Technical Report (informative)

Title: Conflicts between ISO/IEC 9945 (POSIX) and the Linux Standard Base.

Document Number _____

Status: Draft

Author: Andrew Josey, UK National Body

Version 1.0

Date: 12 August 2003

1.Introduction

1.1 Purpose

The purpose of this Type 3 Technical Report (informative) is to document the areas of conflict between ISO/IEC 9945 (POSIX) and the Free Standards Group's Linux Standard Base specification 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 standard in use throughout the world. There is a significant investment in applications developed for the ISO POSIX 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 and the ISO POSIX standards. There is an essential market requirement that the conflicts be resolved so that an application can be written to conform to both standards. Hundreds of millions of dollars of applications are built upon these standards. This report is intended as a starting point to look at resolution of this issue.

1.2 Scope

The JTC 1/SC 22 Linux Study Group meeting (May 2003) recommended future action towards adopting Linux as a JTC1 standard, and most likely adopting the Linux Standard Base (LSB) specification as a Publicly Available Specification (PAS). The Free Standards Group is in the process of applying for PAS submitter status to JTC 1 The Free Standards Group intend to submit the LSB for PAS approval. The scope of this technical report is to identify areas of conflict between the LSB 1.9 specification and the ISO/IEC 9945 (POSIX) standard.

This report is based on the LSB Common Specification 1.9.0-Snapshot dated 6th August 2003, and the forthcoming ISO/IEC 9945:2003 edition dated 15th August 2003.

1.3 Intended Audience

This document is intended to be submitted to JTC1 as a Technical Report. It is anticipated that they should distribute it to workgroups such as the Austin Group and the Linux Standard Base for which it is in scope. It is also intended to be of interest to systems engineers, technical managers and procurement officers.

1.4 Document Overview

This document is organized in the following ways: Section two provides a list of differences that could be possible conflicts or extensions in the System Interfaces. Section three provides a list of differences that could be possible conflicts or extensions in the Shell and Utilities.

1.5 Acknowledgements

Extracts of this document are quoted from the ISO/IEC 9945:2003 and Linux Standard Base documents.

Thanks to Paul Eggert for detailed feedback on the Utility Interfaces section of the document.

2. 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 work in progress version of the LSB specification (LSB Common 1.9.0-Snapshot). 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 manual page rather than referencing ISO/IEC 9945 then that is noted here and its possible that further investigation might determine that there is no conflict.

2.1 Interface definitions

2.1.1 fcntl

LSB permits implementation to set O_LARGEFILE

According to the ISO/IEC 9945, only an application sets fcntl flags, such as O_LARGEFILE. However, the LSB specification also allows implementations to set O_LARGEFILE in a case in which the default behavior matches the O_LARGEFILE behavior, for example off_t is 64 bits. The impact is that applications when calling fcntl with the F_GETFL command may receive the O_LARGEFILE flag set as well as the flags explicitly set by the application.

2.1.2 gethostbyname

The LSB has its own definition of gethostbyname() and does not reference ISO/IEC 9945.

2.1.3 getopt

The LSB documents a number of GNU extensions to getopt() as well as descriptions of the POSIX requirements.
It also references a PASC Interpretation 1003.2 #150, which is incorporated into ISO/IEC 9945 and thus no longer relevant.

2.1.4 gets

The LSB has deprecated the gets() function, whereas it is a first class function in ISO/IEC 9945 and ISO/IEC 9899. Both the LSB and ISO/IEC 9945 recommend use of fgets() in place of gets().

2.1.5 getservbyname

The LSB has its own definition of getservbyname() and does not reference ISO/IEC 9945.

2.1.6 getservent

The LSB has its own definition of getservent() and does not reference ISO/IEC 9945.

2.1.7 ioctl

The LSB has its own definition of ioctl() and does not reference ISO/IEC 9945. As well as a general ioctl() interface this also includes the definition of a socket ioctl() interface.

2.1.8 iswctype

The LSB has its own definition of iswctype() and does not reference ISO/IEC 9945.

2.1.9 kill

Process ID -1 doesn't affect 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?", Linux Weekly News, 20 December 2001, http://lwn.net/2001/1220/kernel.php3

On an editorial note the LSB text for this interface refers to Version 2 of the Single UNIX Specification and could be updated to Version 3 which is equivalent to ISO/IEC 9945.

