



Standards: *Bridging the Generation Gap between Linux[®] and UNIX[®] Systems*

Andrew Josey
Director of Certification
The Open Group
Email: a.josey@opengroup.org

UNIX is a registered trademark of The Open Group
Linux is a registered trademark of Linus Torvalds

Agenda

□ **This talk covers...**

- A review of the latest open systems standards
- How Linux shapes up to them

Agenda (Cont'd)

- Standards Projects
 - POSIX
 - The Single UNIX Specification
 - The Linux Standard Base (LSB)

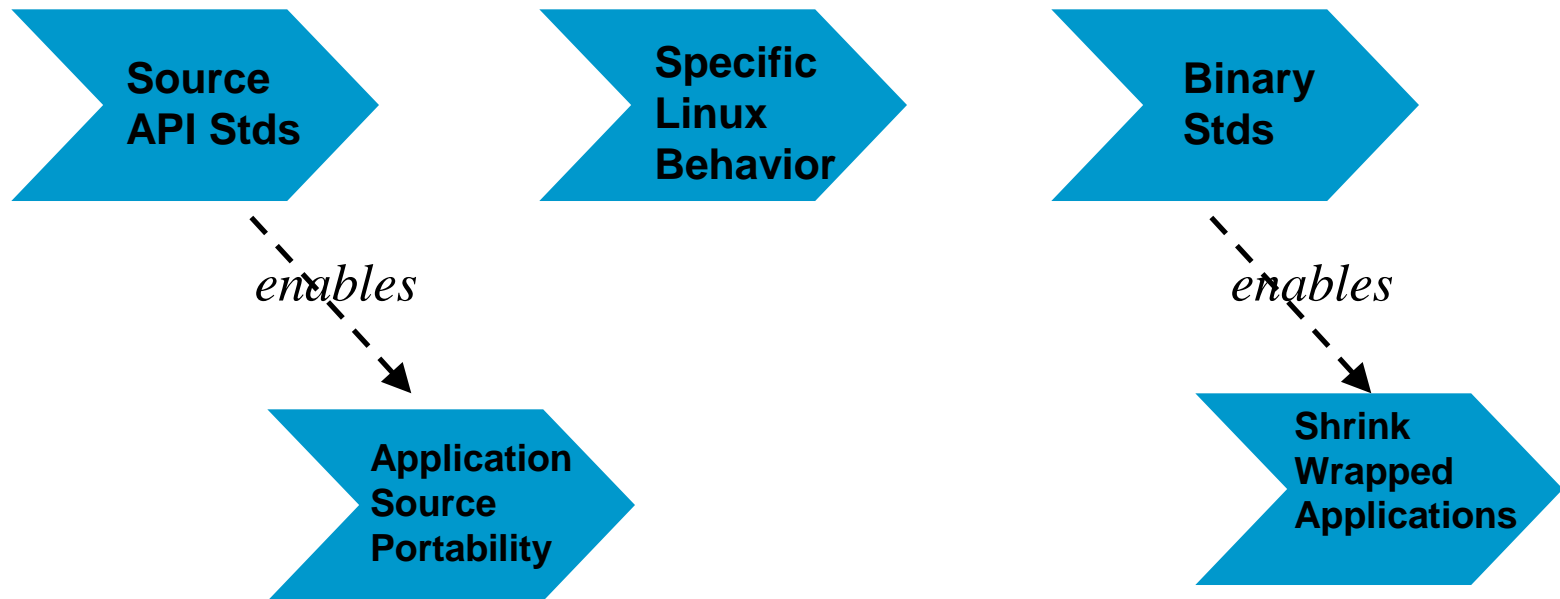
"Despite their well earned reputation as a source of confusion, standards are one of the enabling factors behind the success of Linux. If it weren't for the adoption of the right standards by Linus Torvalds and other developers, Linux would likely be a small footnote in the history of operating systems. "

- Dan Quinlan, Free Standards Group Chairman

The Free Market

- ❑ The key to the growth of the Linux market is the free-market demands placed upon suppliers by Open Standards
- ❑ These systems ultimately compete on quality and added value features to retain customers
- ❑ Dissatisfied customers can move on to another supplier

Background: Source Standards versus Binary Standards



Agenda

□ Standards Projects

➤ POSIX

- The Single UNIX Specification
- The Linux Standard Base

POSIX® /pahz-icks/

- ❑ *POSIX* , pronounced *pahz-icks* as in positive, not *poh-six*, or other variations
- ❑ POSIX is a registered trademark of the IEEE
- ❑ An acronym for *Portable Operating System Interface*

What is POSIX?

- The basic objective is portability
 - Both programmers and application source code
 - Portability of the OS kernel itself and/or application binary code are not objectives
- POSIX is a set of books specifying APIs
 - It is neither a piece of code
 - Nor an operating system

What is an API?

- ❑ Application Program Interface
- ❑ A written contract between system developers and application developers
- ❑ It is not a piece of code, it is a piece of paper defining what the two sets of developers are guaranteed to receive and are in turn responsible for providing

POSIX® /pahz-icks/

- ❑ POSIX is a family of standards developed by the Portable Applications Standards Committee (PASC) of the IEEE Computer Society
- ❑ Main subject areas:
 - System Interfaces (C, Fortran, ADA Bindings)
 - Commands & Utilities
 - Test Methods

POSIX Deliverables

- ❑ *POSIX.1–1990 System Interfaces (C language binding)*
- ❑ POSIX.2–1992 Shell & Utilities
- ❑ POSIX.5–1992 Ada bindings
- ❑ POSIX.1b–1993 Realtime Extension
- ❑ POSIX.1c–1995 Threads Extension
- ❑ POSIX.5b–1996 Ada Realtime Extension
- ❑ POSIX 1003.13–1998 Realtime Profiles

POSIX *Latest Deliverable*

- ❑ *POSIX.1-2001*
 - *Approved December 6th 2001*
- ❑ Developed by the *Austin Group (see later)*
- ❑ Supersedes POSIX.1-1990 and its amendments and POSIX.2-1992 and its amendments
- ❑ A combined system interfaces and utilities specification as a single 4000 page standard

The Success of POSIX

- ❑ The majority of UNIX system suppliers today support POSIX
- ❑ A core component of Linux
- ❑ Many other systems also now claim support for POSIX, for example VMS, MVS, MPE and even Microsoft Windows NT
- ❑ It is now so successful that it has lost its role as a *market differentiator* , it is required infrastructure

POSIX *building blocks*

- ❑ POSIX provides the foundations and building blocks for Open Systems
- ❑ Not all systems support all POSIX functionality
- ❑ Neither is POSIX functionally complete

POSIX.1-1990,.1b-1993 and Linux

- ❑ Linux mostly compatible at the source level with IEEE POSIX 1003.1-1990
- ❑ An early version was certified for FIPS 151-2 compliance (from Open Linux Ltd.)
- ❑ Linux AIO presently has problems
- ❑ Test suites are freely available to measure compliance (VSX-PCTS, LSB-OS):
 - <http://www.opengroup.org/testing/downloads.html>
 - <http://www.linuxbase.org/>

POSIX.1c-1995 and Linux

- ❑ Linux has a partial *pthread*s implementation
 - Most of the APIs are present
 - Some semantic differences
 - NGPT Threads implementation (see next slide) shows a solution possible but not yet fully accepted
- ❑ The *gLSB* includes a section giving advice to the programmer on how to work around the differences.

