TECHNICAL REPORT

ISO/IEC TR 24715

First edition
2006-11-01

Information technology — Programming languages, their environments and system software interfaces — Technical Report on the Conflicts between the ISO/IEC 9945 (POSIX) and the Linux Standard Base (ISO/IEC 23360)

Technologies de l'information — Langages de programmation, leurs environnements et interfaces de logiciel système — Rapport technique sur les conflits entre l'ISO/CEI 9945 (POSIX) et la base normalisée Linux (ISO/CEI 23360)

Reference number
ISO/IEC TR 24715:2006(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

In exceptional circumstances, the joint technical committee may propose the publication of a Technical Report of one of the following types:

— type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;

— type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility of an agreement on an International Standard;

— type 3, when the joint technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example).

Technical Reports of types 1 and 2 are subject to review within three years of publication, to decide whether they can be transformed into International Standards. Technical Reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC TR 24715, which is a Technical Report of type 3, was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces*.

Introduction

The purpose of this Type 3 Technical Report (informative) is to document the areas of conflict between ISO/IEC 9945 (POSIX[1]) and the Free Standards Group's Linux[2] Standard Base specification (ISO/IEC 23360) such that it can be utilized by the appropriate technical committees when considering harmonization between the standards efforts.

ISO/IEC 9945 (POSIX) is an important International Standard in use throughout the world. There is a significant investment in applications developed for the ISO POSIX International Standard. With the emergence of a standardization initiative for the Linux operating system, there are some areas of conflict that have been identified between the Linux Standard Base specification (ISO/IEC 23360) and the ISO POSIX International Standards. There is an essential market requirement that the conflicts be resolved so that an application can be written to conform to both International Standards. Hundreds of millions of dollars of applications are built upon these International Standards. This Technical Report is intended as a starting point to look at resolution of this issue.

[1] POSIX® is a registered trademark of the IEEE.

[2] Linux® is a registered trademark of Linus Torvalds.

Participants

The following people contributed work to this Technical Report:

Stuart Anderson

Andrew Josey

The Austin Group

Paul Eggert

Nick Stoughton

The LSB Workgroup

Information technology — Programming languages, their environments and system software interfaces — Technical Report on the Conflicts between the ISO/IEC 9945 (POSIX) and the Linux Standard Base (ISO/IEC 23360)

1 Scope

The scope of this Technical Report is to identify areas of conflict between the Linux Standard Base (LSB[3]) 3.1 specification (ISO/IEC 23360) and the ISO/IEC 9945 (POSIX) standard.

It is based on the Linux Standard Base Core Specification 3.1, which was submitted to ISO/IEC on 2005-10-31 for publication as ISO/IEC 23360; and ISO/IEC 9945:2003 edition dated 2003-08-15 with ISO/IEC 9945:2003/Cor.1:2004 (published 2004-09-15).

The audience for this Technical Report is the technical workgroups that develop the standards; that is, the Austin Group and the Linux Standard Base workgroup. It is also intended to be of interest to systems engineers, technical managers and procurement officers.

This document is organized in the following clauses:

- Clause 2 provides a list of normative references.
- Clause 3 provides the terms and definitions used in this document.
- Clause 4 provides a list of differences that could be possible conflicts or extensions in the System Interfaces.
- Clause 5 provides a list of differences that could be possible conflicts or extensions in the Shell and Utilities.
- Appendix A provides background information on the POSIX standards and the LSB.

[3] LSB™ is a trademark of the Free Standards Group.

2 Normative References

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 9945-1:2003, *Information technology — Portable Operating System Interface (POSIX) — Part 1: Base Definitions*

ISO/IEC 9945-1:2003/Cor 1:2004

ISO/IEC 9945-2:2003, *Information technology — Portable Operating System Interface (POSIX) — Part 2: System Interfaces*

ISO/IEC 9945-2:2003/Cor 1:2004

ISO/IEC 9945-3:2003, *Information technology — Portable Operating System Interface (POSIX) — Part 3: Shell and Utilities*

ISO/IEC 9945-3:2003/Cor 1:2004

ISO/IEC 9945-4:2003, *Information technology — Portable Operating System Interface (POSIX) — Part 4: Rationale*

ISO/IEC 9945-4:2003/Cor 1:2004

ISO/IEC 23360 (all parts), *Linux Standard Base (LSB) Core Specification 3.1*

3 Terms and Definitions

For the purposes of this document, the terms and definitions of ISO/IEC 9945-1:2003, *Information technology — Portable Operating System Interface (POSIX) — Part 1: Base Definitions apply*.