2.1.10 nice

LSB permits as deprecated behavior, the return value of a successful call to nice() to be 0 (rather than the new nice value). A future version of the LSB is expected to require the new nice value, as specified in the ISO/IEC 9945. Until then, applications need to call the getpriority function, rather than rely on the return value from nice() on LSB systems.

2.1.11 opterr, optind, optopt

The LSB has its own definition of opterr, optind and optopt and does not reference ISO/IEC 9945.

2.1.12 strptime

The LSB documents an issue with limiting the number of leading zeroes. This may be a conflict and needs further investigation as to the interpretation of ISO/IEC 9945 .

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 are 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."

On an editorial note the LSB text for this interface refers to Version 2 of the Single UNIX Specification and could be updated to Version 3 which is equivalent to ISO/IEC 9945.

2.1.13 strtok_r

The LSB has its own definition of strtok_r() and does not reference ISO/IEC 9945.

2.1.14 system

The LSB has its own definition of system() and does not reference ISO/IEC 9945.

2.1.15 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)."

On an editorial note the LSB text for this interface refers to Version 2 of the Single UNIX Specification and could be updated to Version 3 which is equivalent to ISO/IEC 9945.

2.1.16 waitid

The LSB has deprecated the waitid() function, whereas it is a first class function in ISO/IEC 9945 (but in the XSI option group).

2.1.17 waitpid

The LSB does not require implementations to support the WCONTINUED or WIFCONTINUED functionality within waitpid().

2.2 Pthreads Behavior

LSB permits implementations to partially support the ISO/IEC 9945 pthreads definitions. It is noted that this may change in a future revision of the LSB.

LSB states the current exceptions as follows:

"POSIX specifies a concept of per-process rather than per-thread signals. The LSB does not require this behavior; traditional Linux implementations have had per-thread signals only. A related issue is that applications cannot rely on getpid() returning the same value in different threads.

Note: one implication of per-thread signals is that a core dump (for example) may not stop all threads in a given process. This may be an issue when designing ways to stop/start applications.

Applications which create child processes (using fork() and the like) must then wait for them (using waitpid() family of functions) in the same thread as they created them. Note that coding applications this way will work both with full POSIX threads and legacy Linux thread implementations.

POSIX specifies that changing the user or group id instantly affects the behavior of all threads. This behavior is not specified; applications must use their own lock if they need this behavior. Rationale: it seems unnecessary and it is a performance hit (an SMP kernel must lock the user id).

Although this standard doesn't have a way to list processes (/proc or "ps" command line isn't in, right?), it is our intention to not specify one way or the other whether multiple threads appear as separate processes or as a single process.

Applications cannot rely on resource limits (getrusage and setrusage) being maintained per-process rather than per-thread.

Applications must disconnect from the controlling tty before calling pthread_create.

times() need not account for all threads; it may just account for the caller.

Applications must not call pthread_cancel if they call any system libraries (most notably X Window System libraries), as system libraries are not guaranteed to be thread safe. Likewise, for such libraries, only one thread per process may call them.

Applications cannot rely on fcntl/lockf locks being visible per-process rather than per-thread. Likewise for mandatory file locks.

Threaded applications cannot use SIGUSR1 or SIGUSR2."

3. 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 work in progress version of the LSB specification (LSB Common 1.9.0-Snapshot). 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 manual page rather than referencing ISO/IEC 9945 then that is noted here and its possible that further investigation might determine that there is no conflict.

3.1 Utility definitions

LSB does not describe the effect of environment variables on the behavior of the shell and utilities. Since the LSB documents the GNU utility behavior it needs to consider documenting the effects for example of setting the environment variables POSIXLY_CORRECT and _POSIX2_VERSION. With many GNU utilities, setting one or more of these environment variables can alter the utility behavior to be POSIX conforming, for example POSIXLY_CORRECT must be set in the environment, otherwise argument-option processing does not conform to POSIX. Some GNU utilities also require _POSIX2_VERSION=200112 to be set in the environment, if they are to conform to ISO/IEC 9945-3:2003 rather than to ISO/IEC 9945-2:1993. For example "sort input -o output" is disallowed by the newer standard, but is required by the older one.

3.1.1 ar

The LSB lists the following differences:

-T, -C
     need not be accepted.

-l
     has unspecified behavior.

-q
     has unspecified behavior; using -r is suggested.

3.1.2 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 reside in /etc rather than /usr/lib/cron
on LSB implementations.

