Title: API standards for the Linux operating system.

Document Number: _____

Source: Andrew Josey, The Open Group
Editor of ISO/IEC 9945:2002 (POSIX)

Paper submitted to JTC 1/SC 22 Study Group on Linux

Date: 21 Apr 2003

Section 1

1. Introduction

1.1 Purpose

This paper gives a commentary on key API standards and specifications relating to the Linux operating system. This includes a look at the formal IEEE POSIX standards, and industry led efforts such as The Open Group's Single UNIX Specification, the Free Standards Group Linux Standard Base Specification and the Embedded Linux Consortium Platform Specification.

1.2 Scope

This document offers a guide in understanding the various standards applicable to Linux and UNIX systems, including real-time and embedded systems. Detailed information about the contents of the standards and specifications is outside of the scope of this document.

1.3 Intended Audience

This document is being submitted to the JTC 1/SC 22 Study Group on Linux to provide background information on the current status of the Linux operating system with respect to ongoing standardization efforts.

1.4 Document Overview

This document is organized in the following ways: Section two provides an overview of the IEEE POSIX Standards. Section three provides an overview of related industry specifications. Section four looks at Profiles.

Section 2

2. IEEE POSIX Standards

The Linux operating system builds on the foundations of the POSIX standards which themselves are ISO/IEC standards. This section provides an overview of the IEEE POSIX standards, notably IEEE Std 1003.1 and IEEE Std 1003.13.

2.1 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

2.2 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 (see Profiles).

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.

2.3 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.

2.4 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) in producing IEEE Std 1003.1-2001.

The Real-time profile is known as IEEE Std 1003.13-1998. This is discussed in further detail in another section of this document. 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. In general only few of the realtime features are supported at the present time in the commercial versions of the Linux operating system. There are some specialist versions of Linux that support some of the features.

2.5 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/

2.5.1 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).

2.6 ISO/IEC 9945:2002

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

Section Three

3. Related Industry Standards

This section looks at related industry standards initiatives that have built on or taken similar approaches to the IEEE POSIX standards and that are having a direct influence on the Linux operating system.

3.1 The Single UNIX Specification

The Open Group has been the custodian of the specification for the UNIX system and the trademark since 1993. This is a source level API specification which has traditionally built upon the formal IEEE POSIX standards. It is vendor neutral and not tied to any particular implementation.

The project that led to the creation of the Single UNIX Specification started when several vendors (Sun Microsystems, IBM, Hewlett-Packard, Novell/USL, and OSF) joined together to provide a single unified specification of the UNIX system services. By implementing a single common definition of the UNIX system services, third-party independent software vendors (ISVs) would be able to more easily deliver strategic applications on all of these vendors' platforms at once.

A two-pronged approach was used to develop the Single UNIX Specification. First, a set of formal industry specifications was chosen to form the overall base for the work. This would provide stability, vendor neutrality, and lay a well charted course for future application development, taking advantage of the careful work that has gone into developing these specifications. It would also preserve the portability of existing applications already developed to these core models.

The XPG4 Base (1992) was chosen as the stable functional base from which to start. XPG4 Base supports the POSIX.1 system interface and the ISO C standards at its core. It also provided a rich set of 174 commands and utilities.

To this base was added the traditional UNIX System V Interface Definition, (SVID) Edition 3, Level 1 calls, and the OSF Application Environment Specification Full Use interface definitions. This represented the stable central core of the latter two specifications.

The second part of the approach was to incorporate interfaces that were acknowledged common practice but had not yet been incorporated into any formal specification or standard. The intent was to ensure existing applications running on UNIX systems would port with relative ease to a platform supporting the Single UNIX Specification. A survey of real world applications was used to determine what additional interfaces would be required in the specification.

Fifty successful application packages were chosen to be analyzed using the following criteria:

- Ranked in International Data Corp's. 1992, 'Survey of Leading UNIX Applications',

- The application's domain of applicability was checked to ensure that no single application type (for example, databases) was overly represented,

- The application had to be available for analysis either as source code, or as a shared or dynamic linked library.