4 System Interfaces

This section describes possible areas of conflict between the LSB and ISO/IEC 9945 (POSIX) for the System Interfaces.

This description is based on the Linux Standard Base Core Specification 3.1. Note that the descriptions of the known conflicts are taken from the LSB and have not been verified by the Austin Group, thus they may be subject to interpretation of the standard. In some cases, the differences may be upward-compatible extensions. In cases where the LSB provides its own API reference page rather than referencing ISO/IEC 9945, then that is noted here, and it is possible that further investigation might determine that there is no conflict.

4.1 Headers and Interface Definitions

4.1.1 errno

LSB requires [ENOTSUP] to be defined the same as [EOPNOTSUP], whereas ISO/IEC 9945 requires that these *errno* values be unique. A defect report has been filed to the Austin Group on this matter.

4.1.2 fcntl

LSB permits implementations to set O_LARGEFILE.

According to ISO/IEC 9945, only an application sets *fcntl*() flags; for example, O_LARGEFILE. However, the LSB specification also allows an implementation to set the O_LARGEFILE flag in the case where the programming environment is one of:

_POSIX_V6_ILP32_OFFBIG
_POSIX_V6_LP64_OFF64
_POSIX_V6_LPBIG_OFFBIG

See *getconf* and *c99* in ISO/IEC 9945 for a description of these environments. Thus, calling *fcntl*() with the F_GETFL command may return O_LARGEFILE as well as flags explicitly set by the application in the case that both the implementation and the application support an **off_t** of at least 64 bits.

4.1.3 fscanf, fwscanf, scanf, vfscanf, vfwscanf, vscanf, vsscanf, vswscanf,vwscanf, wscanf

The LSB states:

The `%s`, `%S` and `%[` conversion specifiers shall accept an option length modifier `a`, which shall cause a memory buffer to be allocated to hold the string converted. In such a case, the argument corresponding to the conversion specifier should be a reference to a pointer value that will receive a pointer to the allocated buffer. If there is insufficient memory to allocate a buffer, the function may set *errno* to [ENOMEM] and a conversion error results.

According to the LSB this directly conflicts with the ISO C (1999) usage of `%a` as a conversion specifier for hexadecimal float values. While this conversion specifier should be supported, a format specifier such as "`%aseconds`" will have a different meaning on an LSB conforming system.

4.1.4 getopt

The LSB documents a number of GNU extensions to *getopt*() as well as descriptions of the POSIX requirements. Such extensions include argument ordering. The LSB requires LSB implementations to implement the GNU behavior. The POSIXLY_CORRECT environment variable is documented as a method to obtain ISO/IEC 9945 conforming behavior.

4.1.5 kill

Process ID -1 doesn't affect the calling process:

If *pid* is specified as -1, LSB says that *sig* shall not be sent to the calling process, whereas ISO/IEC 9945 states: “If *pid* is -1, *sig* shall be sent to all processes (excluding an unspecified set of system processes) for which the process has permission to send that signal.”

This was a deliberate Linus decision after an unpopular experiment in including the calling process in the 2.5.1 kernel. See “What does it mean to signal everybody?”.[4]

4.1.6 link

link need not follow symbolic links:

ISO/IEC 9945 specifies that pathname resolution shall follow symbolic links during pathname resolution unless the function is required to act on the symbolic link itself, or certain arguments direct that the function act on the symbolic link itself. The ISO/IEC 9945 *link*() function contains no such requirement to operate on a symbolic link. However, a conforming LSB implementation need not follow a symbolic link for the *path1* argument, and hence may allow implementations of *link*() to create a link to a symbolic link itself.

[4] “What does it mean to signal everybody?”, Linux Weekly News, 20 December 2001, lwn.net/2001/1220/kernel.php3.

4.1.7 regexec

The LSB permits certain aspects of regular expression matching to be optional; see Internationalization and Regular Expressions.

4.1.8 strerror_r

The SYNOPSIS for *strerror_r* is defined in the LSB to have a return value of type **extern char ***, whereas ISO / IEC 9945 defines the return value to be type **int**.

Also, according to the LSB it is optional whether an implementation copies a message into the supplied buffer.

Returns string, not error value:

The *strerror_r*() function shall return a pointer to the string corresponding to *errno*. The returned pointer may point within the buffer *buf* (at most *buflen* bytes).

