dtime timestamp shall be set to the current time, as described in XSH Section 2.7.1.

Rationale: Austin Group Defect Report(s) applied: 345.
See http://austingroupbugs.net/view.php?id=345

Change Number: XSH/TC1/D5/0575 [345]

On Page: 1911 Line: 60760 Section: shmget()

In the DESCRIPTION section, change from:

... set equal to the current time

to:

... set to the current time, as described in XSH Section 2.7.1

Rationale: Austin Group Defect Report(s) applied: 345.
See http://austingroupbugs.net/view.php?id=345

Change Number: XSH/TC1/D5/0576 [363]

On Page: 1911 Line: 60774-60776 Section: shmget()

In the ERRORS section, [EINVAL] error:

Remove the words "and size is not 0" from the description of [EINVAL].

Rationale: Austin Group Defect Report(s) applied: 363.
See http://austingroupbugs.net/view.php?id=363
The shmget() function shall fail with error [EINVAL] if "No shared memory segment is to be created and a shared memory segment exists for key but the size of the segment associated with it is less than size and size is not 0". However, if the size of the segment is less than the size argument, then the size argument is guaranteed to be non-zero, as both are unsigned values. Thus, the last condition is redundant and should be removed.

Change Number: XSH/TC1/D5/0577 [344]

On Page: 1912 Line: 60797 Section: shmget()

In the SEE ALSO section, add ftok() prior to shmat().

Rationale: Austin Group Defect Report(s) applied: 344.
See http://austingroupbugs.net/view.php?id=344

Change Number: XSH/TC1/D5/0578 [66]

On Page: 1919 Line: 61038 Section: sigaction()

In the APPLICATION USAGE section, remove the paragraph:

If the signal occurs other than as the result of calling abort(), kill(), or raise(), the behavior is undefined if the signal handler calls any function in the standard library other than one of the functions listed in the table of sync-signal-safe functions in Section 2.4.3, or refers to any object other than errno with static storage duration other than by assigning a value to a static storage duration variable of type volatile sig_atomic_t. Unless all signal handlers have errno set on return as it was on entry, the value of errno is unspecified.

Rationale: Austin Group Defect Report(s) applied: 66.
See http://austingroupbugs.net/view.php?id=66

Change Number: XSH/TC1/D5/0579 [140]

On Page: 1919 Line: 61064 Section: sigaction()

In the APPLICATION USAGE section, add a new a paragraph at the end of the section:

See also the rationale for Realtime Signal Generation and Delivery in Section B.2.4.2.

Rationale: Austin Group Defect Report(s) applied: 140.
See http://austingroupbugs.net/view.php?id=140

Change Number: XSH/TC1/D5/0580 [275]

On Page: 1937 Line: 61633 Section: signal()

In the DESCRIPTION section, change from:

If the signal occurs as the result of calling the abort(), raise(), [CX]kill(), pthread_kill(), or sigqueue()[/CX] function, the signal handler shall not call the raise() function.

to:

The C Standard places a restriction on applications relating to the use of raise() from signal handlers. [CX]This restriction does not apply to POSIX applications, as POSIX.1-2008 requires raise() to be async-signal-safe (see Section 2.4.3 Signal Actions).[/CX]

Rationale: Austin Group Defect Report(s) applied: 275.
See http://austingroupbugs.net/view.php?id=275

Change Number: XSH/TC1/D5/0581 [66]

On Page: 1937 Line: 61635 Section: signal()

In the DESCRIPTION section, change from:

If the signal occurs other than as the result of calling abort(), raise(), [CX]kill(), pthread_kill(), or sigqueue()[/CX], the behavior is undefined if the signal handler refers to any object with static storage duration other than by assigning a value to an object declared as volatile sig_atomic_t, or if the signal handler calls any function in the standard library other than one of the functions listed in Section 2.4. Furthermore, if such a call fails, the value of errno is unspecified.

to:

[CX]If the process is multi-threaded[/CX], or if the process is single threaded and a signal handler is executed other than as the result of:

* The process calling abort(), raise(), [CX]kill(), pthread_kill(), or sigqueue()[/CX] to generate a signal that is not blocked

* [CX]A pending signal being unblocked and being delivered before the call that unblocked it returns[/CX]

the behavior is undefined if the signal handler refers to any object [CX]other than errno[/CX] with static storage duration other than by assigning a value to an object declared as volatile sig_atomic_t, or if the signal handler calls any function defined in this standard other than [CX]one of the functions listed in Section 2.4[/CX].

Rationale: Austin Group Defect Report(s) applied: 66.
See http://austingroupbugs.net/view.php?id=66

Change Number: XSH/TC1/D5/0582 [105]

On Page: 1937 Line: 61644 Section: signal()

In the DESCRIPTION section, add a new paragraph at the end of the DESCRIPTION:

The signal() function shall not change the setting of errno if successful.

Rationale: Austin Group Defect Report(s) applied: 105.
See http://austingroupbugs.net/view.php?id=105

Change Number: XSH/TC1/D5/0583 [392]

On Page: 1952 Line: 62027 Section: sigtimedwait()

In the APPLICATION USAGE section, add a new paragraph to the end of the section:

Note that in order to ensure generated signals are queued and signal values passed to sigqueue() are available in si_value, applications which use sigwaitinfo() or sigtimedwait() need to set the SA_SIGINFO flag for each signal in the set (see 2.4). This means setting each signal to be handled by a three-argument signal catching function, even if the handler will never be called. It is not possible (portably) to set a signal handler to SIG_DFL while setting the SA_SIGINFO flag, because assigning to the sa_handler member of struct sigaction instead of the sa_sigaction member would result in undefined behavior, and SIG_DFL need not be assignment compatible with sa_sigaction. Even if an assignment of SIG_DFL to sa_sigaction is accepted by the compiler, the implementation need not treat this value as special - it could just be taken as the address of a signal catching function.

Rationale: Austin Group Defect Report(s) applied: 392.
See http://austingroupbugs.net/view.php?id=392

Change Number: XSH/TC1/D5/0584 [76]

On Page: 1955 Line: 62165 Section: sigwait()

In the RATIONALE section, change from:

After some consideration, threads were allowed to use semaphores and sem_post() was defined to be async-signal and async-cancel-safe.

to:

After some consideration, threads were allowed to use semaphores and

sem_post() was defined to be async-signal-safe.

Rationale: Austin Group Defect Report(s) applied: 76.
See http://austingroupbugs.net/view.php?id=76

Change Number: XSH/TC1/D5/0585 [68]

On Page: 1958 Line: 62216 Section: sin()

In the RETURN VALUE section, change from:

[MX]If x is subnormal, a range error may occur and x should be returned.[/MX]

to:

[MX]If x is subnormal, a range error may occur[/MX] [MXX]and x should be returned.[/MXX]

[MX]If x is not returned, sin(), sinf(), and sinl() shall return an implementation-defined value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.[/MX]

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0586 [320]

On Page: 1958 Line: 62217 Section: sin()

In the RETURN VALUE section, change from:

... and either a NaN (if supported), or an implementation-defined value shall be returned.

to:

... and a NaN shall be returned.

Rationale: Austin Group Defect Report(s) applied: 320.
See http://austingroupbugs.net/view.php?id=320

Change Number: XSH/TC1/D5/0587 [68]

On Page: 1960 Line: 62285 Section: sinh()

In the RETURN VALUE section, change from:

[MX]If x is subnormal, a range error may occur and x should be returned.[/MX]

to:

[MX]If x is subnormal, a range error may occur[/MX] [MXX]and x should be returned.[/MXX]

[MX]If x is not returned, sinh(), sinhf(), and sinhl() shall return

an implementation-defined value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.[/MX]

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0588 [320]

On Page: 1973 Line: 62659 Section: sqrt()

In the RETURN VALUE section, change from:

... and either a NaN (if supported), or an implementation-defined value shall be returned.

to:

... and a NaN shall be returned.

Rationale: Austin Group Defect Report(s) applied: 320.
See http://austingroupbugs.net/view.php?id=320

Change Number: XSH/TC1/D5/0589 [302]

On Page: 1985 Line: 62813 Section: strcasecmp()

In the DESCRIPTION section, change from:

... current locale of the process ...

to:

... current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when per-thread locales were added.

Change Number: XSH/TC1/D5/0590 [294]

On Page: 1985 Line: 62817 Section: strcasecmp()

In the DESCRIPTION section, change from:

When the LC_CTYPE category of the current locale is from the POSIX locale, strcasecmp() and strncasecmp() shall ...

to:

When the LC_CTYPE category of the locale being used is from the POSIX locale, these functions shall ...