3.1.3 awk

The LSB lists the following differences:

Certain aspects of internationalized regular expressions are optional.

3.1.4 batch

The LSB lists the following differences:

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

3.1.5 bc

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

3.1.6 chown

The following is listed by the LSB as a difference but it is
probably an extension.

"The use of the '.' character as a separator between the specification of
the user name and group name is supported (in addition to the use of the
':' character as specified in the Single UNIX Specification)."

3.1.7 cpio

The LSB lists the following differences:

Certain aspects of internationalized filename globbing are optional.

3.1.8 crontab

The LSB lists the following differences:

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

3.1.9 cut

The LSB lists the following difference:

-n has unspecified behavior.

3.1.10 df

The LSB lists the following differences:

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

If an argument is the absolute file name of a disk device node containing a mounted filesystem, df shows the space available on that filesystem rather than on the filesystem containing the device node (which is always the root filesystem).

3.1.11 du

The LSB lists the following differences:

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

3.1.12 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.

3.1.13 find

The LSB lists the following differences:

Certain aspects of internationalized filename globbing are optional.

3.1.14 fuser

The LSB lists the following differences:

-c has unspecified behavior.

-f has unspecified behavior.

3.1.15 grep

The LSB lists the following differences:

Certain aspects of internationalized regular expressions are optional.

3.1.16 ipcrm

The LSB has its own definition of ipcrm and does not reference ISO/IEC 9945.

3.1.16 ipcs

The LSB has its own definition of ipcs and does not reference ISO/IEC 9945.

3.1.17 ls

The LSB lists the following differences:

-p
     in addition to the behavior of printing a slash for a directory, ls -p may display other characters for other file types.

Since the ISO/IEC 9945 only defines the behavior for directories this appears to be an upward compatible extension.

The LSB states that certain aspects of internationalized filename globbing are optional.

3.1.18 m4

The LSB lists these as differences,

-P forces a m4_ prefix to all builtins.

-I directory

  Add directory to the end of the search path for includes.

These appear to be upward compatible extensions.

3.1.19 more

The LSB lists these as 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.

3.1.20 newgrp

The LSB has its own definition of newgrp and does not reference ISO/IEC 9945.

3.1.21 od

The LSB lists these as differences.

-w, --width[=BYTES]

outputs BYTES bytes per output line.

--traditional

accepts arguments in pre-POSIX form described below

Pre-POSIX Specifications

The LSB supports option intermixtures with the following pre-POSIX specifications:

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

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

-h
is equivalent to -t x2, selects hexadecimal shorts.

-i
is equivalent to -t d2, selects decimal shorts.

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

3.1.22 patch

The LSB lists the following differences:

--binary
reads and write all files in binary mode, except for standard output and /dev/tty. This option has no effect on POSIX-compliant systems.

-u, --unified
interprets the patch file as a unified context diff.

3.1.23 renice

The LSB lists the following differences:

-n increment

has unspecified behavior.

3.1.24 sed

The LSB lists the following differences:

Certain aspects of internationalized regular expressions are optional

3.1.25 split

The LSB lists the following differences:

-a suffix_length

has unspecified behavior but is expected to align with ISO/IEC 9945 in the future.

3.1.26 uname

The LSB lists the following differences:

-a
prints all information (not just the options specified in ISO/IEC 9945

3.1.27 wc

The LSB lists the following differences:

-m
has unspecified behavior. The LSB will require support for this as specified in ISO/IEC 9945 in a future revision.

3.1.28 xargs

The LSB lists the following differences:

-E
has unspecified behavior.

-I
has unspecified behavior.

-L
has unspecified behavior.

3.1.29 Utility related Sourceforge Bugs raised

As a result of this section of the report, and investigation of the latest coreutils package (that implements many common utilities for Linux) a number of bugs have been raised against the LSB specification. The summary list follows below:

Request ID	Summary
783373	wc -m supported?
783371	uname -a behavior conforming?
783369	split -a supported?
783368	patch: differences vs extensions
783366	od description extensions
783365	echo behavior is conforming to POSIX
783361	du: POSIXLY_CORRECT behavior
783359	df POSIXLY_CORRECT behavior
783357	chown utility: _POSIX2_VERSION
783355	Utility descriptions should describe environment variables

3.2 Internationalization

The LSB makes certain internationalization aspects optional.

3.2.1 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 (including egrep), sed, and awk.