Return value:

On success, *strerror_r*() shall return a pointer to the generated message string (determined by the setting of the LC_MESSAGES category in the current locale). Otherwise, *strerror_r*() shall return the string corresponding to “Unknown error”.

4.1.9 strptime

The LSB documents an issue with limiting the number of leading zeroes.

LSB states:

“Number of leading zeroes limited.”

The Single UNIX Specification, Version 2 specifies fields for which “leading zeros are permitted but not required”; however, applications must not expect to be able to supply more leading zeroes for these fields than would be implied by the range of the field. Implementations may choose to either match an input with excess leading zeroes, or treat this as a non-matching input. For example, `%j` has a range of 001 to 366, so 0, 00, 000, 001, and 045 are acceptable inputs, but inputs such as 0000, 0366, and the like are not.

Rationale:

glibc developers consider it appropriate behavior to forbid excess leading zeroes. When trying to parse a given input against several format strings, forbidding excess leading zeroes could be helpful. For example, if one matches 0011-12-26 against `%m-%d-%Y` and then against `%Y-%m-%d`, it seems useful for the first match to fail, as it would be perverse to parse that date as November 12, year 26. The second pattern parses it as December 26, year 11.

The Single UNIX Specification is not explicit that an unlimited number of leading zeroes is required, although it may imply this. The LSB explicitly allows implementations to have either behavior. Future versions of this standard may require implementations to forbid excess leading zeroes.

An Interpretation Request is currently pending against ISO/IEC 9945 for this matter.

4.1.10 unlink

May return [EISDIR] on directories:

The LSB states that if *path* specifies a directory, a return of [EISDIR] is permitted instead of [EPERM] as required by ISO/IEC 9945.

LSB notes that: “The Linux kernel has deliberately chosen [EISDIR] for this case and does not expect to change (Al Viro, personal communication).”

4.1.11 waitpid

The LSB does not require implementations to support the WCONTINUED or WIFCONTINUED functionality within *waitpid*(). It is worth noting that these are XSI extensions (that is, only mandatory for UNIX systems, and so base POSIX conformance is not impacted).

4.2 ISO/IEC 9945 System Interfaces not in the LSB

It is believed that the LSB includes all the mandatory “base” interfaces from ISO/IEC 9945. There are a number of optional system interfaces from ISO/IEC 9945 that are not in the LSB. Whilst not a direct conflict, a consequence of this is that some applications written to ISO/IEC 9945 will not port to a strict LSB implementation. It is recommended that development of a porting guide be considered.

A number of known areas are noted below:

4.2.1 waitid

The LSB does not include the *waitid*() function, whereas it is a first-class function in ISO/IEC 9945 (but in the XSI option group). The consequence of this is that an application written to conform to ISO/IEC 9945 using *waitid*() does not conform to the LSB.

4.2.2 Realtime Threads and Advanced Realtime Thread Options

The LSB does not support functionality associated with the Realtime options defined by ISO/IEC 9945 as the Realtime Threads, and Advanced Realtime Threads Option

Groups (see ISO/IEC 9945, Base Definitions, Section 2.1.5.2). Consequently, any applications using those features will not be LSB conforming. This is documented within the LSB in the *libpthread* section.

4.2.3 XSI STREAMS Option

The LSB does not support functionality associated with the XSI STREAMS option defined by ISO/IEC 9945.

5 Shell and Utilities Interfaces

This section describes possible areas of conflict between the LSB and ISO/IEC 9945 (POSIX) for the Shell and Utilities.

This description is based on the Linux Standard Base Core Specification 3.1. Note that the descriptions of the known conflicts are taken from the LSB and have not been verified by the Austin Group, thus they may be subject to interpretation of the standard. Deprecated differences are not listed since they are assumed to be removed at some future point. In some cases, the differences may be upward-compatible extensions. In cases where the LSB provides its own API reference page rather than referencing ISO/IEC 9945, then that is noted here, and it is possible that further investigation might determine that there is no conflict.

5.1 Built-in cd, getopts, read, umask and wait

ISO/IEC 9945 requires that a number of commands be provided as built-in to the command language interpreter (as detailed in XCU Section 1.13) and also accesible via the *exec* family of functions as defined in XSH and directly invocable by standard utilities that require to invoke them (*env*, *find*, *nice*, *nohup*, *time* and *xargs*). The LSB requires that the *cd*, *getopts*, *read*, *umask* and *wait* utilities be built-in to the command language interpreter and forbids standard utilities from invoking them.