Rationale: Austin Group Defect Report(s) applied: 294.
See http://austingroupbugs.net/view.php?id=294

Change Number: XSH/TC1/D5/0591 [283]

On Page: 1985 Line: 62819 Section: strcasecmp()

In the DESCRIPTION section, add a new paragraph to the end of the section:

The behavior is undefined if the locale argument to strcasecmp_l() or strncasecmp_l() is the special locale object LC_GLOBAL_LOCALE or is not a valid locale object handle.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0592 [283]

On Page: 1985 Line: 62829 Section: strcasecmp()

In the ERRORS section, change from:

The strcasecmp_l() and strncasecmp_l() functions may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0593 [283]

On Page: 1991 Line: 62993 Section: strcoll()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to strcoll_l() is the special locale object LC_GLOBAL_LOCALE or is not a valid locale object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0594 [283]

On Page: 1991 Line: 63007 Section: strcoll()

In the ERRORS section, delete:

The strcoll_l() function may fail if:

[EINVAL] Locale is not a valid locale object handle.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0595 [75]

On Page: 1999 Line: 63237 Section: strerror()

In the DESCRIPTION section, change from:

The string pointed to shall not be modified by the application.
The string may be overwritten by a subsequent call to strerror().
[CX]The string may be overwritten by a subsequent call to strerror_l()
in the same thread.[/CX]

to:

The application shall not modify the string returned. [CX]The returned
string pointer might be invalidated or[/CX] the string content might be
overwritten by a subsequent call to strerror(), [CX]or by a subsequent
call to strerror_l() in the same thread[/CX].

Rationale: Austin Group Defect Report(s) applied: 75.
See http://austingroupbugs.net/view.php?id=75

Change Number: XSH/TC1/D5/0596 [447]

On Page: 1999 Line: 63244 Section: strerror()

In the DESCRIPTION section, change from:

The strerror() and strerror_l() functions shall not change the setting
of errno if successful.

Since no return value is reserved to indicate an error, an application
wishing to check for error situations should set errno to 0, then call
strerror(), then check errno.

to:

The strerror() and strerror_l() functions shall not change the setting
of errno if successful.

Since no return value is reserved to indicate an error of strerror(), an
application wishing to check for error situations should set errno to 0,
then call strerror(), then check errno. Similarly, since strerror_l()
is required to return a string for some errors, an application wishing
to check for all error situations should set errno to 0, then call
strerror_l(), then check errno.

Rationale: Austin Group Defect Report(s) applied: 447.
See http://austingroupbugs.net/view.php?id=447

Change Number: XSH/TC1/D5/0597 [382,428]

On Page: 1999 Line: 63253 Section: strerror()

In the DESCRIPTION section, change from (retaining the CX shading):

If the value of errnum is a valid error number, the message string shall
indicate what error occurred; otherwise, if these functions complete

13498 successfully, the message string shall indicate that an unknown error
13499 occurred.
13500
13501 to:
13502
13503 If the value of errnum is a valid error number, the message string
13504 shall indicate what error occurred; if the value of errnum is zero,
13505 the message string shall either be an empty string or indicate that no
13506 error occurred; otherwise, if these functions complete successfully,
13507 the message string shall indicate that an unknown error occurred.
13508
13509 *Rationale:* Austin Group Defect Report(s) applied: 382,428.
13510 See http://austingroupbugs.net/view.php?id=382
13511 See http://austingroupbugs.net/view.php?id=428
13512 While the standard does not currently require it, the intent
13513 was that strerror(0) should be required to succeed, and there are existing
13514 applications that rely on this behavior.
13515
13516
13517 **Change Number: XSH/TC1/D5/0598** [283]
13518
13519
13520 On Page: 1999 Line: 63255 Section: strerror()
13521
13522 In the DESCRIPTION section, add a new paragraph to the end of the section:
13523
13524 [CX]The behavior is undefined if the locale argument to strerror_l()
13525 is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
13526 object handle.[/CX]
13527
13528 *Rationale:* Austin Group Defect Report(s) applied: 283.
13529 See http://austingroupbugs.net/view.php?id=283
13530
13531
13532 **Change Number: XSH/TC1/D5/0599** [382,428]
13533
13534
13535 On Page: 2000 Line: 63267 Section: strerror()
13536
13537 In the ERRORS section for the first [EINVAL] error, change from:
13538
13539 ... is not a valid error number.
13540
13541 to:
13542
13543 ... is neither a valid error number nor zero.
13544
13545 *Rationale:* Austin Group Defect Report(s) applied: 382,428.
13546 See http://austingroupbugs.net/view.php?id=382
13547 See http://austingroupbugs.net/view.php?id=428
13548 While the standard does not currently require it, the intent
13549 was that strerror(0) should be required to succeed, and there are existing
13550 applications that rely on this behavior.
13551
13552
13553 **Change Number: XSH/TC1/D5/0600** [283]
13554
13555
13556 On Page: 2000 Line: 63268 Section: strerror()
13557
13558 In the ERRORS section, delete:
13559
13560 The strerror_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0601 [382,428]

On Page: 2000 Line: 63303 Section: strerror()

In the RATIONALE section, add a new paragraph to the end of the section:

Some applications rely on being able to set errno to 0 before calling a function with no reserved value to indicate an error, then call strerror(errno) afterwards to detect whether an error occurred (because errno changed) or indicate success (because errno remained zero). This usage pattern requires that strerror(0) succeed with useful results. Previous versions of the standard did not specify the behavior when errnum is zero.

Rationale: Austin Group Defect Report(s) applied: 382,428.
See http://austingroupbugs.net/view.php?id=382
See http://austingroupbugs.net/view.php?id=428
While the standard does not currently require it, the intent was that strerror(0) should be required to succeed, and there are existing applications that rely on this behavior.

Change Number: XSH/TC1/D5/0602 [302]

On Page: 2004 Line: 63413 Section: strfmon()

change from:

... the locale of the process ...

to:

... the current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when per-thread locales were added.

Change Number: XSH/TC1/D5/0603 [283]

On Page: 2004 Line: 63418 Section: strfmon()

In the DESCRIPTION section, add a new paragraph to the end of the section:

The behavior is undefined if the locale argument to strfmon_l() is the special locale object LC_GLOBAL_LOCALE or is not a valid locale object handle.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0604 [283]

On Page: 2004 Line: 63427 Section: strfmon()

In the ERRORS section, delete:

The strfmon_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0605 [283]

On Page: 2011 Line: 63682 Section: strftime()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to strftime_l() is the special locale object LC_GLOBAL_LOCALE or is not a valid locale object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0606 [283]

On Page: 2011 Line: 63689 Section: strftime()

In the ERRORS section, change from:

The strftime_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0607 [193]

On Page: 2014 Line: 63830 Section: strftime()

In the RATIONALE section, in the Conversion Specification column, change from:

"%05%Y"

to:

"%05Y"

Rationale: Austin Group Defect Report(s) applied: 193.
See http://austingroupbugs.net/view.php?id=193

Change Number: XSH/TC1/D5/0608 [193]

On Page: 2014 Line: 63832 Section: strftime()

In the RATIONALE section, in the Conversion Specification column,
change the '1' after the 'y' in %+3C%y1+12345 to a column separator
(a ! in tbl formatting notation).

Rationale: Austin Group Defect Report(s) applied: 193.
See http://austingroupbugs.net/view.php?id=193

Change Number: XSH/TC1/D5/0609 [75]

On Page: 2032 Line: 64344 Section: strsignal()

In the DESCRIPTION section, change from:

The string pointed to shall not be modified by the application, but may
be overwritten by a subsequent call to strsignal() or setlocale().

to:

The application shall not modify the string returned. The returned
pointer might be invalidated or the string content might be overwritten
by a subsequent call to strsignal() or setlocale().

Rationale: Austin Group Defect Report(s) applied: 75.
See http://austingroupbugs.net/view.php?id=75

Change Number: XSH/TC1/D5/0610 [302]

On Page: 2036 Line: 64492 Section: strtod()

change from:

... the locale of the process ...

to:

... the current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0611 [94]

On Page: 2036 Line: 64498 Section: strtod()

In the DESCRIPTION section, change from:

If the subject sequence is empty or does not have the expected form, no conversion shall be performed; the value of str is stored in the ...

to:

If the subject sequence is empty or does not have the expected form, no conversion shall be performed; the value of nptr is stored in the ...

Rationale: Austin Group Defect Report(s) applied: 94.
See http://austingroupbugs.net/view.php?id=94
Correct a reference to the nptr argument in the case when no conversion is performed.

Change Number: XSH/TC1/D5/0612 [105]

On Page: 2036 Line: 64500 Section: strtod()

In the DESCRIPTION section, remove the CX shading from the paragraphs:

The strtod() function shall not change the setting of errno if successful.

Since 0 is returned on error and is also a valid return on success, an application wishing to check for error situations should set errno to 0, then call strtod(), strtof(), or strtold(), then check errno.

and then change the first paragraph from:

The strtod() function shall not change the setting of errno if successful.

to:

These functions shall not change the setting of errno if successful.