From the group of fifty, the top ten were selected carefully, ensuring that no more than two representative application packages in a particular problem space were chosen. The ten chosen applications were:

AutoCAD; Cadence; FrameMaker; Informix; Island Write/Paint; Lotus 1-2-3; SAS (4GL); Sybase; Teamwork; WordPerfect

APIs used by the applications that were not part of the base specifications were analyzed:

- If an API was used by any of the top ten applications, it was considered for inclusion.

- If an API was not used by one of the top ten, but was used by any three of the remaining 40 applications, it was considered for inclusion.

- While the investigation of these 50 applications was representative of large complex applications, it still was not considered as a broad enough survey, so an additional 3500 modules were scanned. If an API was used at least seven times in modules that came from at least two platforms (to screen out vendor specific libraries), then the interface was considered for inclusion.

When the survey was complete, there were 130 interfaces that did not already appear in the base specification. These interfaces were predominantly BSD interfaces that had never been covered in XPG4 Base, the SVID, or the AES, but did represent common practice in UNIX system applications developed originally on BSD-derived platforms. Such things as sockets and the 4.3BSD memory management calls were commonly used in many applications.

The goal was to ensure that APIs in common use were included, even if they were not in the formal specifications that made up the base. Making the Single UNIX Specification a superset of existing base specifications ensured any existing applications should work unmodified.

The Single UNIX Specification has evolved through several iterations; Version 2 in 1997 incorporated updates to the formal standards, as well as industry driven additions such as large file handling, dynamic linking, datasize neutrality and extended threads functionality.

Together with the specification, The Open Group has provided a certification program, the latest instance of which is UNIX 98 for the Single UNIX Specification Version 2. This is supported by a family of test suites providing over 95% coverage of the specification. The certification program has been cited in many major procurements including NASA SEWP III.

The Open Group also makes a number of test tools for the UNIX System freely available for measuring conformance to its specifications, see http://www.opengroup.org/testing/downloads.html

The majority of the latest version of the Single UNIX Specification, Version 3, was produced by the Austin Group (see earlier), and comprises the Base Specifications, Issue 6 , plus X/Open Curses, Issue 4, Version 2.

Detailed information on the Single UNIX Specification, including accessing the version 3 specification in html is available at
http://www.unix.org/

A list of the interfaces in Version 3 of the Single UNIX Specification together with comparative information on the presence of the interface in other specifications is available at
http://www.unix.org/v3-apis.html

HTML and PDF versions of many of the specifications for the Single UNIX Specification are available from The Open Group's online publication catalog at
http://www.opengroup.org/publications/

Information on the UNIX certification program which operates under The Open Group's Open Brand, can be found at
http://www.opengroup.org/openbrand/

The Practical Guide to the Open Brand is available at
http://www.opengroup.org/openbrand/Certification_Guide/

The register of Certified Products is available at
http://www.opengroup.org/openbrand/register/

3.2 The Linux Standard Base Specification

The Linux Standard Base (LSB) Specification is an application binary interface standard for shrink-wrapped applications. The LSB draws on the source standards of IEEE POSIX 1003.1-1990 and The Open Group Single UNIX Specification Version 2 for many of its interfaces although does not formally defer to them preferring to document any differences where they exist. 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 and IA64. 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 support for the IA64 and PPC32
At the time of writing there are nineteen 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

3.3 The Embedded Linux Consortium Platform Specification

The Embedded Linux Consortium Platform Specification (ELCPS) is a profile specification built upon the Linux Standard Base Specification. This is a source level API profile and is not intended to specify a binary interface. Its target is embedded devices running the Linux operating system. This is discussed in further detail in the next section on profiles.

Section Four

4. Profiles.

Profiles are an important concept in procurement of real-world systems based on IEEE POSIX. They address multiple problems, firstly the proliferation of standards, which can lead to confusion and sometimes conflicting standards; secondly, they overcome some of the generality of the base standards, which include multiple options and have a large range of applicability; thirdly, they provide ability to subset the base standard into smaller building blocks which can be combined to more closely fit specific smaller footprint devices and systems.