5.2 Utility Definitions

5.2.1 ar

The ar utility is deprecated in the LSB.

5.2.2 at

The LSB lists the following differences:

- **-d** is functionally equivalent to the **-r** option specified in ISO/IEC 9945.
- **-r** need not be supported on LSB implementations, but the **-d** option is equivalent.
- **-t** *time* need not be supported.

The files **at.allow** and **at.deny** may reside in **/etc** rather than **/usr/lib/cron** on LSB implementations.

5.2.3 awk

The LSB lists the following difference:

Certain aspects of internationalized regular expressions are optional.

5.2.4 batch

The LSB lists the following difference:

The files **at.allow** and **at.deny** may reside in **/etc** rather than **/usr/lib/cron** on LSB implementations.

5.2.5 bc

In order to obtain ISO/IEC 9945 conforming behavior, applications are required to use the **-s** or **--standard** option to *bc*.

The **-w** or **--warn** option can provide warnings for *bc* extensions being used over the ISO/IEC 9945 definition.

5.2.6 chgrp

The **-L**, **-H**, and **-P** options need not be supported by LSB implementations. ISO/IEC 9945 defines these options for manipulating symbolic links.

5.2.7 chown

The **-L**, **-H**, and **-P** options need not be supported by LSB implementations. ISO/IEC 9945 defines these options for manipulating symbolic links.

5.2.8 crontab

The LSB lists the following difference:

The files **cron.allow** and **cron.deny** may reside in **/etc** rather than **/usr/lib/cron** on LSB implementations.

5.2.9 cut

The LSB lists the following difference:

-n has unspecified behavior.

ISO/IEC 9945 defines the behavior of **–n**:

“Do not split characters. When specified with the **-b** option, each element in list of the form low- high (hyphen-separated numbers) shall be modified as follows:

If the byte selected by low is not the first byte of a character, low shall be decremented to select the first byte of the character originally selected by low. If the byte selected by high is not the last byte of a character, high shall be decremented to select the last byte of the character prior to the character originally selected by high, or zero if there is no prior character. If the resulting range element has high equal to zero or low greater than high, the list element shall be dropped from list for that input line without causing an error.

Each element in *list* of the form low- shall be treated as above with high set to the number of bytes in the current line, not including the terminating <newline>. Each element in list of the form – high shall be treated as above with low set to 1. Each element in *list* of the form *num* (a single number) shall be treated as above with low set to *num* and high set to *num*."

5.2.10 df

The LSB lists the following differences:

If the **-k** option is not specified, disk space is shown in unspecified units. If the **-P** option is specified, the size of the unit shall be printed on the header line in the format "`%4s-blocks`". Applications should specify **-k**.

The XSI option **-t** has unspecified behavior. Applications should not specify **-t**.

The LSB states that **-t** is used for a different purpose on a common implementation.

Operand may identify special file:

If an argument is the absolute filename of a special file containing a mounted filesystem, *df* shall show the space available on that filesystem rather than on the filesystem containing the special file (which is typically the root filesystem).

The LSB states that in ISO/IEC 9945 the XSI optional behavior permits an operand to name a special file, but appears to require the operation be performed on the filesystem containing the special file. A defect report has been submitted for this case.

5.2.11 du

The LSB lists the following difference:

If the **-k** option is not specified, disk space is shown in unspecified units. Applications should specify **-k**.

5.2.12 echo

Unlike the behavior specified in ISO/IEC 9945, LSB states that support for options is implementation-defined, and that the behavior of *echo* if any arguments contain backslashes is also implementation-defined. Applications are advised not to run *echo* with a first argument starting with a hyphen, or with any arguments containing backslashes; they must use *printf* in those cases.

The behavior specified here is similar to that specified by the Single UNIX Specification Version 3 without the XSI option. However, the LSB forbids all options and ISO/IEC 9945 forbids only **-n**.

5.2.13 *file*

The LSB lists the following difference:

The **-M**, **-h**, **-d**, and **-i** options need not be supported.

5.2.14 *find*

The LSB lists the following differences:

Some elements of the Pattern Matching Notation are optional; see Internationalization and Pattern Matching Notation.

-H need not be supported.

-L need not be supported.

-exec ... + argument aggregation need not be supported.

5.2.15 *fuser*

The LSB lists the following differences:

-c has unspecified behavior.

-f has unspecified behavior.