3.2.2 Filename Globbing

Utilities that perform filename globbing (also known as Pattern Matching Notation) shall do it 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.

3.3 Shell Exceptions

The LSB documents the following exceptions for the shell (sh utility) from ISO/IEC 9945.

3.3.1 Pathname of $0

When the shell searches for a command name in the PATH and finds a shell script, ISO/IEC 9945 specifies that it shall pass the command name as argv[0] and in the child shell script, $0 shall be set from argv[0].

(Note there is a defect report pending on this issue)

However, for an LSB shell, the system may implement either this behavior or $0 may be set to an absolute pathname of the shell script.

3.3.2 Sourcing non-executable files

When PATH is used to locate a file for the dot utility, and a matching file is on the PATH but is not readable, the behavior is undefined (unlike ISO/IEC 9945 which LSB states requires the shell to continue searching through the rest of the PATH , see the "dot" man page under "Special built in utilities")

3.3.3 Globalized Pattern Matching

For filename globbing, globalized implementations shall provide the functionality defined 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.

APPENDIX A

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

A1. 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/WG15.

This section provides an overview of the POSIX standards.

A1.1 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 the Portable Application Standards Committee (PASC) is available from:
http://www.pasc.org

A1.2 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.

A1.3 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 the 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.

A1.4 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 current known as IEEE P1003.13-200x.

A1.5 The Austin Group

The Austin Group is the working group that manages the POSIX.1 specificaton. It is a joint working group of members of the IEEE Portable Applications Standards Committee, members of The Open Group, and members of ISO/IEC Joint Technical Committee 1. Participation is free and open to all interested parties.

The Austin Group arose out of discussions amongst the parties which started in early 1998, which led to an initial meeting and formation of the group in September 1998. The purpose for this group has been to revise, combine, and update the following standards: ISO/IEC 9945-1, ISO/IEC 9945-2, IEEE Std 1003.1, IEEE Std 1003.2, and the Base Specifications of The Open Group Single UNIX Specification.

After two meetings, an agreement was signed in July 1999 between The Open Group and the Institute of Electrical and Electronics Engineers (IEEE), Inc., to formalize the project with the first draft of the

revised specifications ("the revision") being made available at the same time. Under this agreement, The Open Group and IEEE agreed to share joint copyright of the resulting work.

The base document for the revision was The Open Group's Base volumes of its Single UNIX Specification, Version 2. These were selected since they were a superset of the existing POSIX.1 and POSIX.2 specifications and had some organizational aspects that would benefit the audience for the new revision.

The approach to specification development has been one of "write once, adopt everywhere", with the deliverables being a set of specifications that carry the IEEE POSIX designation, The Open Group's Technical Standard designation, and an ISO/IEC designation (see below). This set of specifications also forms the core of the Single UNIX Specification, Version 3. The Open Group and the IEEE approved the Austin Group specifications in late 2001, as The Open Group Base Specifications, Issue 6, and IEEE Std 1003.1-2001 respectively. ISO/IEC approval followed about twelve months later.

The Austin Group Specifications consist of the following :

* Base Definitions, Issue 6 (XBD)
* Shell and Utilities, Issue 6 (XCU)
* System Interfaces, Issue 6 (XSH)
* Rationale (Informative)

The revision has tried to minimize the number of changes that implementations which conform to the earlier versions of the approved standards would require to bring them into conformance with the current standard. Specifically, the scope of the project excluded doing any ``new'' work, but rather collecting into a single document what had been spread across a number of documents, and presenting it in what had been proven in practice to be a more effective way. Some changes to prior conforming implementations were unavoidable, primarily as a consequence of resolving conflicts found in prior revisions, or which became apparent when bringing the various pieces together. Also, since the revision now references the 1999 version of the ISO C standard, there are a number of unavoidable changes that have been made which will affect applications portability.

In early 2003, the Austin Group obtained approval for Technical Corrigendum 1, and the 2003 edition of the specifications have been published.

More information on the Austin Group, including how to join and participate is available from its web site at
http://www.opengroup.org/austin/

A html version of the specification is freely available from The Open Group's Single UNIX Specification web site at
http://www.unix.org/version3/

A1.5.1 Relationship to the ISO C Standard

The most recent revision to the ISO C standard occurred in 1999. The ISO C standard is itself independent of any operating system in so

much as it may be implemented in many environments including hosted environments.