Rationale: Austin Group Defect Report(s) applied: 105.
See http://austingroupbugs.net/view.php?id=105

Change Number: XSH/TC1/D5/0613 [453]

On Page: 2039 Line: 64605 Section: strtoimax()

In the RETURN VALUE section, change from:

If no conversion could be performed, zero shall be returned.

to:

If no conversion could be performed, 0 shall be returned [CX]and errno may be set to [EINVAL][/CX]. [CX]If the value of base is not supported, 0 shall be returned and errno shall be set to [EINVAL].[/CX]

Rationale: Austin Group Defect Report(s) applied: 453.
See http://austingroupbugs.net/view.php?id=453

Change Number: XSH/TC1/D5/0614 [453]

On Page: 2039 Line: 64610 Section: strtoimax()

13813
13814 In the ERRORS section, change from:
13815
13816 These functions shall fail if:
13817
13818 [ERANGE] The value to be returned is not representable.
13819
13820 These functions may fail if:
13821
13822 [EINVAL] The value of base is not supported.
13823
13824 to:
13825
13826 These functions shall fail if:
13827
13828 [EINVAL] [CX]The value of base is not supported.[/CX]
13829
13830 [ERANGE] The value to be returned is not representable.
13831
13832 These functions may fail if:
13833
13834 [EINVAL] No conversion could be performed.
13835
13836 *Rationale:* Austin Group Defect Report(s) applied: 453.
13837 See http://austingroupbugs.net/view.php?id=453
13838
13839
13840 On Page: 2039 Line: 64617 Section: strtoimax()
13841
13842 In the APPLICATION USAGE section, change from:
13843
13844 None.
13845
13846 to:
13847
13848 Since the value of *endptr is unspecified if the value of base is not
13849 supported, applications should either ensure that base has a supported
13850 value (0 or between 2 and 36) before the call, or check for an [EINVAL]
13851 error before examining *endptr.
13852
13853 *Rationale:* Austin Group Defect Report(s) applied: 453.
13854 See http://austingroupbugs.net/view.php?id=453
13855
13856
13857 **Change Number: XSH/TC1/D5/0615** [177]
13858
13859
13860 On Page: 2041 Line: 64700-64701 Section: strtok()
13861
13862 In the EXAMPLES section, change from:
13863
13864 Breaking a Line
13865
13866 The following example uses strtok() to break a line into two character
13867 strings separated by any combination of <space>, <tab>, or <newline>
13868 characters.
13869
13870 #include <string.h>
13871 ...
13872 struct element {
13873 char *key;
13874 char *data;
13875 };

```
...
char line[LINE_MAX];
char *key, *data;
...
key = strtok(line, " \n");
data = strtok(NULL, " \n");
...
```

to:

```
Find First two Fields in a Buffer

The following example uses strtok() to find two character strings (a
key and data associated with that key) separated by any combination of
<space>, <tab>, or <newline> characters at the start of the array of
characters pointed to by buffer.

#include <string.h>
...
char    *buffer;
...
struct element {
    char    *key;
    char    *data;
} e;
...
// Load the buffer...
...
// Get the key and its data...
e.key = strtok(buffer, " \t\n");
e.data = strtok(NULL, " \t\n");
// Process the rest of the contents of the buffer...
...
```

Rationale: Austin Group Defect Report(s) applied: 177.
See http://austingroupbugs.net/view.php?id=177

Change Number: XSH/TC1/D5/0616 [453]

On Page: 2044 Line: 64778 Section: strtol()

In the DESCRIPTION section, change from:

```
... the value of str is stored ...
```

to:

```
... the value of str shall be stored ...
```

Rationale: Austin Group Defect Report(s) applied: 453.
See http://austingroupbugs.net/view.php?id=453

Change Number: XSH/TC1/D5/0617 [105]

On Page: 2044 Line: 64780 Section: strtol()

In the DESCRIPTION section, remove the CX shading from the paragraphs:

```
The strtol() function shall not change the setting of errno if successful.
```

Since 0, {LONG_MIN} or {LLONG_MIN}, and {LONG_MAX} or {LLONG_MAX} are returned on error and are also valid returns on success, an application wishing to check for error situations should set errno to 0, then call strtol() or strtoll(), then check errno.

and then change the first paragraph from:

The strtol() function shall not change the setting of errno if successful.

to:

These functions shall not change the setting of errno if successful.

Rationale: Austin Group Defect Report(s) applied: 105.
See http://austingroupbugs.net/view.php?id=105

Change Number: XSH/TC1/D5/0618 [453]

On Page: 2044 Line: 64785 Section: strtol()

In the RETURN VALUE section, change from:

If no conversion could be performed, 0 shall be returned [CX]and errno may be set to [EINVAL][/CX].

to:

If no conversion could be performed, 0 shall be returned [CX]and errno may be set to [EINVAL][/CX].

[CX]If the value of base is not supported, 0 shall be returned and errno shall be set to [EINVAL].[/CX]

Rationale: Austin Group Defect Report(s) applied: 453.
See http://austingroupbugs.net/view.php?id=453

Change Number: XSH/TC1/D5/0619 [453]

On Page: 2044 Line: 64791 Section: strtol()

In the ERRORS section, change from:

These functions shall fail if:

[ERANGE] The value to be returned is not representable.

These functions may fail if:

[EINVAL] [CX] The value of base is not supported. [/CX]

to:

These functions shall fail if:

[EINVAL] [CX] The value of base is not supported. [/CX]

[ERANGE] The value to be returned is not representable.

These functions may fail if:

[EINVAL] No conversion could be performed.

Rationale: Austin Group Defect Report(s) applied: 453.
See http://austingroupbugs.net/view.php?id=453

Change Number: XSH/TC1/D5/0620 [453]

On Page: 2044 Line: 64798 Section: strtol()

In the APPLICATION USAGE section, change from:

None.

to:

Since the value of *endptr is unspecified if the value of base is not supported, applications should either ensure that base has a supported value (0 or between 2 and 36) before the call, or check for an [EINVAL] error before examining *endptr.

Rationale: Austin Group Defect Report(s) applied: 453.
See http://austingroupbugs.net/view.php?id=453

Change Number: XSH/TC1/D5/0621 [105]

On Page: 2049 Line: 64884 Section: strtoul()

In the DESCRIPTION section, remove the CX shading from the paragraphs:

The strtoul() function shall not change the setting of errno if successful.

Since 0, {ULONG_MAX}, and {ULLONG_MAX} are returned on error and are also valid returns on success, an application wishing to check for error situations should set errno to 0, then call strtoul() or strtoull(), then check errno.

and then change the first paragraph from:

The strtoul() function shall not change the setting of errno if successful.

to:

These functions shall not change the setting of errno if successful.

Rationale: Austin Group Defect Report(s) applied: 105.
See http://austingroupbugs.net/view.php?id=105

Change Number: XSH/TC1/D5/0622 [453]

On Page: 2049 Line: 64889 Section: strtoul()

In the RETURN VALUE section, change from:

If no conversion could be performed, 0 shall be returned [CX]and errno may be set to [EINVAL][/CX].

14065
14066 to:
14067
14068 If no conversion could be performed, 0 shall be returned [CX]and
14069 errno may be set to [EINVAL][/CX].
14070
14071 [CX]If the value of base is not supported, 0 shall be returned and
14072 errno shall be set to [EINVAL].[/CX]
14073
14074 *Rationale:* Austin Group Defect Report(s) applied: 453.
14075 See http://austingroupbugs.net/view.php?id=453
14076
14077
14078 **Change Number: XSH/TC1/D5/0623** [453]
14079
14080
14081 On Page: 2049 Line: 64902 Section: strtoul()
14082
14083 In the APPLICATION USAGE section, change from:
14084
14085 None.
14086
14087 to:
14088
14089 Since the value of *endptr is unspecified if the value of base is
14090 not supported, applications should either ensure that base has a
14091 supported value (0 or between 2 and 36) before the call, or check
14092 for an [EINVAL] error before examining *endptr.
14093
14094 *Rationale:* Austin Group Defect Report(s) applied: 453.
14095 See http://austingroupbugs.net/view.php?id=453
14096
14097
14098 **Change Number: XSH/TC1/D5/0624** [283]
14099
14100
14101 On Page: 2052 Line: 64952 Section: strxfrm()
14102
14103 In the DESCRIPTION section, add a new paragraph to the end of the section:
14104
14105 [CX]The behavior is undefined if the locale argument to strxfrm_l()
14106 is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
14107 object handle.[/CX]
14108
14109 *Rationale:* Austin Group Defect Report(s) applied: 283.
14110 See http://austingroupbugs.net/view.php?id=283
14111
14112
14113 **Change Number: XSH/TC1/D5/0625** [283]
14114
14115
14116 On Page: 2052 Line: 64963 Section: strxfrm()
14117
14118 In the ERRORS section, delete:
14119
14120 The strxfrm_l() function may fail if:
14121
14122 [EINVAL] locale is not a valid locale object handle.
14123
14124 *Rationale:* Austin Group Defect Report(s) applied: 283.
14125 See http://austingroupbugs.net/view.php?id=283
14126
14127

Change Number: XSH/TC1/D5/0626 [302]

On Page: 2052 Line: 64969 Section: strxfrm()

change from:

... the locale of the process ...

to:

... the current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0627 [146,428]

On Page: 2057 Line: 65056 Section: symlink()

In the DESCRIPTION section, add a new paragraph after the existing
third paragraph:

If path2 names a symbolic link, symlink() shall fail and set errno to
[EEXIST].