5.2.16 *grep*

The LSB lists the following difference:

Some elements of the Pattern Matching Notation are optional; see Internationalization and Pattern Matching Notation.

5.2.17 *ipcrm*

The LSB states that if any of the **-q**, **-Q**, **-s**, **-S**, **-m**, or **-M** arguments are given, *ipcrm* shall behave as described in ISO/IEC 9945. Otherwise, *ipcrm* shall remove the resource of the specified type identified by *id*.

5.2.18 ipcs

The LSB has its own definition of *ipcs* and does not reference ISO/IEC 9945. This definition is stated to contain substantial differences from ISO/IEC 9945.

5.2.19 ls

For *ls*, the LSB only lists compatible extensions, no differences. Where ISO/IEC 9945 states that there is implementation-defined behavior for **-l** on a special file, the LSB defines that behavior; and the **-p** extension is upwardly-compatible.

5.2.20 more

The LSB lists the following differences:

The *more* command need not respect the *LINES* and *COLUMNS* environment variables.

The *more* command need not support the following interactive commands:

G
G
u
control u
control f
newline
j
k
r
R
m
' (return to mark)
/!
?
N
:e
:t
control g
ZZ

-num specifies an integer which is the screen size (in lines).

-e has unspecified behavior.

-i has unspecified behavior.

-n has unspecified behavior.

-p – either (1) clear the whole screen and then display the text (instead of the usual scrolling behavior), or (2) provide the behavior specified by ISO/IEC 9945. In the latter case, the syntax is "**-p** *command*".

-t has unspecified behavior.

5.2.21 newgrp

The LSB states that implementations need not support the **–l** option as required in ISO/IEC 9945.

5.2.22 od

Pre-POSIX and XSI Specifications:

The LSB supports mixing options between the mandatory and XSI optional synopsis forms in ISO POSIX (2003). The LSB shall support the following options:

-a is equivalent to **-t** *a*, selects named characters.

-b is equivalent to **-t** *o1*, selects octal bytes.

-c is equivalent to **-t** *c*, selects characters.

-d is equivalent to **-t** *u2*, selects unsigned decimal two byte units.

-f is equivalent to **-t** *fF*, selects floats.

-i is equivalent to **-t** *d2*, selects decimal two byte units.

-l is equivalent to **-t** *d4*, selects decimal longs.

-o is equivalent to **-t** *o2*, selects octal two byte units.

-x is equivalent to **-t** *x2*, selects hexadecimal two byte units.

Note that the XSI option **-s** need not be supported.

5.2.23 renice

The LSB lists the following difference:

-n *increment* has unspecified behavior.

5.2.24 sed

The LSB lists the following difference:

Certain aspects of internationalized regular expressions are optional.

5.2.25 xargs

The LSB lists the following differences:

-E has unspecified behavior.

-I has unspecified behavior.

-L has unspecified behavior.

5.3 Internationalization

The LSB makes certain internationalization aspects optional.

5.3.1 Regular Expressions

Utilities that process regular expressions shall support Basic Regular Expressions and Extended Regular Expressions as specified in ISO/IEC 9945 with the following exceptions:

Range expression (such as `[a-z]`) can be based on code point order instead of collating element order.

Equivalence class expression (such as `[=a=]`) and multi-character collating element expression (such as `[.ch.]`) are optional.

Handling of a multi-character collating element is optional.

This affects at least the following utilities: *grep*, *sed*, and *awk*.

5.3.2 Pattern Matching Notation

Utilities that perform Pattern Matching Notation shall do so as specified in ISO/IEC 9945 with the following exceptions:

Range expression (such as `[a-z]`) can be based on code point order instead of collating element order.

Equivalence class expression (such as `[=a=]`) and multi-character collating element expression (such as `[.ch.]`) are optional.

Handling of a multi-character collating element is optional.

5.4 ISO/IEC 9945 Utility Interfaces not in the LSB

The LSB includes all the mandatory “base” utilities from ISO/IEC 9945. There are a number of optional utility interfaces in ISO/IEC 9945 that are not in the LSB. Whilst not a direct conflict, a consequence of this is that some applications written to ISO/IEC

9945 will not port to a strict LSB implementation. It is recommended that development of a porting guide be considered.