4.1 What is a Profile?

A profile is a collection of one or more specifications, or in the case of IEEE Std 1003.13 and the Embedded Linux Consortium Platform Specification a collection of certain subsets of a standard, that can be used to define a certain application environment. A profile may be specified at various different levels of detail, for example, the Linux Standard Base Specification profiles over thirty other specifications; the UNIX 98 Workstation profile collects the UNIX 98 requirements and the Common Desktop Environment requirements together without much details and is thus a "thin profile". The converse of this is a detailed or so-called "thick profile", such as IEEE Std 1003.13 and ELCPS, which select certain options and collections of functions within an individual standard.

4.2 Profile Selection

If you are considering a procurement you have several options in stating your conformance requirements:

1. Select an existing profile
2. Tailor an existing profile -- reusing existing building blocks, either reducing functionality in a profile, or adding functionality to a profile
3. Build your own profile

There are tradeoffs you might want to consider, for example selection of an existing profile should yield higher applications portability and increase the probability of multiple vendors being able to supply systems meeting that profile, thereby increasing your choice of solutions.

4.3 IEEE Std 1003.13 Profiles

IEEE Std 1003.13-1998 is a family of four related real-time profiles ranging in size from the very small through a full-featured platform conforming to essentially all of IEEE Stds 1003.1, 1003.1b (real-time), and 1003.1c (threads), and the parallel Ada binding 1003.5b, with realtime options chosen. The smaller profiles specify just that subset of POSIX interfaces needed to "clothe" widely-used small kernels such as pSOS and VxWorks (both from Wind River), and VRTX32 (now from Mentor), and the ORKID interface standard (from VITA), which although very similar in approach and function, differ greatly in interface details.

Standardization of these interfaces is expected to yield the same benefits

for embedded and real-time systems as standardization of the UNIX system interfaces did for workstations. In addition, the 1003.13 interfaces have been chosen to allow multi-computer distributed systems to be built, such as those used in factory automation. Such systems are typically set up as a hierarchy, with a few large-profile machines at the top, and a large number of smaller profile machines at the bottom controlling this or that piece of machinery, perhaps with an intermediate layer of supervisory machines between top and base, and all communicating with peers, superiors, and subordinates, as needed.

Note that RT Linux from FSM Labs turns the 1003.13 hierarchy upsidedown, with the smaller PSE51/52 realtime threads in control, and the full-figured Linux system (similar in functionality to PSE54) as just another thread under control of the realtime threads. This is the opposite of what the PASC SSWG-RT had imagined when 1003.13-1998 was written, but it nonetheless works.

4.3.1 Minimal Real-time System Profile IEEE Std 1003.13 PSE51

This profile is intended for embedded systems, with a single multi-threaded process, no file system, no user and group support and only selected options from IEEE Std 1003.1b-1993.

4.3.2 Real-time Controller Profile IEEE Std 1003.13 PSE52

This profile is an extension of the minimal real-time system profile with added support for a file system. There is only a single multi-threaded process. Files and directories are supported, there is no user and group support, and all options from IEEE Std 1003.1b-1993 are supported except the process related ones.

Note: The above two profiles, PSE 51 and PSE 52 assume that the entire computer is the an implicit "process", with one or more POSIX threads running within. This supports hardware lacking the memory-management features that are needed to implement POSIX processes. The other odd-looking thing that is nonetheless correct is that even in profiles lacking real processes, IEEE Std 1003.13 still requires interprocess communications APIs. This is done to support the construction of hierarchies of computers from small (PSE51/52) to large (PSE54).

4.3.3 Dedicated Real-time Profile IEEE Std 1003.13 PSE53

This profile is an extension of the minimal real-time system profile with the addition of support for multiple processes. There is no file system, and no user and group support. All options from IEEE Std 1003.1b-1993 are supported except for the file-related ones.

Note that one major change between IEEE Std 1003.13-1998 and the draft revision P1003.13-200x is that PSE53 will now require a file system. This was the single most often requested change, and also was the main difference between proposed fifth profiles and the four profiles of 1003.13-1998. Observing that the RTOS market had evolved since 1003.13-1998 was developed, the PASC SSWG-RT has decided that extending PSE53 was a better solution than adding a profile.