Rationale: Austin Group Defect Report(s) applied: 146,428.
See http://austingroupbugs.net/view.php?id=146
See http://austingroupbugs.net/view.php?id=428

Change Number: XSH/TC1/D5/0628 [461]

On Page: 2057 Line: 65077 Section: symlink()

In the DESCRIPTION section, change from:

... the current working directory is used ...

to:

... the current working directory shall be used ...

Rationale: Austin Group Defect Report(s) applied: 461.
See http://austingroupbugs.net/view.php?id=461

Change Number: XSH/TC1/D5/0629 [146,428]

On Page: 2058 Line: 65087 Section: symlink()

In the ERRORS section, change the [EEXIST] error from:

[EEXIST] The path2 argument names an existing file or symbolic link.

to:

[EEXIST] The path2 argument names an existing file.

Rationale: Austin Group Defect Report(s) applied: 146,428.
See http://austingroupbugs.net/view.php?id=146
See http://austingroupbugs.net/view.php?id=428

Change Number: XSH/TC1/D5/0630 [146,428,436]

On Page: 2058 Line: 65095 Section: symlink()

In the ERRORS section, change the [ENOENT] error from:

[ENOENT] A component of path2 does not name an existing file or path2 is an empty string.

to:

[ENOENT] A component of the path prefix of path2 does not name an existing file or path2 is an empty string.

[ENOENT] or [ENOTDIR]
The path2 argument contains at least one non-<slash> character and ends with one or more trailing <slash> characters. If path2 names an existing file, an [ENOENT] error shall not occur.

Rationale: Austin Group Defect Report(s) applied: 146,428,436.
See http://austingroupbugs.net/view.php?id=146
See http://austingroupbugs.net/view.php?id=428
See http://austingroupbugs.net/view.php?id=436

Change Number: XSH/TC1/D5/0631 [324]

On Page: 2058 Line: 65102 Section: symlink()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of the path prefix of path2 is not a directory.

to:

A component of the path prefix of path2 names an existing file that is neither a directory nor a symbolic link to a directory.

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0632 [278]

On Page: 2058 Line: 65108 Section: symlink()

In the ERRORS section, add (after the [EBADF] error):

[ENOTDIR] The path2 argument is not an absolute path and fd is a file descriptor associated with a non-directory file.

Rationale: Austin Group Defect Report(s) applied: 278.
See http://austingroupbugs.net/view.php?id=278

Change Number: XSH/TC1/D5/0633 [278]

On Page: 2058 Line: 65116 Section: symlink()

In the ERRORS section, delete:

The symlinkat() function may fail if:

[ENOTDIR] The path2 argument is not an absolute path and fd is neither AT_FDCWD nor a file descriptor associated with a directory.

Rationale: Austin Group Defect Report(s) applied: 278.
See http://austingroupbugs.net/view.php?id=278

Change Number: XSH/TC1/D5/0634 [151]

On Page: 2059 Line: 65129 Section: symlink()

In the RATIONALE section, delete the first sentence:

Since POSIX.1-2008 does not require any association of file times with symbolic links, there is no requirement that file times be updated by symlink().

Rationale: Austin Group Defect Report(s) applied: 151.
See http://austingroupbugs.net/view.php?id=151

Change Number: XSH/TC1/D5/0635 [68]

On Page: 2074 Line: 65706 Section: tan()

In the RETURN VALUE section, change from:

If the correct value would cause underflow, and is not representable, a range error may occur, and [MX]either 0.0 (if supported), or[/MX] an implementation-defined value shall be returned.

to:

If the correct value would cause underflow, [MXX]and is not representable[/MXX], a range error may occur, and tan(), tanf(), and tanl() shall return [MXX]0.0, or[/MXX] (if IEC 60559 Floating-Point is not supported) an implementation-defined value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0636 [68]

On Page: 2074 Line: 65710 Section: tan()

In the RETURN VALUE section, change from:

[MX]If x is subnormal, a range error may occur and x should be returned.[/MX]

to:

[MX]If x is subnormal, a range error may occur[/MX] [MXX]and x should be returned.[/MXX]

[MX]If x is not returned, tan(), tanf(), and tanl() shall return an implementation-defined value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.[/MX]

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0637 [68]

On Page: 2074 Line: 65713 Section: tan()

In the RETURN VALUE section, change the MX shading to MXX for:

If the correct value would cause underflow, and is representable, a range error may occur and the correct value shall be returned.

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0638 [68]

On Page: 2076 Line: 65792 Section: tanh()

In the RETURN VALUE section, change from:

[MX]If x is subnormal, a range error may occur and x should be returned.[/MX]

to:

[MX]If x is subnormal, a range error may occur[/MX] [MXX]and x should be returned.[/MXX]

[MX]If x is not returned, tanh(), tanhf(), and tanhl() shall return an implementation-defined value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.[/MX]

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0639 [79]

On Page: 2079 Line: 65840 Section: tcdrain()

In the DESCRIPTION section change from:

If the calling process is blocking or ignoring SIGTTOU signals ...

to:

If the calling thread is blocking SIGTTOU signals or the process is ignoring SIGTTOU signals ...

Rationale: Austin Group Defect Report(s) applied: 79.
See http://austingroupbugs.net/view.php?id=79

Change Number: XSH/TC1/D5/0640 [79]

On Page: 2079 Line: 65848 Section: tcdrain()

In the ERRORS section, add:

[EIO] The process group of the writing process is orphaned, the calling thread is not blocking SIGTTOU, and the process is not ignoring SIGTTOU.

Rationale: Austin Group Defect Report(s) applied: 79.
See http://austingroupbugs.net/view.php?id=79

Change Number: XSH/TC1/D5/0641 [79]

On Page: 2079 Line: 65850 Section: tcdrain()

In the ERRORS section, delete:

The [...] function may fail if:

[EIO] The process group of the writing process is orphaned, and the writing process is not ignoring or blocking SIGTTOU.

Rationale: Austin Group Defect Report(s) applied: 79.
See http://austingroupbugs.net/view.php?id=79

Change Number: XSH/TC1/D5/0642 [79]

On Page: 2081 Line: 65895 Section: tcflow()

In the DESCRIPTION section change from:

If the calling process is blocking or ignoring SIGTTOU signals ...

to:

If the calling thread is blocking SIGTTOU signals or the process is ignoring SIGTTOU signals ...

Rationale: Austin Group Defect Report(s) applied: 79.
See http://austingroupbugs.net/view.php?id=79

Change Number: XSH/TC1/D5/0643 [79]

On Page: 2081 Line: 65903 Section: tcflow()

In the ERRORS section, add:

[EIO] The process group of the writing process is orphaned, the calling thread is not blocking SIGTTOU, and the process is not ignoring SIGTTOU.

Rationale: Austin Group Defect Report(s) applied: 79.
See http://austingroupbugs.net/view.php?id=79

Change Number: XSH/TC1/D5/0644 [79]

On Page: 2081 Line: 65905 Section: tcflow()

In the ERRORS section, delete:

The [...] function may fail if:

[EIO] The process group of the writing process is orphaned, and the writing process is not ignoring or blocking SIGTTOU.

Rationale: Austin Group Defect Report(s) applied: 79.
See http://austingroupbugs.net/view.php?id=79

Change Number: XSH/TC1/D5/0645 [79]

On Page: 2083 Line: 65943 Section: tcflush()

In the DESCRIPTION section change from:

If the calling process is blocking or ignoring SIGTTOU signals ...

to:

If the calling thread is blocking SIGTTOU signals or the process is ignoring SIGTTOU signals ...

Rationale: Austin Group Defect Report(s) applied: 79.
See http://austingroupbugs.net/view.php?id=79

Change Number: XSH/TC1/D5/0646 [79]

On Page: 2083 Line: 65951 Section: tcflush()

In the ERRORS section, add:

[EIO] The process group of the writing process is orphaned, the calling thread is not blocking SIGTTOU, and the process is not ignoring SIGTTOU.

Rationale: Austin Group Defect Report(s) applied: 79.
See http://austingroupbugs.net/view.php?id=79

Change Number: XSH/TC1/D5/0647 [79]

On Page: 2083 Line: 65953 Section: tcflush()

In the ERRORS section, delete:

The [...] function may fail if:

[EIO] The process group of the writing process is orphaned, and the writing process is not ignoring or blocking SIGTTOU.

Rationale: Austin Group Defect Report(s) applied: 79.
See http://austingroupbugs.net/view.php?id=79

Change Number: XSH/TC1/D5/0648 [421]

On Page: 2089 Line: 66089 Section: tcgetsid()

In the RETURN VALUE section, change from:

Otherwise, a value of (pid_t)-1 shall be returned, ...

to:

Otherwise, a value of -1 shall be returned, ...

Rationale: Austin Group Defect Report(s) applied: 421.
See http://austingroupbugs.net/view.php?id=421

Change Number: XSH/TC1/D5/0649 [79]

On Page: 2090 Line: 66132 Section: tcsendbreak()

In the DESCRIPTION section change from:

If the calling process is blocking or ignoring SIGTTOU signals ...

to:

If the calling thread is blocking SIGTTOU signals or the process is ignoring SIGTTOU signals ...

Rationale: Austin Group Defect Report(s) applied: 79.
See http://austingroupbugs.net/view.php?id=79

Change Number: XSH/TC1/D5/0650 [79]

On Page: 2090 Line: 66139 Section: tcsendbreak()

In the ERRORS section, add:

[EIO] The process group of the writing process is orphaned, the calling thread is not blocking SIGTTOU, and the process is not ignoring SIGTTOU.

Rationale: Austin Group Defect Report(s) applied: 79.
See http://austingroupbugs.net/view.php?id=79

Change Number: XSH/TC1/D5/0651 [79]

On Page: 2090 Line: 66141 Section: tcsendbreak()

In the ERRORS section, delete:

The [...] function may fail if:

[EIO] The process group of the writing process is orphaned, and the writing process is not ignoring or blocking SIGTTOU.

Rationale: Austin Group Defect Report(s) applied: 79.
See http://austingroupbugs.net/view.php?id=79

Change Number: XSH/TC1/D5/0652 [79]

On Page: 2092 Line: 66203 Section: tcsetattr()

In the DESCRIPTION section change from:

If the calling process is blocking or ignoring SIGTTOU signals ...

to:

If the calling thread is blocking SIGTTOU signals or the process is ignoring SIGTTOU signals ...

Rationale: Austin Group Defect Report(s) applied: 79.
See http://austingroupbugs.net/view.php?id=79

Change Number: XSH/TC1/D5/0653 [79]

On Page: 2093 Line: 66215 Section: tcsetattr()

In the ERRORS section, add:

[EIO] The process group of the writing process is orphaned, the calling thread is not blocking SIGTTOU, and the process is not ignoring SIGTTOU.

Rationale: Austin Group Defect Report(s) applied: 79.
See http://austingroupbugs.net/view.php?id=79

Change Number: XSH/TC1/D5/0654 [79]

On Page: 2093 Line: 66217 Section: tcsetattr()

In the ERRORS section, delete:

The [...] function may fail if:

[EIO] The process group of the writing process is orphaned, and the writing process is not ignoring or blocking SIGTTOU.

Rationale: Austin Group Defect Report(s) applied: 79.
See http://austingroupbugs.net/view.php?id=79

Change Number: XSH/TC1/D5/0655 [79]

On Page: 2095 Line: 66280 Section: tcsetpgrp()

In the DESCRIPTION section change from:

If the calling process is blocking or ignoring SIGTTOU signals ...

to:

If the calling thread is blocking SIGTTOU signals or the process is ignoring SIGTTOU signals ...

Rationale: Austin Group Defect Report(s) applied: 79.
See http://austingroupbugs.net/view.php?id=79

Change Number: XSH/TC1/D5/0656 [79]

On Page: 2095 Line: 66288 Section: tcsetpgrp()

In the ERRORS section, add:

[EIO] The process group of the writing process is orphaned, the calling thread is not blocking SIGTTOU, and the process is not ignoring SIGTTOU.

Rationale: Austin Group Defect Report(s) applied: 79.
See http://austingroupbugs.net/view.php?id=79

Change Number: XSH/TC1/D5/0657 [291]

On Page: 2102 Line: 66538-66539 Section: tempnam()

In the EXAMPLES section, change both instances of "filename" to "pathname".

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XSH/TC1/D5/0658 [137]

On Page: 2102 Line: 66543 Section: tempnam()

In the EXAMPLES section, change from:

```
char *directory = "/tmp";
char *fileprefix = "file";
```

to:

```
const char *directory = "/tmp";
const char *fileprefix = "file";
```

Rationale: Austin Group Defect Report(s) applied: 137.
See http://austingroupbugs.net/view.php?id=137

Change Number: XSH/TC1/D5/0659 [137]

On Page: 2103 Line: 66559 Section: tempnam()

In the SEE ALSO section, add mkdtemp() and mkstemp().

Rationale: Austin Group Defect Report(s) applied: 137.
See http://austingroupbugs.net/view.php?id=137

Change Number: XSH/TC1/D5/0660 [68]

On Page: 2105 Line: 66603 Section: tgamma()

In the RETURN VALUE section, add a new section after L66603:

If the correct value would cause underflow, [MXX]and is not representable[/MXX], a range error may occur, and tgamma(), tgammaf(), and tgammal() shall return [MXX]0.0, or[/MXX] (if IEC 60559 Floating-Point is not supported) an implementation-defined value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.

[MXX]If the correct value would cause underflow, and is representable, a range error may occur and the correct value shall be returned.[/MXX]

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0661 [320]

On Page: 2105 Line: 66606 Section: tgamma()

In the RETURN VALUE section, change from:

... and either a NaN (if supported), or an implementation-defined value shall be returned.

to:

... and a NaN shall be returned.

Rationale: Austin Group Defect Report(s) applied: 320.
See http://austingroupbugs.net/view.php?id=320

Change Number: XSH/TC1/D5/0662 [68]

On Page: 2106 Line: 66635 Section: tgamma()

In the ERRORS section, add to the end of the "may fails" errors:

Range Error The result underflows.

If the integer expression (math_errhandling & MATH_ERRNO) is non-zero, then errno shall be set to [ERANGE]. If the integer expression (math_errhandling & MATH_ERREXCEPT) is non-zero, then the underflow floating-point exception shall be raised.

Rationale: Austin Group Defect Report(s) applied: 68.
See http://austingroupbugs.net/view.php?id=68

Change Number: XSH/TC1/D5/0663 [106]

On Page: 2107 Line: 66673 Section: time()

In the ERRORS section, change from:

No errors are defined.

to:

The time() function may fail if:

[EOVERFLOW] The number of seconds since the Epoch will not fit in an object of type time_t.

with CX shading.

Rationale: Austin Group Defect Report(s) applied: 106.
See http://austingroupbugs.net/view.php?id=106

Change Number: XSH/TC1/D5/0664 [350]

On Page: 2108 Line: 66712 Section: time()

In the RATIONALE section, change from:

The time() function returns a value in seconds (type time_t) while times() returns a set of values in clock ticks (type clock_t). Some historical implementations, such as 4.3 BSD, have mechanisms capable of returning more precise times (see below). A generalized timing scheme to unify these various timing mechanisms has been proposed but not adopted.

to:

The time() function returns a value in seconds while clock_gettime() and gettimeofday() return a struct timespec (seconds and nanoseconds) and struct timeval (seconds and microseconds), respectively, and are therefore capable of returning more precise times. The times() function is also capable of more precision than time() as it returns a value in clock ticks, although it returns the elapsed time since an arbitrary point such as system boot time, not since the epoch.

Rationale: Austin Group Defect Report(s) applied: 350.
See http://austingroupbugs.net/view.php?id=350

Change Number: XSH/TC1/D5/0665 [106]

On Page: 2108 Line: 66718 Section: time()

In the RATIONALE section, add after the 2nd paragraph a new paragraph:

On some systems the time() function is implemented using a system call that does not return an error condition in addition to the return value. On these systems it is impossible to differentiate between valid and invalid return values and hence overflow conditions cannot be reliably detected.

Rationale: Austin Group Defect Report(s) applied: 106.
See http://austingroupbugs.net/view.php?id=106

Change Number: XSH/TC1/D5/0666 [350]

On Page: 2108 Line: 66724 Section: time()

In the RATIONALE section, delete the sentence:

4.3 BSD includes time() only as an alternate function to the more flexible gettimeofday() function.

Rationale: Austin Group Defect Report(s) applied: 350.
See http://austingroupbugs.net/view.php?id=350

Change Number: XSH/TC1/D5/0667 [350]

On Page: 2108 Line: 66732 Section: time()

In the SEE ALSO section, add clock_gettime(), times(), utimes(), and utimensat() to the SEE ALSO list.

Rationale: Austin Group Defect Report(s) applied: 350.
See http://austingroupbugs.net/view.php?id=350

Change Number: XSH/TC1/D5/0668 [14]

On Page: 2122 Line: 67175 Section: tmpfile()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0669 [291]

On Page: 2123 Line: 67207 Section: tmpnam()

In the DESCRIPTION section, change from:

... valid filename and that is not the same as the name of an existing file.

to:

... valid pathname that does not name an existing file.

Rationale: Austin Group Defect Report(s) applied: 291.
See http://austingroupbugs.net/view.php?id=291

Change Number: XSH/TC1/D5/0670 [291,429]

On Page: 2123 Line: 67227-67235 Section: tmpnam()

In the EXAMPLES and APPLICATION USAGE sections, change all instances of "filename" to "pathname".

Rationale: Austin Group Defect Report(s) applied: 291,429.
See http://austingroupbugs.net/view.php?id=291
See http://austingroupbugs.net/view.php?id=429

Change Number: XSH/TC1/D5/0671 [283]

On Page: 2126 Line: 67304 Section: tolower()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to tolower_l() is the special locale object LC_GLOBAL_LOCALE or is not a valid locale object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0672 [283]

On Page: 2126 Line: 67310 Section: tolower()

In the ERRORS section, change from:

The tolower_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0673 [283]

On Page: 2127 Line: 67347 Section: toupper()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to toupper_l() is the special locale object LC_GLOBAL_LOCALE or is not a valid locale object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0674 [283]

On Page: 2127 Line: 67352 Section: toupper()

In the ERRORS section, change from:

The toupper_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0675 [302]

On Page: 2129 Line: 67386 Section: towctrans()

In the DESCRIPTION section, change from:

... current locale of the process ...

to:

... current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0676 [283]

On Page: 2129 Line: 67395 Section: towctrans()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to towctrans_l()
is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0677 [283]

On Page: 2129 Line: 67402 Section: towctrans()

In the ERRORS section, delete:

The towctrans_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0678 [302]

On Page: 2131 Line: 67442 Section: towlower()

In the DESCRIPTION section, change from:

... the locale of the process ...

to:

... the current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0679 [283]

On Page: 2131 Line: 67445 Section: towlower()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to towlower_l()
is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0680 [283]

On Page: 2131 Line: 67451 Section: towlower()

In the ERRORS section, change from:

The towlower_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.
See http://austingroupbugs.net/view.php?id=283

Change Number: XSH/TC1/D5/0681 [302]

On Page: 2133 Line: 67491 Section: towupper()

change from:

... the locale of the process ...

15073
15074 to:
15075
15076 ... the current locale ...
15077
15078 *Rationale:* Austin Group Defect Report(s) applied: 302.
15079 See http://austingroupbugs.net/view.php?id=302
15080 These changes were overlooked during the revision when
15081 per-thread locales were added.
15082
15083
15084 **Change Number: XSH/TC1/D5/0682** [283]
15085
15086
15087 On Page: 2133 Line: 67494 Section: towupper()
15088
15089 In the DESCRIPTION section, add a new paragraph to the end of the section:
15090
15091 [CX]The behavior is undefined if the locale argument to towupper_l()
15092 is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
15093 object handle.[/CX]
15094
15095 *Rationale:* Austin Group Defect Report(s) applied: 283.
15096 See http://austingroupbugs.net/view.php?id=283
15097
15098
15099 **Change Number: XSH/TC1/D5/0683** [283]
15100
15101
15102 On Page: 2133 Line: 67500 Section: towupper()
15103
15104 In the ERRORS section, change from:
15105
15106 The towupper_l() function may fail if:
15107
15108 [EINVAL] locale is not a valid locale object handle.
15109
15110 to:
15111
15112 No errors are defined.
15113
15114 *Rationale:* Austin Group Defect Report(s) applied: 283.
15115 See http://austingroupbugs.net/view.php?id=283
15116
15117
15118 **Change Number: XSH/TC1/D5/0684** [346]
15119
15120
15121 On Page: 2135 Line: 67539 Section: trunc()
15122
15123 For trunc(), at P2135 L67539 (RETURN VALUE), add:
15124
15125 [MX]The result shall have the same sign as x.[/MX]
15126
15127 At L67547 (APPLICATION USAGE), replace "None." with:
15128
15129 The integral value returned by these functions need not be expressible
15130 as an intmax_t. The return value should be tested before assigning it
15131 to an integer type to avoid the undefined results of an integer overflow.
15132
15133 These functions may raise the inexact floating-point exception if the
15134 result differs in value from the argument.
15135

Rationale: Austin Group Defect Report(s) applied: 346.
See http://austingroupbugs.net/view.php?id=346

Change Number: XSH/TC1/D5/0685 [324]

On Page: 2136 Line: 67591 Section: truncate()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of the path prefix is not a directory, ...

to:

A component of the path prefix names an existing file that is neither a directory nor a symbolic link to a directory, ...

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0686 [75]

On Page: 2140 Line: 67654 Section: ttyname()

In the DESCRIPTION section, change from:

The return value may point to static data whose content is overwritten by each call.

to:

The application shall not modify the string returned. The returned pointer might be invalidated or the string content might be overwritten by a subsequent call to ttyname().

Rationale: Austin Group Defect Report(s) applied: 75.
See http://austingroupbugs.net/view.php?id=75

Change Number: XSH/TC1/D5/0687 [87,93]

On Page: 2151 Line: 67920 Section: ungetc()

In the DESCRIPTION section, change from:

A successful intervening call (with the stream pointed to by stream) to a file-positioning function (fseek(), fsetpos(), or rewind()) shall discard any pushed-back bytes for the stream.

to:

A successful intervening call (with the stream pointed to by stream) to a file-positioning function (fseek(), [CX]fseeko()[/CX] fsetpos(), or rewind()) [CX]or fflush()[/CX] shall discard any pushed-back bytes for the stream.

Rationale: Austin Group Defect Report(s) applied: 87,93.

15199 See http://austingroupbugs.net/view.php?id=87
15200 See http://austingroupbugs.net/view.php?id=93
15201 The fseeko() function is added to the list of file positioning functions.
15202 The fflush() function is also added as a function that causes the discard
15203 of any pushed-back bytes for the stream.
15204
15205
15206 **Change Number: XSH/TC1/D5/0688** [87]
15207
15208
15209 On Page: 2151 Line: 67927 Section: ungetc()
15210
15211 In the DESCRIPTION section, change from:
15212
15213 The value of the file-position indicator for the stream after reading
15214 or discarding all pushed-back bytes shall be the same as it was before
15215 the bytes were pushed back.
15216
15217 to:
15218
15219 The value of the file-position indicator for the stream after all
15220 pushed-back bytes have been read, or discarded by calling fseek(),
15221 [CX]fseeko(),[/CX] fsetpos(), or rewind() [CX](but not fflush())[/CX],
15222 shall be the same as it was before the bytes were pushed back.
15223
15224 *Rationale:* Austin Group Defect Report(s) applied: 87.
15225 See http://austingroupbugs.net/view.php?id=87
15226
15227
15228 **Change Number: XSH/TC1/D5/0689** [14]
15229
15230
15231 On Page: 2151 Line: 67946 Section: ungetc()
15232
15233 In the SEE ALSO section, add a reference to XSH Section 2.5.
15234
15235 *Rationale:* Austin Group Defect Report(s) applied: 14.
15236 See http://austingroupbugs.net/view.php?id=14
15237 This is an editorial improvement
15238
15239
15240 **Change Number: XSH/TC1/D5/0690** [87,93]
15241
15242
15243 On Page: 2152 Line: 67962 Section: ungetwc()
15244
15245 In the DESCRIPTION section, change from:
15246
15247 A successful intervening call (with the stream pointed to by stream)
15248 to a file-positioning function (fseek(), fsetpos(), or rewind()) shall
15249 discard any pushed-back characters for the stream.
15250
15251 to:
15252
15253 A successful intervening call (with the stream pointed to by stream)
15254 to a file-positioning function (fseek(), [CX]fseeko()[/CX] fsetpos(),
15255 or rewind()) [CX]or fflush()[/CX] shall discard any pushed-back characters
15256 for the stream.
15257
15258 *Rationale:* Austin Group Defect Report(s) applied: 87,93.
15259 See http://austingroupbugs.net/view.php?id=87
15260 See http://austingroupbugs.net/view.php?id=93
15261 The fseeko() function is added to the list of file positioning functions.

15262 The fflush() function is also added as a function that causes the discard of
15263 any pushed-back bytes for the stream.
15264
15265
15266 **Change Number: XSH/TC1/D5/0691** [87]
15267
15268
15269 On Page: 2152 Line: 67971 Section: ungetwc()
15270
15271 In the DESCRIPTION section, change from:
15272
15273 The value of the file-position indicator for the stream after reading
15274 or discarding all pushed-back characters shall be the same as it was
15275 before the characters were pushed back.
15276
15277 to:
15278
15279 The value of the file-position indicator for the stream after all
15280 pushed-back characters have been read, or discarded by calling fseek(),
15281 [CX]fseeko(),[/CX] fsetpos(), or rewind() [CX](but not fflush())[/CX],
15282 shall be the same as it was before the characters were pushed back.
15283
15284 *Rationale:* Austin Group Defect Report(s) applied: 87.
15285 See http://austingroupbugs.net/view.php?id=87
15286
15287
15288 **Change Number: XSH/TC1/D5/0692** [14]
15289
15290
15291 On Page: 2153 Line: 67992 Section: ungetwc()
15292
15293 In the SEE ALSO section, add a reference to XSH Section 2.5.
15294
15295 *Rationale:* Austin Group Defect Report(s) applied: 14.
15296 See http://austingroupbugs.net/view.php?id=14
15297 This is an editorial improvement
15298
15299
15300 **Change Number: XSH/TC1/D5/0693** [461]
15301
15302
15303 On Page: 2154 Line: 68031 Section: unlink()
15304
15305 In the DESCRIPTION section, change from:
15306
15307 ... the current working directory is used ...
15308
15309 to:
15310
15311 ... the current working directory shall be used ...
15312
15313 *Rationale:* Austin Group Defect Report(s) applied: 461.
15314 See http://austingroupbugs.net/view.php?id=461
15315
15316
15317 **Change Number: XSH/TC1/D5/0694** [324]
15318
15319
15320 On Page: 2155 Line: 68050 Section: unlink()
15321
15322 In the ERRORS section, for the [ENOTDIR] error, change from:
15323
15324 A component of the path prefix is not a directory, ...

15325
15326 to:
15327
15328 A component of the path prefix names an existing file that is neither
15329 a directory nor a symbolic link to a directory, ...
15330
15331 *Rationale:* Austin Group Defect Report(s) applied: 324.
15332 See http://austingroupbugs.net/view.php?id=324
15333 This is an editorial issue clarifying the intent of the
15334 standard.
15335
15336
15337 **Change Number: XSH/TC1/D5/0695** [278]
15338
15339
15340 On Page: 2155 Line: 68066 unlink()
15341
15342 In the ERRORS section, add (after the [EBADF] error):
15343
15344 [ENOTDIR] The path argument is not an absolute path and fd is a file
15345 descriptor associated with a non-directory file.
15346
15347 *Rationale:* Austin Group Defect Report(s) applied: 278.
15348 See http://austingroupbugs.net/view.php?id=278
15349
15350
15351 **Change Number: XSH/TC1/D5/0696** [278]
15352
15353
15354 On Page: 2156 Line: 68085 Section: unlink()
15355
15356 In the ERRORS section, delete:
15357
15358 [ENOTDIR] The path argument is not an absolute path and fd is neither
15359 AT_FDCWD nor a file descriptor associated with a directory.
15360
15361 *Rationale:* Austin Group Defect Report(s) applied: 278.
15362 See http://austingroupbugs.net/view.php?id=278
15363
15364
15365 **Change Number: XSH/TC1/D5/0697** [96]
15366
15367
15368 On Page: 2160 Line: 68233 Section: unlockpt()
15369
15370 In the RATIONALE section, change from:
15371
15372 None.
15373
15374 to:
15375
15376 See RATIONALE for posix_openpt().
15377
15378 On L68237, in the SEE ALSO section, add posix_openpt().
15379
15380 *Rationale:* Austin Group Defect Report(s) applied: 96.
15381 See http://austingroupbugs.net/view.php?id=96
15382 Add reference to posix_openpt() to the RATIONALE and SEE
15383 ALSO sections.
15384
15385
15386 **Change Number: XSH/TC1/D5/0698** [167]
15387

15388
15389 On Page: 2161 Line: 68256 Section: unsetenv()
15390
15391 In the DESCRIPTION section, remove the text:
15392
15393 If the application modifies environ or the pointers to which it points,
15394 the behavior of unsetenv() is undefined.
15395
15396 *Rationale:* Austin Group Defect Report(s) applied: 167.
15397 See http://austingroupbugs.net/view.php?id=167
15398 The text relating to undefined behavior on modification of
15399 environ is removed as this is now covered by general text added in XBD
15400 Section 8.1.
15401
15402
15403 **Change Number: XSH/TC1/D5/0699** [185]
15404
15405
15406 On Page: 2161 Line: 68264 Section: unsetenv()
15407
15408 In the ERRORS section, change from:
15409
15410 [EINVAL] The name argument is a null pointer, points to an empty string,
15411 or points to a string containing an '=' character.
15412
15413 to:
15414
15415 [EINVAL] The name argument points to an empty string or points to a
15416 string containing an '=' character.
15417
15418 *Rationale:* Austin Group Defect Report(s) applied: 185.
15419 See http://austingroupbugs.net/view.php?id=185
15420
15421
15422 **Change Number: XSH/TC1/D5/0700** [290]
15423
15424
15425 On Page: 2162 Line: 68296 Section: uselocale()
15426
15427 In the DESCRIPTION section change "a null pointer" to "(locale_t)0".
15428
15429 *Rationale:* Austin Group Defect Report(s) applied: 290.
15430 See http://austingroupbugs.net/view.php?id=290
15431
15432
15433 **Change Number: XSH/TC1/D5/0701** [334]
15434
15435
15436 On Page: 2162 Line: 68301 Section: uselocale()
15437
15438 In the RETURN VALUE section, change from:
15439
15440 The uselocale() function returns the locale handle from the previous
15441 call for the current thread. If there was no such previous call, the
15442 function shall return the value LC_GLOBAL_LOCALE.
15443
15444 to:
15445
15446 Upon successful completion, the uselocale() function shall return
15447 the locale handle from the previous call for the current thread,
15448 or LC_GLOBAL_LOCALE if there was no such previous call. Otherwise,
15449 uselocale() shall return (locale_t)0 and set errno to indicate the error.
15450

Rationale: Austin Group Defect Report(s) applied: 334.
See http://austingroupbugs.net/view.php?id=334

Change Number: XSH/TC1/D5/0702 [324]

On Page: 2164 Line: 68355 Section: utime()

In the ERRORS section, for the [ENOTDIR] error, change from:

A component of the path prefix is not a directory, ...

to:

A component of the path prefix names an existing file that is neither a directory nor a symbolic link to a directory, ...

Rationale: Austin Group Defect Report(s) applied: 324.
See http://austingroupbugs.net/view.php?id=324
This is an editorial issue clarifying the intent of the standard.

Change Number: XSH/TC1/D5/0703 [14]

On Page: 2168 Line: 68459 Section: vfprintf()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0704 [14]

On Page: 2170 Line: 68503 Section: vfscanf()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0705 [14]

On Page: 2171 Line: 68540 Section: vfwprintf()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0706 [14]

On Page: 2172 Line: 68581 Section: vfwscanf()

In the SEE ALSO section, add a reference to XSH Section 2.5.

Rationale: Austin Group Defect Report(s) applied: 14.
See http://austingroupbugs.net/view.php?id=14
This is an editorial improvement

Change Number: XSH/TC1/D5/0707 [421]

On Page: 2183 Line: 68781 Section: wait()

In the RETURN VALUE section, change from:

Otherwise, (pid_t)-1 shall be returned, ...

to:

Otherwise, -1 shall be returned, ...

Rationale: Austin Group Defect Report(s) applied: 421.
See http://austingroupbugs.net/view.php?id=421

Change Number: XSH/TC1/D5/0708 [166]

On Page: 2185 Line: 68859 Section: wait()

In the EXAMPLES section, change from:

```
int status;
```

to:

```
int sav_errno = errno;
int status;
```

Rationale: Austin Group Defect Report(s) applied: 166.
See http://austingroupbugs.net/view.php?id=166

Change Number: XSH/TC1/D5/0709 [166]

On Page: 2186 Line: 68887 Section: wait()

In the EXAMPLES section, change from:

```
    }
}
```

to:

```
    }
    errno = sav_errno;
}
```

Rationale: Austin Group Defect Report(s) applied: 166.
See http://austingroupbugs.net/view.php?id=166

Change Number: XSH/TC1/D5/0710 [69]

On Page: 2186 Line: 68919 Section: wait()

In the EXAMPLES section, insert before L68919 (before the final line of the example code):

```
return 0; /* NOTREACHED */
```

Rationale: Austin Group Defect Report(s) applied: 69.
See http://austingroupbugs.net/view.php?id=69

Change Number: XSH/TC1/D5/0711 [154]

On Page: 2190 Line: 69071 Section: waitid()

In the DESCRIPTION section, change from:

WCONTINUED Status shall be returned for any child that was stopped and has been continued.

to:

WCONTINUED Status shall be returned for any continued child process whose status either has not been reported since it continued from a job control stop or has been reported only by calls to waitid() with the WNOWAIT flag set.

Rationale: Austin Group Defect Report(s) applied: 154.
See http://austingroupbugs.net/view.php?id=154

Change Number: XSH/TC1/D5/0712 [154]

On Page: 2190 Line: 69078 Section: waitid()

In the DESCRIPTION section, change from:

WSTOPPED Status shall be returned for any child that has stopped upon receipt of a signal.

to:

WSTOPPED Status shall be returned for any child that has stopped upon receipt of a signal, and whose status either has not been reported since it stopped or has been reported only by calls to waitid() with the WNOWAIT flag set.

Rationale: Austin Group Defect Report(s) applied: 154.
See http://austingroupbugs.net/view.php?id=154

Change Number: XSH/TC1/D5/0713 [153]

On Page: 2190 Line: 69084 Section: waitid()

In the DESCRIPTION section (final paragraph), change from:

15640
15641 ... with the status of the process. The si_signo member shall always be
15642 equal to SIGCHLD.
15643
15644 to:
15645
15646 ... with the status of the process; the si_signo member shall be set
15647 equal to SIGCHLD. If waitid() returns because WNOHANG was specified and
15648 status is not available for any process specified by idtype and id, then
15649 the si_signo and si_pid members of the structure pointed to by infop
15650 shall be set to zero and the values of other members of the structure
15651 are unspecified.
15652
15653 *Rationale:* Austin Group Defect Report(s) applied: 153.
15654 See http://austingroupbugs.net/view.php?id=153
15655
15656
15657 **Change Number: XSH/TC1/D5/0714** [88]
15658
15659
15660 On Page: 2195 Line: 69147 Section: wcrtomb()
15661
15662 In the SYNOPSIS section, change from:
15663
15664 #include <stdio.h>
15665
15666 to:
15667
15668 #include <wchar.h>
15669
15670 *Rationale:* Austin Group Defect Report(s) applied: 88.
15671 See http://austingroupbugs.net/view.php?id=88
15672
15673
15674 **Change Number: XSH/TC1/D5/0715** [105]
15675
15676
15677 On Page: 2195 Line: 69168 Section: wcrtomb()
15678
15679 In the DESCRIPTION section, add a new paragraph at the end of the
15680 DESCRIPTION:
15681
15682 The wcrtomb() function shall not change the setting of errno if
15683 successful.
15684
15685 *Rationale:* Austin Group Defect Report(s) applied: 105.
15686 See http://austingroupbugs.net/view.php?id=105
15687
15688
15689 **Change Number: XSH/TC1/D5/0716** [294]
15690
15691
15692 On Page: 2197 Line: 69221 Section: wcscasecmp()
15693
15694 In the DESCRIPTION section, change from:
15695
15696 When the LC_CTIME category of the current locale is from the POSIX locale,
15697 these functions shall behave as if the strings had been converted to
15698 lowercase and then a byte comparison performed. Otherwise, the results
15699 are unspecified.
15700
15701 The information for wcscasecmp_l() and wcsncasecmp_l() about the case
15702 of the characters comes from the locale represented by locale.

15703
15704 to:
15705
15706 The wcscasecmp() and wcsncasecmp() functions use the current locale to
15707 determine the case of the wide characters.
15708
15709 The wcscasecmp_l() and wcsncasecmp_l() functions use the locale
15710 represented by locale to determine the case of the wide characters.
15711
15712 When the LC_CTYPE category of the locale being used is from the POSIX
15713 locale, these functions shall behave as if the wide-character strings
15714 had been converted to lowercase and then a comparison of wide-character
15715 codes performed. Otherwise, the results are unspecified.
15716
15717 *Rationale:* Austin Group Defect Report(s) applied: 294.
15718 See http://austingroupbugs.net/view.php?id=294
15719
15720
15721 **Change Number: XSH/TC1/D5/0717** [283]
15722
15723
15724 On Page: 2197 Line: 69225 Section: wcscasecmp()
15725
15726 In the DESCRIPTION section, add a new paragraph to the end of the section:
15727
15728 The behavior is undefined if the locale argument to wcscasecmp_l() or
15729 wcsncasecmp_l() is the special locale object LC_GLOBAL_LOCALE or is not
15730 a valid locale object handle.
15731
15732 *Rationale:* Austin Group Defect Report(s) applied: 283.
15733 See http://austingroupbugs.net/view.php?id=283
15734
15735
15736 **Change Number: XSH/TC1/D5/0718** [283]
15737
15738
15739 On Page: 2197 Line: 69236 Section: wcscasecmp()
15740
15741 In the ERRORS section, change from:
15742
15743 The wcscasecmp_l() and wcsncasecmp_l() functions may fail if:
15744
15745 [EINVAL] locale is not a valid locale object handle.
15746
15747 to:
15748
15749 No errors are defined.
15750
15751 *Rationale:* Austin Group Defect Report(s) applied: 283.
15752 See http://austingroupbugs.net/view.php?id=283
15753
15754
15755 **Change Number: XSH/TC1/D5/0719** [302]
15756
15757
15758 On Page: 2202 Line: 69364 Section: wcscoll()
15759
15760 In the DESCRIPTION section, change from:
15761
15762 ... current locale of the process ...
15763
15764 to:
15765

15766 ... current locale ...
15767
15768 *Rationale:* Austin Group Defect Report(s) applied: 302.
15769 See http://austingroupbugs.net/view.php?id=302
15770 These changes were overlooked during the revision when
15771 per-thread locales were added.
15772
15773
15774 **Change Number: XSH/TC1/D5/0720** [283]
15775
15776
15777 On Page: 2202 Line: 69368 Section: wcscoll()
15778
15779 In the DESCRIPTION section, add a new paragraph to the end of the section:
15780
15781 [CX]The behavior is undefined if the locale argument to wcscoll_l()
15782 is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
15783 object handle.[/CX]
15784
15785 *Rationale:* Austin Group Defect Report(s) applied: 283.
15786 See http://austingroupbugs.net/view.php?id=283
15787
15788
15789 **Change Number: XSH/TC1/D5/0721** [283]
15790
15791
15792 On Page: 2202 Line: 69380 Section: wcscoll()
15793
15794 In the ERRORS section, delete:
15795
15796 The wcscoll_l() function may fail if:
15797
15798 [EINVAL] locale is not a valid locale object handle.
15799
15800 *Rationale:* Austin Group Defect Report(s) applied: 283.
15801 See http://austingroupbugs.net/view.php?id=283
15802
15803
15804 **Change Number: XSH/TC1/D5/0722** [109,105]
15805
15806
15807 On Page: 2219 Line: 69818 Section: wcsnrtombs()
15808
15809 In the DESCRIPTION section, change from:
15810
15811 [CX]The wcsrtombs() function need not be thread-safe ... [/CX]
15812
15813 to:
15814
15815 [CX]The wcsnrtombs() and wcsrtombs() functions need not be thread-safe
15816 ... [/CX]
15817
15818 The wcsrtombs() function shall not change the setting of errno if
15819 successful.
15820
15821 *Rationale:* Austin Group Defect Report(s) applied: 109,105.
15822 See http://austingroupbugs.net/view.php?id=109
15823 See http://austingroupbugs.net/view.php?id=105
15824
15825
15826 **Change Number: XSH/TC1/D5/0723** [302]
15827
15828

On Page: 2224 Line: 69988 Section: wcstod()

In the DESCRIPTION section, change from:

... the locale of the process ...

to:

... the current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.
See http://austingroupbugs.net/view.php?id=302
These changes were overlooked during the revision when
per-thread locales were added.

Change Number: XSH/TC1/D5/0724 [105]

On Page: 2224 Line: 69996 Section: wcstod()

In the DESCRIPTION section, remove the CX shading from the paragraphs:

The wcstod() function shall not change the setting of errno if successful.

Since 0 is returned on error and is also a valid return on success, an
application wishing to check for error situations should set errno to 0,
then call wcstod(), wcstof(), or wcstold(), then check errno.

and then change the first paragraph:

The wcstod() function shall not change the setting of errno if successful.

to:

These functions shall not change the setting of errno if successful.

Rationale: Austin Group Defect Report(s) applied: 105.
See http://austingroupbugs.net/view.php?id=105

Change Number: XSH/TC1/D5/0725 [105]

On Page: 2231 Line: 70204 Section: wcstol()

In the DESCRIPTION section, remove the CX shading from the paragraphs:

These functions shall not change the setting of errno if successful.

Since 0, {LONG_MIN} or {LLONG_MIN} and {LONG_MAX} or {LLONG_MAX} are
returned on error and are also valid returns on success, an application
wishing to check for error situations should set errno to 0, then call
wcstol() or wcstoll(), then check errno.

Rationale: Austin Group Defect Report(s) applied: 105.
See http://austingroupbugs.net/view.php?id=105

Change Number: XSH/TC1/D5/0726 [109]

On Page: 2235 Line: 70283 Section: wcstombs()

15892
15893 In the DESCRIPTION secton, delete:
15894
15895 The wcstombs() function need not be thread-safe.
15896
15897 *Rationale:* Austin Group Defect Report(s) applied: 109.
15898 See http://austingroupbugs.net/view.php?id=109
15899
15900
15901 **Change Number: XSH/TC1/D5/0727** [105]
15902
15903
15904 On Page: 2238 Line: 70365 Section: wcstoul()
15905
15906 In the DESCRIPTION section, remove the CX shading from the paragraphs:
15907
15908 The wcstoul() function shall not change the setting of errno if successful.
15909
15910 Since 0, {ULONG_MAX}, and {ULLONG_MAX} are returned on error and 0 is
15911 also a valid return on success, an application wishing to check for error
15912 situations should set errno to 0, then call wcstoul() or wcstoull(),
15913 then check errno.
15914
15915 and then change the first paragraph from:
15916
15917 The wcstoul() function shall not change the setting of errno if successful.
15918
15919 to:
15920
15921 These functions shall not change the setting of errno if successful.
15922
15923 *Rationale:* Austin Group Defect Report(s) applied: 105.
15924 See http://austingroupbugs.net/view.php?id=105
15925
15926
15927 **Change Number: XSH/TC1/D5/0728** [302]
15928
15929
15930 On Page: 2242 Line: 70463 Section: wcsxfrm()
15931
15932 Change from:
15933
15934 ... the locale of the process ...
15935
15936 to:
15937
15938 ... the current locale ...
15939
15940 *Rationale:* Austin Group Defect Report(s) applied: 302.
15941 See http://austingroupbugs.net/view.php?id=302
15942 These changes were overlooked during the revision when
15943 per-thread locales were added.
15944
15945
15946 **Change Number: XSH/TC1/D5/0729** [283]
15947
15948
15949 On Page: 2242 Line: 70469 Section: wcsxfrm()
15950
15951 In the DESCRIPTION section, add a new paragraph to the end of the section:
15952
15953 [CX]The behavior is undefined if the locale argument to wcsxfrm_l()
15954 is the special locale object LC_GLOBAL_LOCALE or is not a valid locale