The POSIX and Single UNIX Specification have a long history of building on the ISO C standard and deferring to it where applicable. Revisions of POSIX.1 prior to the Austin Group specification built upon the ISO C standard by reference only, and also allowed support for traditional C as an alternative. The Single UNIX Specification in contrast, included manual pages for the ISO C interfaces.

The Austin Group took the latter approach. The standard developers believed it essential for a programmer to have a single complete reference place. They also recognized that deference to the formal standard had to be addressed for the duplicate interface definitions which occur in both the ISO C standard and their document.

It was agreed that where an interface has a version in the ISO C standard, the DESCRIPTION section should describe the relationship to the ISO C standard and markings added as appropriate within the manual page to show where the ISO C standard has been extended.

A block of text was added to the start of each affected reference page stating whether the page is aligned with the ISO C standard or extended. Each page was parsed for additions beyond the ISO C standard (that is, including both POSIX and UNIX extensions), and these extensions are marked as CX extensions (for C Extensions).

A1.6 ISO/IEC 9945

In late 2002, the ISO/IEC Joint Technical Committee approved the joint revision to POSIX and the Single UNIX Specification as an International Standard. Designated as ISO/IEC 9945:2002, the joint revision forms the core of The Open Group's Single UNIX Specification Version 3 (IEEE 1003.1-2001, POSIX.1).

The combining of the IEEE POSIX specifications and the Single UNIX Specification into ISO/IEC 9945:2002 Parts 1 to 4 replaces the existing ISO/IEC 9945-1:1996 (IEEE 1003.1, 1996 version), and ISO/IEC 9945-2:1993 (IEEE Std 1003.2, 1992 version).

ISO/IEC 9945 consists of the following parts, under the general title Information technology Portable Operating System Interface (POSIX):

Part 1: Base Definitions
Part 2: System Interfaces
Part 3: Shell and Utilities
Part 4: Rationale

The 2003 Edition of the Austin Group specifications is to be published as ISO/IEC 9945:2003 on August 15 2003.

A2.2 The Linux Standard Base Specification

The Linux Standard Base (LSB) Specification is an application binary interface standard for shrink-wrapped applications. The purpose is to

allow commonality amongst the many Linux distributions.

The LSB draws on the source standards of IEEE POSIX 1003.1-1990 and The Open Group Single UNIX Specification for many of its interfaces although does not formally defer to them preferring to document any differences where they exist, such as where certain aspects of Linux cannot currently conform to the industry standards, one particular example being the area of threads. Some interfaces are not included in the LSB, since they are outside the remit of a binary runtime environment, typically these are development interfaces or user level tools . The LSB also extends the source standards in other areas (such as graphics), and includes the necessary details such as the binary execution file formats to support a high volume application platform.

Although in theory the LSB is not tied to the GNU/Linux operating system, in practise the binary definitions are tightly coupled to the Linux operating system and the GNU C compiler.

The LSB is available as a family of specifications supporting a number of processor architectures including IA32, PPC32, PPC64, IA64, S390 and S390X. There is a generic specification, common to all the processor architectures known as the "generic LSB" (or gLSB), and for each processor architecture an architecture-specific specification ("archLSB") describing the details that vary by processor architecture.

To support the specification, the LSB includes a number of development tools, including test suites, and a set of reference conforming applications. Binary versions of the test suites and reference applications are used for formal LSB certification of runtime environments. All the major Linux vendors today have certified LSB systems.

LSB 1.2, introduced in January 2002, was the first version of the specification to have an equivalent LSB certification program. LSB 1.2 certification, which commenced in July 2002 is limited to the IA32 ABI. LSB 1.3 certification includes additional support for the IA64, PPC32, PPC64, S390 and S390X architectures. At the time of writing there are twenty-two certified runtime environments from 9 vendors.

The specification is evolving quite rapidly. LSB 1.3, introduced in January 2003, adds some internationalization, PAM, packaging, static C++ linking, bug fixes, plus IA64, PPC32, and soon PPC64, S390, S390X, and maybe Hammer. LSB 2.0 is planned for Janary 2004.

Detailed information on the LSB is available from:
http://www.linuxbase.org

Detailed information on the LSB Certification Program is available from the LSB Certification Authority at
http://www.opengroup.org/lsb/cert/

The Guide to LSB Certification is available at:
http://www.opengroup.org/lsb/cert/docs/LSB_Certification_Guide.html

The LSB Certification Register can be viewed at:
http://www.opengroup.org/lsb/cert/register.html