Next Generation POSIX Threads (NGPT)

- ❑ User-level library providing the POSIX threads API
- ❑ Uses a modified clone() interface
- ❑ Integrates into glibc as a LinuxThreads replacement
- ❑ NGPT 2.0.0 released June 2002
- ❑ Recommend kernels 2.4.19pre10 or better, or 2.5.8 or greater
- ❑ *Unclear on future vis-à-vis glibc threads functionality*

POSIX.2-1992 and Linux

- ❑ Recent distributions are getting closer to POSIX.2 and provide most of the required utilities
- ❑ Usually differences hidden by POSIXLY_CORRECT environment variable
- ❑ Known differences: split, du, df
- ❑ The *gLSB* includes a set of man pages for the commands and utilities listing the differences when applicable

Agenda

□ Standards Projects

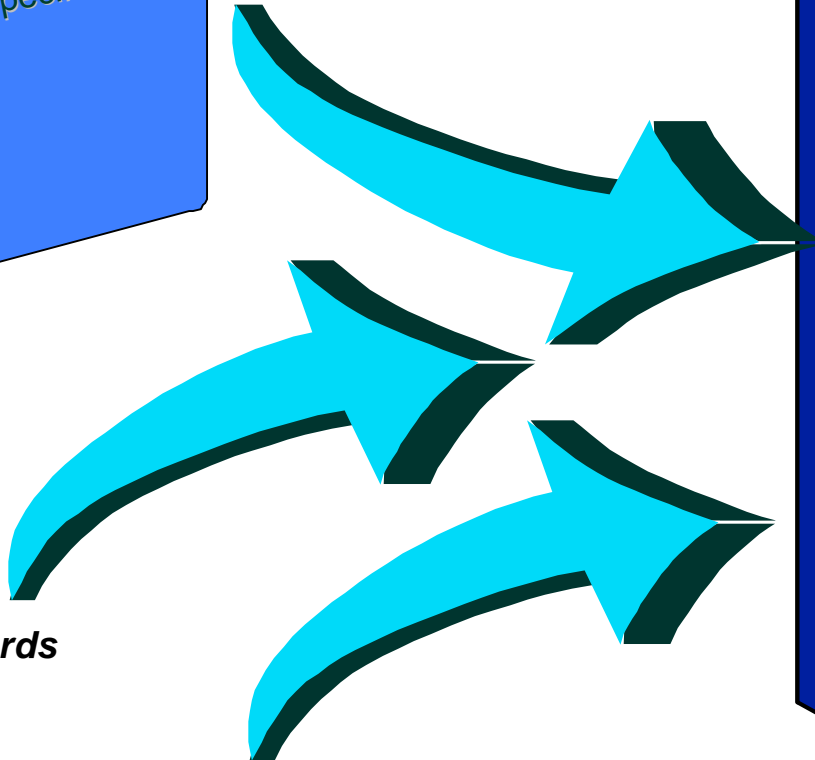
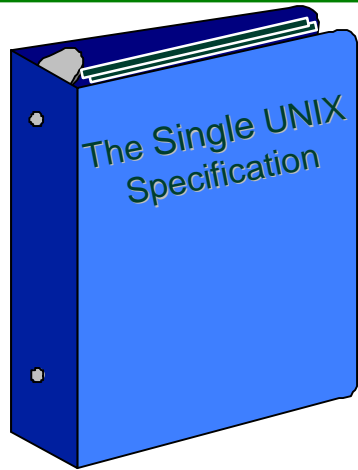
- POSIX
- **The Single UNIX Specification**
- The Linux Standard Base



The Single UNIX Specification

- ❑ Designed to give software developers a single set of APIs to be supported by every UNIX system
- ❑ Shifts the focus from incompatible UNIX system product implementations to compliance to a single set of APIs
- ❑ If an OS meets the specification and commonly available applications run on it, then it is open.