4.3.4 Multi-Purpose Real-time Profile IEEE Std 1003.13 PSE54

This profile is intended for multi-purpose real-time applications as well as interactive users. It includes multiple processes, each of which may be multi-threaded. All base 1003.1 functionality is incorporated including a file system and user and group support. All the options from IEEE Std 1003.1b-1993 are supported. Support for IEEE Std 1003.2-1992 is also required.

Note: The above two profiles, PSE 53 and PSE 54 assume that the hardware is capable of implementing multiple processes.

4.4 The Austin Group Subprofiling Considerations

The Austin Group considered including a set of options similar to the "Units of Functionality'' contained in IEEE Std 1003.13-1998. However, as the development of IEEE Std 1003.1-2001 continued, the standard developers felt it premature to fix these in normative text.

The approach instead was to include a general requirement in normative text regarding subprofiling, which allows subprofiles to subset both mandatory and optional functionality required for POSIX Conformance or XSI Conformance. Such profiles are required to organize the subsets into Subprofiling Option Groups. An informative section, Appendix E, is included in the Austin Group specifications, containing a representative set of subprofiling options applicable to specialized realtime systems. The Austin Group specification does not require that the presence of Subprofiling Option Groups be testable at compile-time (as symbols defined in any header) or at runtime (via sysconf() or getconf).

4.4 The Embedded Linux Consortium Platform Specification (ELCPS)

The ELCPS takes a similar approach to IEEE Std 1003.13, although does not address any realtime functionality. Its purpose is to define embedded applications environments based on the Linux Operating system.

It draws on the subprofiling considerations outlined in the Austin Group specifications, and provides forty-four function groupings and three so-called System Environments (SE), the Minimal Environment, the Intermediate Environment and the Full Environment . Each SE is a superset of the one below it. An SE may optionally provide functionality not required.

4.4.1 The Minimal System Environment

The Minimal SE is intended for deeply embedded systems, with requirements only for a single multi-threaded process, no file system, no interactive users support and only selected function groupings from the LSB. It requires 371 functions from 7 function groupings.

4.4.2 The Intermediate System Environment

The Intermediate SE is an extension of the Minimal SE with the addition of required support for multiple processes, file system, asynchronous I/O and

dynamic linking. It requires 667 functions from 27 function groupings.

4.4.3 The Full System Environment

The Full SE provides a rich multipurpose Linux operating system environment, with the omission of any utilities. It requires 1121 functions coming from 44 groups.

4.5 FIPS Profiles

The Federal Information Processing Standards (FIPS) are a series of U.S. government procurement standards managed and maintained on behalf of the U.S. Department of Commerce by the National Institute of Standards and Technology (NIST). During the 1990s NIST produced a number of FIPS profiles (FIPS 151-1, FIPS 151-2 and FIPS 189) to support compatibility and interoperability among US Government systems implementing POSIX.

FIPS 151-2 superseded FIPS 151-1 and profiled IEEE Std 1003.1-1990 selecting certain options from the base standard and raising certain minimum values for implementation defined constants. The FIPS 151-2 requirements only mapped to the base standard (IEEE Std 1003.1-1990) and these have now been folded into the revision of 1003.1.
FIPS 189 was based on IEEE Std 1003.2-1992.

FIPS 151-2 and FIPS 189 have now been withdrawn.

4.6 UNIX System Profiles

The Open Group UNIX System profiles build upon the POSIX standards using them as building blocks. Three profiles are included for UNIX 98 which is the mark for systems conforming to the Single UNIX Specification, Version 2.

UNIX 98 is the base profile standard providing a rich programming environment.

UNIX 98 Workstation is the base standard plus the Common Desktop Environment (CDE) graphical interface and programming libraries.

UNIX 98 Server is the base standard plus server related functionality for internet/intranet services and Java support.

Since these build upon the base standard they are multi-user, timesharing systems and not directly suitable as an embedded solution. Some applicability for real-time systems can be obtained by procuring a UNIX 98 platform supporting the Real-time Feature Group, thereby providing a development environment supporting all the features and options in IEEE Std 1003.1b-1993 and IEEE Std 1003.1c-1995.