A number of known areas are noted below:

Partial coverage of the User Portability Utilities option:

The LSB does not require a full implementation of the User Portability Utilities option; for example, it does require support for *at*, *batch*, *crontab*, *csplit*, *df*, *du*, *expand*, *file*, *man*, *more*, *newgrp*, *nice*, *patch*, *ps*, *renice*, *split*, *time*, and *unexpand*. However, it omits support for *alias*, *bg*, *ctags*, *ex*, *fc*, *fg*, *jobs*, *mesg*, *nm*, *strings*, *tabs*, *talk*, *tput*, *unalias*, *uudecode*, *uuencode*, *vi*, *who*, and *write*. This is primarily since the LSB is targeted as a run-time environment for applications rather than a user environment.

Development options:

The LSB does not require implementations to support the various development options with ISO/IEC 9945.

Appendix A: Background Information

This appendix contains background information on the POSIX standards and the Linux Standard Base specification.

A.1 POSIX Standards

The POSIX standards are the foundations of the UNIX system and Linux API sets. The development body for the POSIX standards has been the IEEE in association with ISO/JTC1/SC22. Today, the technical development body is known as the Austin Group (see below).

This section provides an overview of the POSIX standards.

A.1.1 The Portable Application Standards Committee (PASC)

The IEEE Computer Society's Portable Application Standards Committee (PASC) is the group that has and continues to develop the POSIX family of standards. Historically, the major work has been undertaken within Project 1003 (POSIX) with the best known standard being IEEE Std 1003.1 (also known as POSIX 1003.1, colloquially termed “dot 1”). The goal of the PASC standards has been to promote application portability at the source code level.

More information about PASC is available from www.pasc.org.

A.1.2 IEEE POSIX 1003.1 System Application Interface (C API)

Historically, this has been the base standard upon which the POSIX family of standards has been built. In keeping with its original focus on the UNIX system, it is aimed at interactive timesharing computing environments. The latest version of this standard was produced by the Austin Group (see later). In general, the Linux operating system aims to comply with the POSIX 1003.1 standard.

The first edition of IEEE Std 1003.1 was published in 1988. Subsequent editions were published in 1990, 1996, and 2001. The 1990 edition was a revision to the 1988 edition and became the stable base standard onto which further amendments were added. The 1990 edition was also approved as an international standard – ISO/IEC 9945-1:1990.

The 1996 edition added the IEEE Std 1003.1b-1993, IEEE Std 1003.1c-1995, and 1003.1i-1995 amendments to the base standard, keeping the stable core text unchanged. The 1996 edition of IEEE Std 1003.1 was also approved as an international standard – ISO/IEC 9945-1:1996.

In 1998, the first real-time profile standard – IEEE Std 1003.13-1998 – was published, enabling POSIX to address embedded real-time applications and smaller footprint devices.

In 1999, the decision was taken to commence the first major revision to the core base standard in ten years, including a merger with the 1003.2 standards for Shell and Utilities which had been a separate standard up to this point. It was agreed that this work be undertaken by the Austin Group (see later). As part of this decision, the PASC decided to cease rolling amendments to the base standard after completion of IEEE Stds 1003.1a, 1003.1d, 1003.1g, 1003.1j, 1003.1q, and 1003.2b. These projects were rolled into the 2001 edition of IEEE Std 1003.1. It was decided to convert other projects in progress to standalone documents.

A.1.3 IEEE POSIX 1003.2 Shell and Utilities

This standard defines a standard source-level interface to the shell and utility functionality required by application programs, including shell scripts. This standard has been incorporated into IEEE Std 1003.1-2001 produced by the Austin Group. The compliance level of the Linux operating system is harder to determine for the shell and utilities.

A.1.4 IEEE POSIX Standards for Real-Time

The PASC Real-Time System Services Working Group (SSWG-RT) has developed a series of standards that amend IEEE Std 1003.1-1990 and a profile standard (IEEE Std 1003.13-1998).

The real-time amendments to IEEE Std 1003.1-1990 are as follows:

IEEE Std 1003.1b-1993 Realtime Extension

IEEE Std 1003.1c-1995 Threads

IEEE Std 1003.1d-1999 Additional Realtime Extensions

IEEE Std 1003.1j-2000 Advanced Realtime Extensions

IEEE Std 1003.1q-2000 Tracing

These have all been folded in as options within the revision project undertaken by the Austin Group (see below).

The real-time profile is known as IEEE Std 1003.13-1998. At the time of writing, there is a revision to IEEE Std 1003.13-1998 in progress to align it with IEEE Std 1003.1-2001. This project is currently known as IEEE P1003.13-200x.