The Single UNIX Specification Version 2



Formal Standards

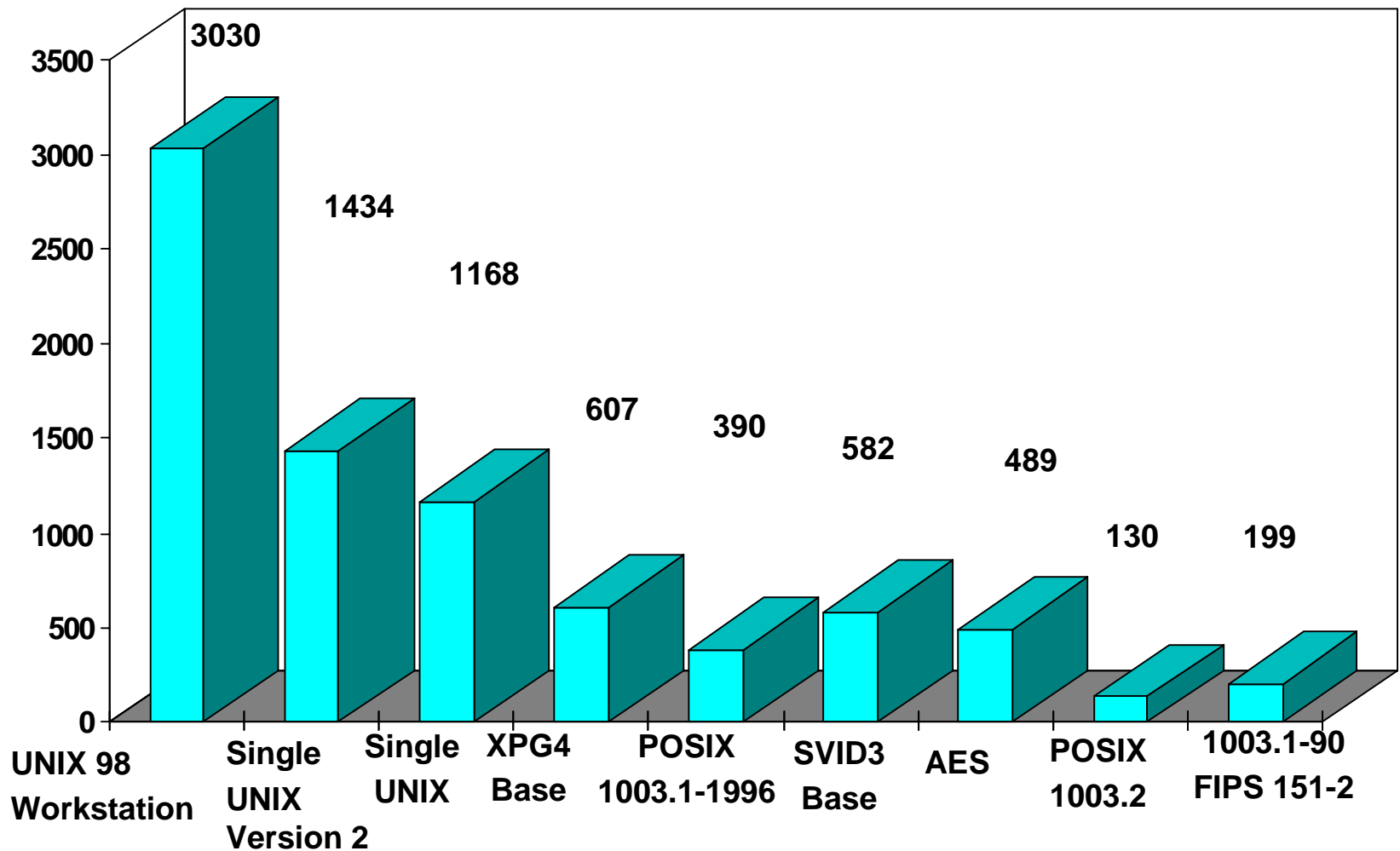
Industry Practice

Evolution

POSIX/UNIX/LSB Synergy

- ❑ The Single UNIX Specification builds on the foundation of POSIX and ISO C
- ❑ The LSB specification builds on the foundation of the Single UNIX Specification Version 2

Interface Counts



Linux Status

- ❑ Recent versions of glibc have been getting closer to full coverage of the *Single UNIX Specification Version 2*
- ❑ Closer still to the *Single UNIX Specification Version 3*
- ❑ The LSB test program shares core common test suites which have been a catalyst to some of this

Linux Status (Cont'd)

- ❑ Some features from Version 2 of the Single UNIX Specification not implemented:
 - STREAMS (isastream, getmsg, putmsg, etc.)
 - XTI (t_alloc, etc)

Linux Status

- Some features partially implemented :
 - Wide characters (complete support in glibc 2.2 and later)
 - Pthreads
 - Large file semantics

Linux Utilities

- ❑ Some differences
- ❑ Mainly due to the standards being aligned with System V historical definitions
 - notably the shell (sh)
 - Some efforts made with *bash* to align with POSIX as a result of the LSB test efforts (see the *VSC-lite* test suite which contains tests for the utilities and shell)

Linux Utilities (Cont'd)

- ❑ In general, where Linux provides an alternative functionality to that in the Single UNIX specification, the standard version has not been provided
 - Compress/uncompress vs gzip/gunzip
 - SCCS utilities vs RCS
- ❑ As per *POSIX.2*; see the *gLSB specification* for details of the differences

The Single UNIX Specification Version 3

- ❑ Developed by the Austin Common Standards Revision Group
- ❑ An open industry initiative to revise the core *POSIX standard* and the *Single UNIX Specification*; standards that lie at the heart of the Linux operating system

The *Austin Group*

- ❑ Electronic participation
- ❑ Participation in the group is free
- ❑ Deliverables:
 - IEEE Std 1003.1-2001 (POSIX.1)
 - The Open Group Base Specifications Issue 6
 - (they are the same document!)
 - Html version freely available on the web (*see later for URL*)

About the *Austin Group*

- Over 600 participants on the mailing list
- Wide industry support:
 - *AT&T, Compaq, Fujitsu, HP, IBM, Lucent, Microsoft, Red Hat, SGI, Siemens, Sun*
 - *DoD, USENIX, Canada Customs and Revenue Agency*
- Participation in the *Austin Group* from the open source community includes:
 - *The Linux Standard Base, NetBSD, FreeBSD, and many others*

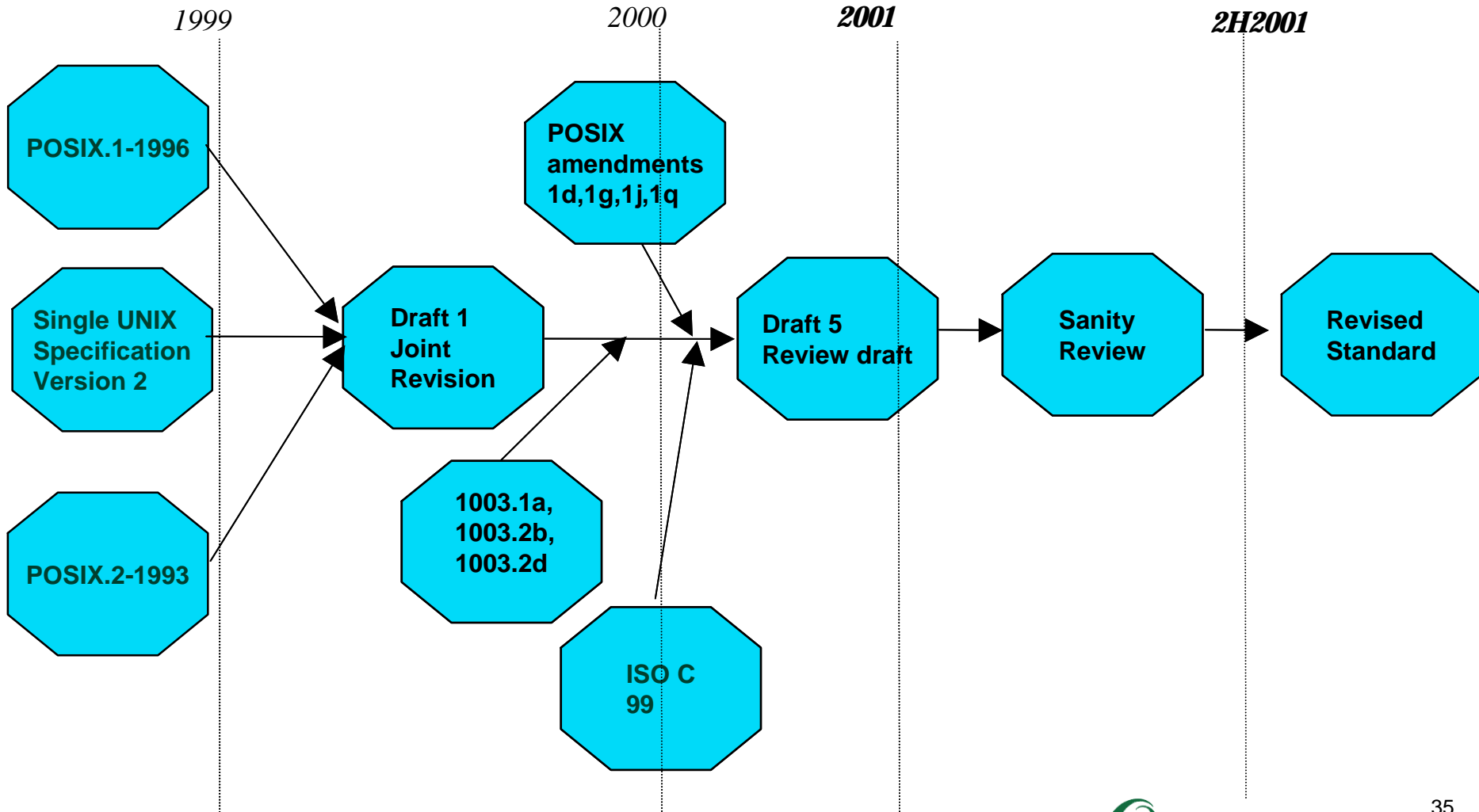
Objectives

- ❑ To target the joint specification at the programmer / user rather than the system implementer
- ❑ Organization based on the Core volumes of the Single UNIX Specification, organized alphabetically, and including Rationale
- ❑ To Produce a standard on schedule

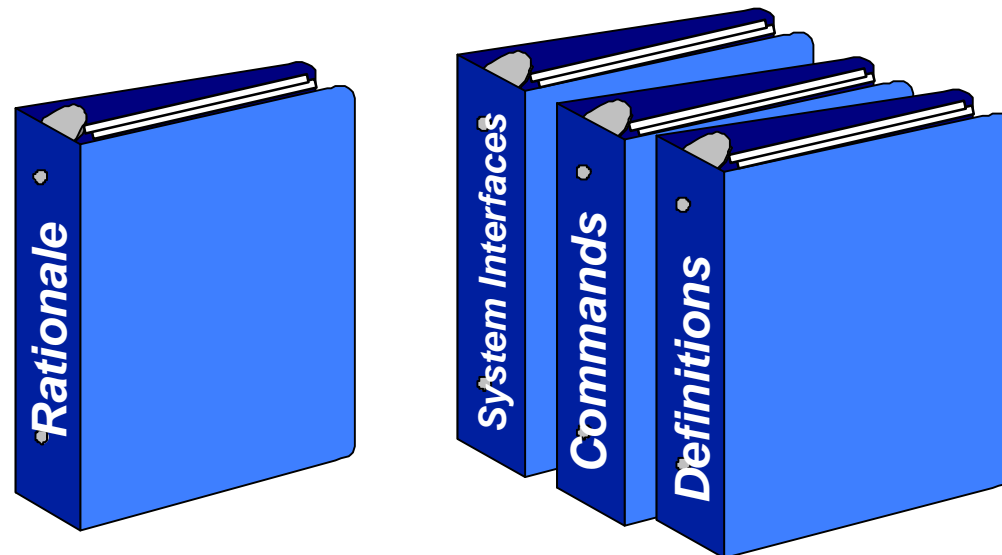
Scope of the revision

- ❑ Production of a single document to be adopted by multiple parties
- ❑ Minimize the number of changes required to implementations of earlier versions of the Base documents for the revision
- ❑ Limit new work items to those related to integration and consistency, resolving any conflicts
- ❑ Alignment with the ISO C 1999 standard

Roadmap to the Single UNIX Specification Version 3



The New Common Specification

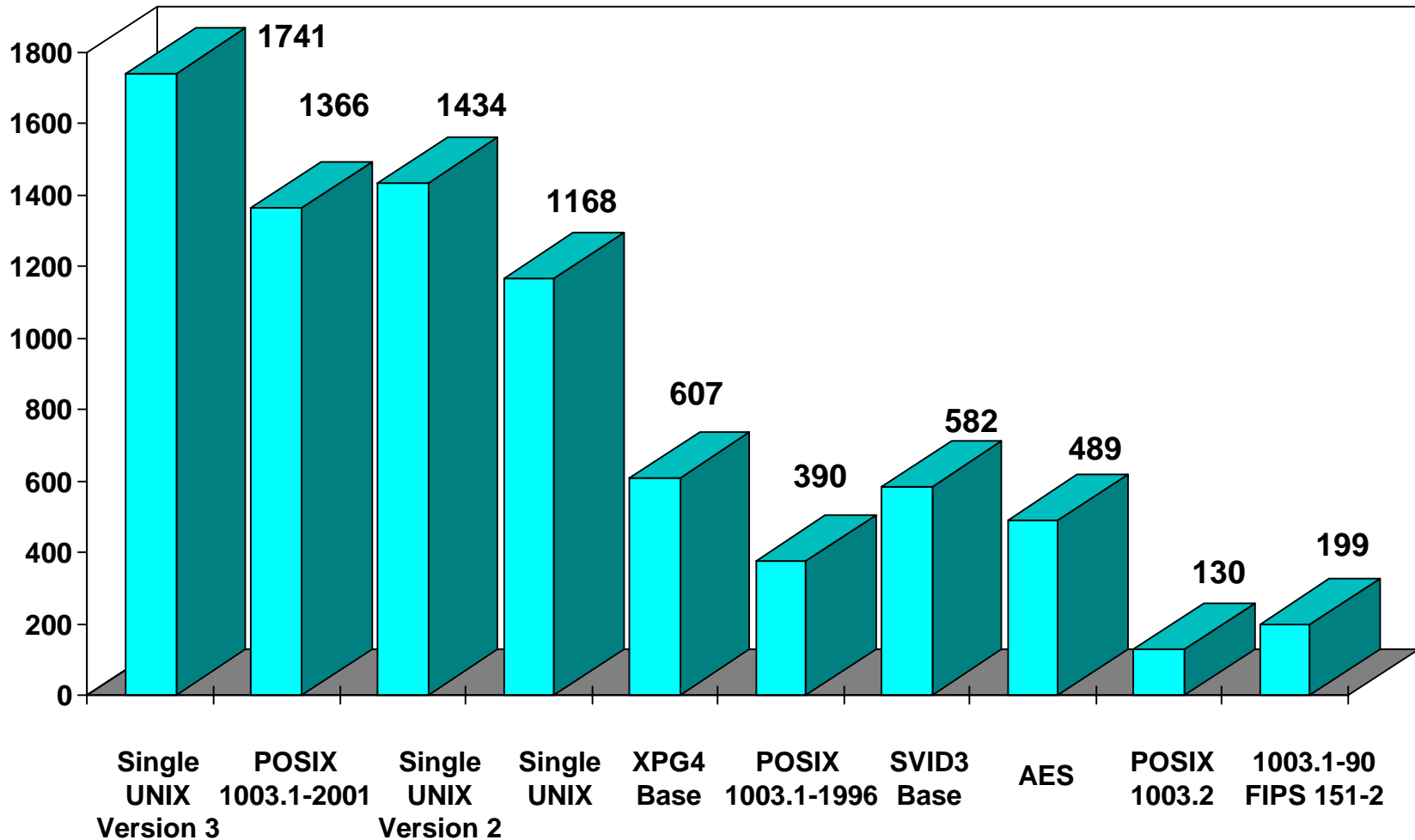


IEEE Std 1003.1-2001,
The Open Group Base
Specifications Issue 6

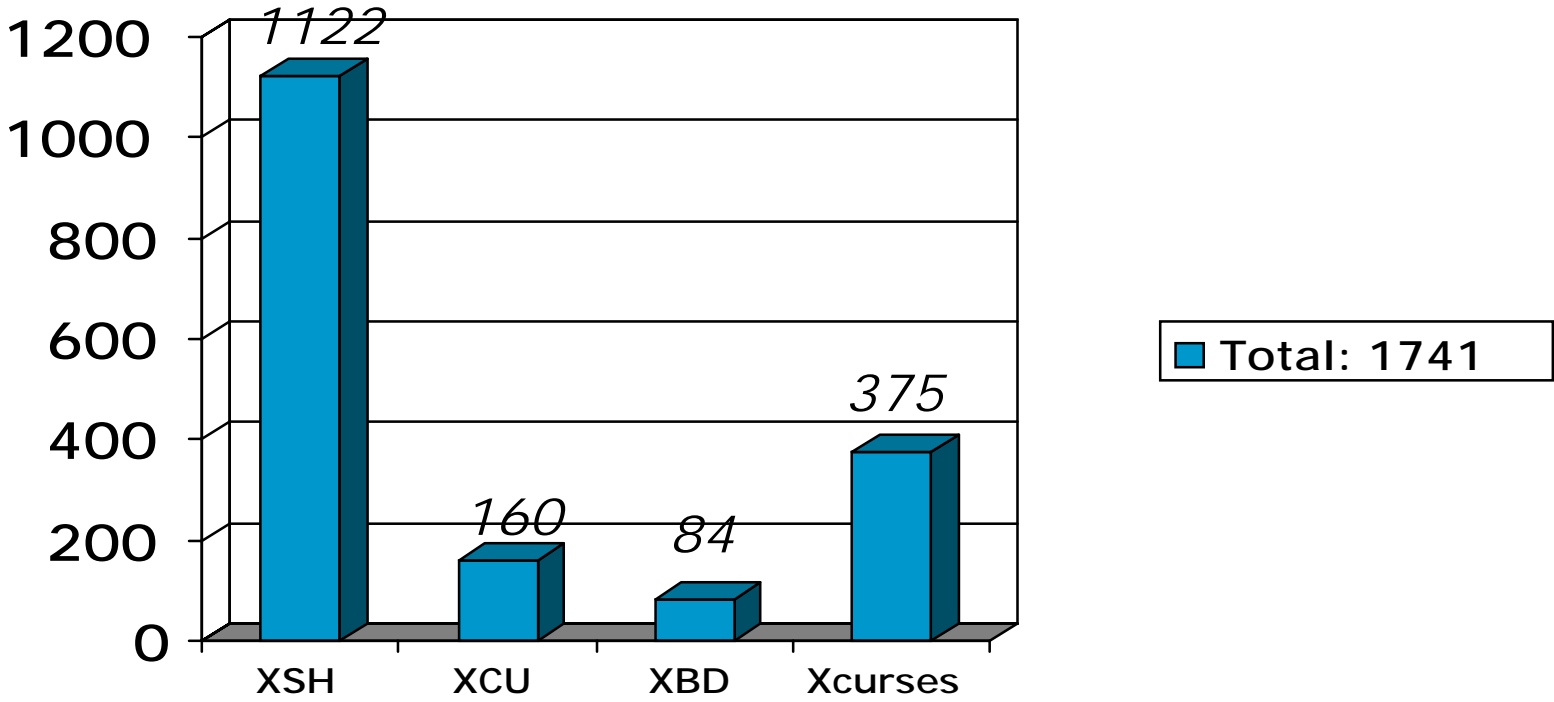
Approvals Status

- ❑ The Open Group September 12th 2001
- ❑ IEEE December 6th 2001
- ❑ Published in 1Q2002, in hardcopy (3700 pages, 9kg!!), electronic and CDROM
- ❑ Expected ISO/IEC Approval in 2002

Portability Functions



Single UNIX Specification Version 3 Interface Count



Formal Standards Alignment

- ❑ IEEE Std 1003.1-2001 (POSIX.1)
 - The Base Specifications Issue 6 is technically identical to POSIX.1, they are one and the same document (*word for word, a single publication*)
- ❑ ISO/IEC 9899:1999, Programming Languages – C (ISO C)

Options

- ❑ Encryption
- ❑ Realtime
- ❑ Realtime Threads
- ❑ Advanced Realtime
- ❑ Advanced Realtime Threads
- ❑ Tracing
- ❑ XSI STREAMS
- ❑ Legacy

Feature Test Macros

- ❑ `_XOPEN_SOURCE=600`
 - Used by applications to request the implementation make available the symbols and prototypes from the Single UNIX Spec V3
 - Subsumes the definition of the POSIX macro `_POSIX_C_SOURCE=200112L`

Compiling an Application

```
c99 -D_XOPEN_SOURCE=600 myapp.c \  
    -o myapp -l c
```

```
c99 -D_XOPEN_SOURCE=600 myrtapp.c \  
    -o myrtapp -l c -l rt
```

```
c99 -D_XOPEN_SOURCE=600 myrthreadsapp.c \  
    -o myrthreadsapp -l rt -l pthread
```

New Features

1003.1d

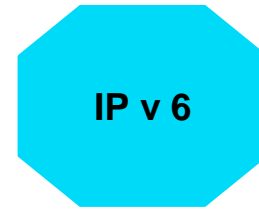
- ❑ Process creation via *posix_spawn()*
- ❑ *Sporadic Server Scheduling policy*
- ❑ *Execution time monitoring*
- ❑ *Time-outs for selected blocking functions*

New Features

- Enhanced threads functions:
 - barriers
 - spin locks
- Additional realtime functionality:
 - monotonic clock

New Features

- 1003.1q , Tracing
- Provides tools to access and manage a stream of event data :
 - Traced process can record a trace event
 - Controller process can manage a trace stream
 - Analyzer process can retrieve traced events



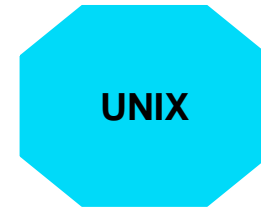
New Features

- IP version 6
 - Overcomes the shortage of address space
 - Designed for better manageability
 - *Security enhancements*
IPSEC
 - *Quality of service*

New Features

□ API Enhancements

- new functions `if_freenameindex()`, `if_indextoname()`, `if_nameindex()` and `if_nametoindex()` have been added to the Sockets Interfaces.
- New functions `inet_ntop()` and `inet_pton()` have been added to the IP Address Resolution Interfaces.



New Features

- ❑ The UNIX feature set is now available as an option within POSIX
- ❑ The new document set forms the core of the Single UNIX Specification Version 3
- ❑ Extended pthreads functions

New Features

- ISO/IEC 9899:1999 ISO C (c99)
 - *Language changes*
 - *New keywords: inline, restrict, _Bool, _Complex, _Imaginary,*
 - *New type: long long*
 - *Complex number and complex maths*
 - *Floating point environment support*
 - *Type generic math*
 - *Other library changes*

Key Changes

- ❑ Legacy and obsolescent features dropped
 - Much of this was to compromise between System V and BSD
- ❑ Job control and certain options in POSIX.1 mandated (FIPS 151-2 alignment)
 - ditto
- ❑ Corrigenda incorporated

Key Changes (Cont'd)

- ❑ XTI dropped
 - Is retained as a separate specification but not required for POSIX or UNIX conformance
- ❑ STREAMS optional
 - An optional feature group for those systems that wish to support the functionality

Technical Corrigendum No. 1

- ❑ Expected to be approved 4Q2002
 - Draft 3 available August 6 2002
- ❑ Intended to be non-controversial
- ❑ It contains no new APIs (functions/utilities), however it can add enumeration symbol and non-function #defines and reserve additional namespaces.
- ❑ Typically TC items fix contradictions between different parts of the standard, add consistency between the standard and overriding standards, or fix security-related problems.

Where to Obtain the Specification?

- ❑ The html version is online at
 - <http://www.UNIX-systems.org/version3/>
- ❑ PDF either electronically or on CDROM can be ordered from <http://www.opengroup.org/pubs/catalog/un.htm>
 - The PDF is free to members of The Open Group
- ❑ Available also on CDROM with *The Authorized Guide to the Single UNIX Specification Version 3*

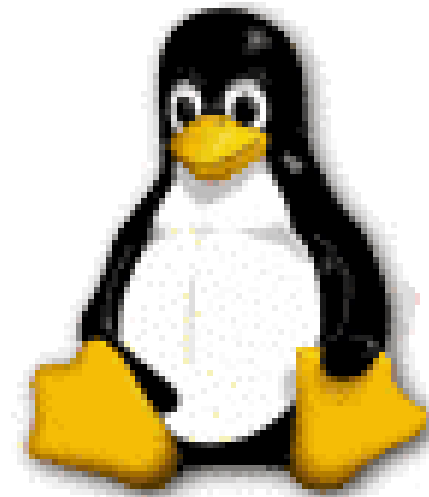
Linux Status

- Latest versions of *glibc* have implemented much of the Austin Group specification
 - C99 Alignment
 - Some differences in the I18N behavior
 - See <http://gcc.gnu.org/c99status.html> for the latest status
 - Concentrated on the XSI mandatory parts of the XSH document
 - Remaining differences due to underlying kernel

Agenda

□ Standards Projects

- POSIX
- The Single UNIX Specification
- **The Linux Standard Base**



LSB Overview

- ❑ Mission & Goals
- ❑ Why is the LSB needed?
- ❑ Organization
- ❑ What LSB does NOT attempt to do & why
- ❑ Architectural Overview - How will it work?
- ❑ Current LSB Roadmap

Mission

- ❑ The LSB develops and promotes a set of standards that increases compatibility among Linux runtime environments and enables software applications to run on any compliant Linux system.
- ❑ In addition, the LSB helps coordinate efforts to recruit software vendors to port and write products for Linux.

Goals

- ❑ To preserve backwards compatibility without locking out future progress
- ❑ To allow runtime environments to still be unique (and provide added value) by only standardizing the base
- ❑ To avoid fragmentation of the base functionality

Why is the LSB Needed?

- ❑ Allows ISVs to:
 - Minimize issues in porting code from another Linux platform
 - Allows a package to perform the same way regardless of the Linux runtime environment or emulation of such

Organization

- Steering Committee Chair - George Kraft IV
 - FSG liaisons - Scott McNeil, Dan Quinlan
 - Technical Sub-Committees
 - Written Spec - Technical Lead - Stuart Anderson
 - Test Suites - Technical Lead - Andrew Josey
 - Sample Implementation - Technical Lead - Ralf Flaxa
 - Build Environment – Technical Lead – Chris Yeoh
 - LSB Futures – Technical Lead – Matt Taggart
 - *Plus Many volunteers*

What is not covered?

- ❑ It does not mean there will only be one "Linux"
 - One port - multiple platform choices
- ❑ It does not specify the kernel level
 - The kernel can be any version that provides conforming interfaces
- ❑ It does not cover languages other than C
 - Other languages will be covered over time
 - Possible to statically link libraries for other language bindings or include interpreter

What is not covered?

- ❑ It does not specify a desktop environment
 - Desktop integration is a work in progress
 - Use desktop independent toolkits (Qt, Gtk)
- ❑ It does not fully cover system administration
 - Settled areas are specified - Others will be added

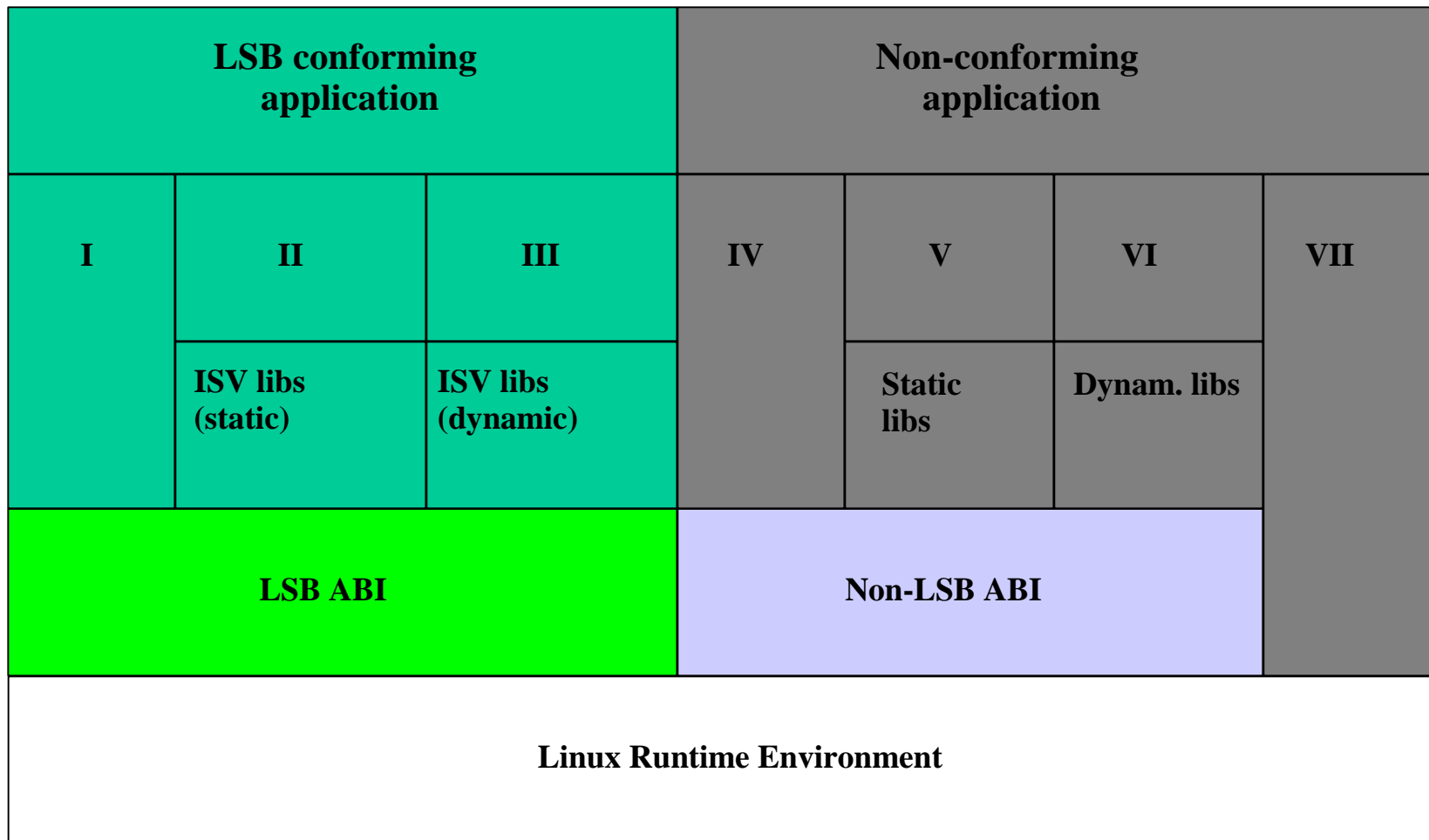
LSB Specification

- A *generic, processor-independent* specification
 - The Generic LSB Specification known as the *gLSB*
- One or more processor *architecture-specific* supplemental specifications that build upon the *gLSB* to provide a complete binary specification
 - Known as *archLSB specifications*
 - IA32, IA64, PPC32

LSB Specification

- ❑ Object file formats
- ❑ Dynamic linking
- ❑ Base libraries
- ❑ Utility libraries
- ❑ Graphics Libraries
- ❑ Packaging
- ❑ Commands and Utilities
- ❑ Standard Shell
- ❑ Users and Groups
- ❑ Filesystem Hierarchy
- ❑ System Initialization

Architectural Overview



Standards Alignment

□ Built on industry standards

- IEEE Std POSIX 1003.1-1996 (POSIX.1)
- IEEE Std POSIX 1003.2-1992 (POSIX.2)
- The Open Group Single UNIX Specification Version 2 (*aka* UNIX 98)
- AT&T (Caldera/SCO) SVID, Issue 3 1989-2001 (SVID.3)
- See www.linuxbase.org/spec/gLSB/gLSB/rstandards.html

36 Specifications referenced at this time

LSB Milestones

- ❑ *gLSB 1.0* Specification published June 2001
- ❑ Pilot Beta test program launched August 2001
- ❑ *gLSB 1.1* Specification published December 31 2001
- ❑ *archLSB* for IA32 published December 31 2001
- ❑ *archLSB* for IA64 expected 2002
- ❑ *gLSB 1.2* Specification published July 2002
- ❑ *archLSB* for IA32, PPC32, published July 2002
- ❑ LSB Certification launched July 2002

gLSB 1.2 vs 1.1/1.0

- ❑ Upward compatible
- ❑ Some exceptions
 - Bug fixes
 - Additions of functions identified as needed by applications
 - Improved data definitions
 - Changes needed to support *archLSBs* (*note these were previously called psLSBs*)

Current Status (Cont'd)

- ❑ Source Test Suites released December 2001
 - LSB-FHS
 - VSX-PCTS
 - LSB-OS
 - LSB-USRGROUPS
- ❑ Binary Test Suite release December 31st 2001
- ❑ Certification Pilot January-June 2002
- ❑ Certification Program launch July 2002

The Free Standards Group LSB Certification Program



- A voluntary program.
- Open to any product meeting the conformance requirements
- Not restricted to Linux-based systems or Linux-based applications.
- Once a supplier has achieved certification for a product, they are permitted to use the LSB trademark in connection with that product.

Understanding LSB Certification

- ❑ To become familiar with the LSB Certification program, you should read the following program documents:
 - The *LSB Certification Policy*
 - The *LSB Product Standards*
- ❑ <http://www.freestandards.org/certification/>

The LSB Certification Policy

- ❑ The foundation of the program.
- ❑ Defines the types of products that can be certified.
- ❑ Defines what it means to be certified.
- ❑ Defines what is required to certify a product .
- ❑ Defines how to make sure that a product remains certified.

LSB Product Standards

- The detailed conformance requirements against which a product can be certified.
 - A mapping between certification, the LSB specifications and the test suites needed to demonstrate conformance.
 - An *LSB Product Standard* for each type of product that can be certified.
- Current Product Standards :
 - the *LSB Application for IA32 Version 1.2*
 - the *LSB Runtime Environment for IA32 Version 1.2*

LSB Certification

- Product Types that can be (currently) certified
 - LSB Runtime Environment Certification
 - For platforms providing services that conform to the LSB specifications.
 - LSB Application Certification
 - For applications conforming to the LSB specifications.
 - LSB applications are the consumers of the services provided by LSB Runtime Environments.

LSB Application Certification: Test Campaign

- ❑ The application must execute correctly on the two selected LSB runtime environments and the LSB Sample Implementation
- ❑ Completion of an LSB Application Conformance Statement
- ❑ Test Journal output from the LSB Application Checker tool (*lsbappchk*)
- ❑ For the test journal, any FAIL, UNRESOLVED, UNREPORTED or UNINITIATED results need to be resolved

LSB Runtime Environment Certification: Test Campaign

- ❑ Completion of the LSB Runtime Environment Conformance Statement
- ❑ Test Journal output from the LSB runtime environment test suite
- ❑ Test Journal output from the LSB Library Checker tool (*lsblibcchk*)
- ❑ For the test journal output, any FAIL, UNRESOLVED, UNREPORTED or UNINITIATED results need to be resolved
- ❑ For the test journal output, any FIP results need to be manually resolved
- ❑ A list of at least two binary applications from the LSB Application battery that the applicant warrants works correctly

First LSB Certified Systems



- *Mandrake Linux ProSuite 8.2 + first update CD*
- *Red Hat Linux 7.3 with glibc 2.2.5-39+kernel 2.4.18-10 or later*
- *SuSE Linux 8.0 Professional + aaa_base and Kernel Update*

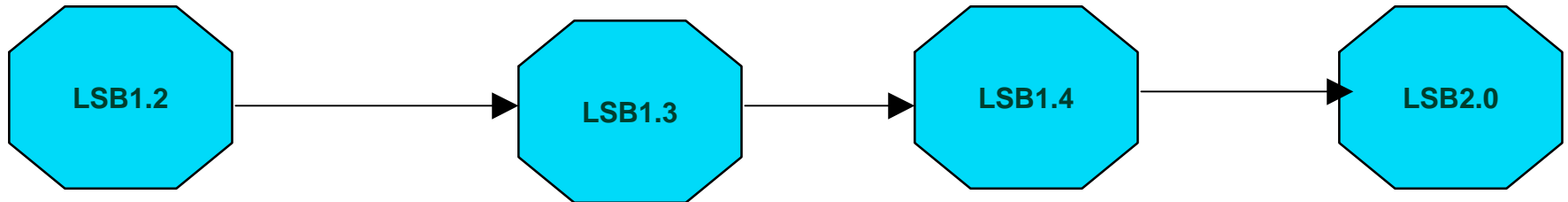
LSB Roadmap

August 2002

Jan 2003

Aug 2003

Jan 2004



- C++ spec
- PPC32 test/cert/si
- IA64 spec/test/cert/si
- PPC64 & zSeries spec/test/si
- Li18nux integration 1
- Misc. technical, editorial changes to spec/test/si

- PPC64 & zSeries cert
- Test Suite Expansion
- C++ cert
- Li18nux integration 2
- LSB Development Environment spec/tests
- Misc. technical and editorial changes to spec/tests/si

- Modularise and reorganize the LSB
- Uplift to SUSv3
- Add new library ABIs to the specification (UI & graphics)
- LSB Development Environment certification
- Misc. technical and editorial changes to spec, tests, & SI

Proposed Schedule

LSB Schedule	LSB v1.3 (4Q02)	LSB v1.4 (2Q03)	LSB v2.0 (4Q03)
Development Ends	10/18/02	05/23/03	10/17/03
Source Freeze	10/25/02	05/30/03	10/24/03
Public Review	10/28 - 11/08	06/02 - 06/13	10/27 - 11/07
SpecAuth Approval	11/22/02	06/20/03	11/21/03
FSG Approval	12/13/02	06/27/03	12/12/03

LSB Resources

- ❑ LSB main site – www.linuxbase.org
- ❑ Free Standards Group – www.freestandards.org/
- ❑ The LSB Specification - www.linuxbase.org/spec/
- ❑ LSB Test
 - www.linuxbase.org/test/
- ❑ LSB Sample Implementation
 - www.linuxbase.org/impl/
- ❑ LSB Futures
 - www.linuxbase.org/futures
- ❑ LSB Certification
 - www.opengroup.org/lsb/cert/

Summary

- ❑ Standards activities
 - alive and ongoing
 - free to participate in
- ❑ Significant cross-fertilization of projects
- ❑ Significant opportunities to grow the market and prevent fragmentation

Further Information

- *The Linux Standard Base*
 - *www.linuxbase.org*
- *The Austin Group*
 - *www.opengroup.org/austin*
- *The Single UNIX Specification*
 - *www.UNIX-systems.org*
- *These slides*
 - *www.UNIX-systems.org/slides.html*

How You Can Help?

- ❑ To participate in the *Austin Group*, join the mailing list by visiting the web site at:
 - www.opengroup.org/austin
- ❑ To participate in the Linux Standard Base visit:
 - www.linuxbase.org
- ❑ To join the Free Standards Group, visit:
 - www.freestandards.org



Standards: Bridging the Generation Gap between Linux® and UNIX® Systems

Andrew Josey
Director of Certification
The Open Group
Email: a.josey@opengroup.org

UNIX is a registered trademark of The Open Group
Linux is a registered trademark of Linus Torvalds

15-Aug-2002

THE *Open* GROUP