The latest UNIX profile is known as UNIX 03, and supports Version 3 of the Single UNIX Specification.

Appendix A. Feature Matrix

This matrix summarizes the requirements of the profiles mentioned above.
Key:o=option, *=optional if pthreads supported, P=partial non-internationalized.

Feature	PSE 51	PSE 52	PSE 53	PSE 54	Min SE	Int SE	Full SE	FIPS 151-2	UNIX 98	LSB 1.2	UNIX 03
Processes	-	-	X	X	-	X	X	X	X	X	X
Pipes	-	-	X	X	-	X	X	X	X	X	X
Files and Directories	-	X	-	X	-	X	X	X	X	X	X
Basic I/O	X	X	X	X	X	X	X	X	X	X	X
Signals	X	X	X	X	X	X	X	X	X	X	X
Users and Groups	-	-	-	X	-	-	X	X	X	X	X
File Synchronization	X	X	X	X	X	X	X	-	X	X	X
Memory Mapped Files	-	X	-	X	-	X	X	-	X	X	X
Memory Protection	-	-	X	X	-	X	X	-	X	X	X
Process Priority Scheduling	-	-	X	X	-	-	-	-	o	-	o
Memory Locking	X	X	X	X	-	-	-	-	o	-	o
Synchronized I/O	X	X	X	X	-	-	-	-	o	-	o
Asynchronized I/O	-	X	X	X	-	X	X	-	o	o	o
Hi Resolution Clocks & Timers	X	X	X	X	-	-	-	-	o	-	o
Realtime Signals	X	X	X	X	-	-	-	-	o	-	o
Semaphores	X	X	X	X	-	-	-	-	o	-	o
Shared Memory	X	X	X	X	-	-	-	-	o	-	o
IPC Message Passing	X	X	X	X	-	-	-	-	o	-	o
Threads	X	X	X	X	*	*	*	-	X	o	X
Thread Safe Functions	X	X	X	X	*	*	*	-	X	-	X
Thread Attribute Stack Address	X	X	X	X	*	*	*	-	X	-	X
Thread Attribute Stack Size	X	X	X	X	*	*	*	-	X	-	X
Thread Process Shared	-	-	X	X	*	*	*	-	X	-	X
Thread Priority Scheduling	X	X	X	X	*	*	*	-	o	-	o
Thread Priority Inheritence	X	X	X	X	*	*	*	-	o	-	o
Thread Priority Protection	X	X	X	X	*	*	*	-	o	-	o

Feature	PSE 51	PSE 52	PSE 53	PSE 54	Min SE	Int SE	Full SE	FIPS 151-2	UNIX 98	LSB 1.2	UNIX 03
Sockets	-	-	-	-	-	-	X	-	X	X	X
XCURSES	-	-	-	-	-	-	-	-	X	P	X
ISO C89	-	-	-	-	-	-	X	X	X	X	-
ISO C99	-	-	-	-	-	-	-	-	-	-	X
Shell & Utilities											
_POSIX2_C_BIND	X	X	X	X	-	-	-	-	X	X	X
_POSIX2_C_DEV	X	X	X	X	-	-	-	-	X	-	X
_POSIX2_CHAR_TERM	-	-	-	X	-	-	-	-	X	-	X
_POSIX2_FORT_DEV	-	-	-	-	-	-	-	-	o	-	o
_POSIX2_FORT_RUN	-	-	-	X	-	-	-	-	o	-	o
_POSIX2_LOCALEDEF	-	-	-	-	-	-	-	-	X	-	X
_POSIX2_SW_DEV	X	X	X	X	-	-	-	-	o	-	o
_POSIX2_UPE	-	-	-	X	-	-	-	-	X	-	X

Appendix B. Contact Details & Permissions

The author can be contacted as follows:

Andrew Josey
The Open Group
Apex Plaza, Forbury Road,
Reading, England, RG1 1AX
Email: a.josey@opengroup.org

Permissions: