NetBSD goes POSIX:2008 and C1X

Takehiko NOZAKI
<tnozaki@NetBSD.org>
What is the POSIX?
What is the POSIX?
- Portable
What is the POSIX?

- Portable
- Operating System
What is the POSIX?

- Portable
- Operating System
- Interface
What is the POSIX?

- Portable
- Operating System
- Interface
- [for UNIX]
What does Interface means?
What does Interface mean?

- User Interface (UI)
What does **Interface** mean?

- **User Interface (UI)**
  - **Command-line User Interface (CUI)**
What does Interface means?

- User Interface (UI)
  - Command-line User Interface (CUI)
  - Graphical User Interface (GUI)
What does Interface means?

- User Interface (UI)
  - Command-line User Interface (CUI)
  - Graphical User Interface (GUI)
- Application Programming Interface (API)
What does **Interface** mean?

- **User Interface (UI)**
  - Command-line User Interface (CUI)
  - Graphical User Interface (GUI)
- **Application Programming Interface (API)**
- **Application Binary Interface (ABI)**
POSIX target is...
POSIX target is...

- User Interface (UI)
  - Command-line User Interface (CUI)
  - Graphical User Interface (GUI)
- Application Programming Interface (API)
- Application Binary Interface (ABI)
Who revised the POSIX?
Who revised the POSIX?

- The Austin Common Standard Revision Group (The Austin Group)
Who is the member of the Austin Group?
Who is the member of the Austin Group?

- The Open Group (TOG)
Who is the member of the Austin Group?

- The Open Group (TOG)
- IEEE PASC
Who is the member of the Austin Group?

- The Open Group (TOG)
- IEEE PASC
- ISO/IEC JTC1/SC22/PAG
The Open Group (TOG)
The Open Group (TOG)

- A vendor/technology-neutral consortium
The Open Group (TOG)

- A vendor/technology-neutral consortium
- The certifying body for the UNIX trademark
IEEE PASC
IEEE PASC

- Institute of Electrical and Electronics Engineers (米国電気電子学会)
IEEE PASC

- Portable Application Standards Committee
IEEE PASC

ISO/IEC JTC1/SC22/PAG

- International Organization for Standardization (国際標準化機構)
ISO/IEC JTC1/SC22/PAG

- International Organization for Standardization
- International Electrotechnical Commission (国際電気標準会議)
ISO/IEC JTC1/SC22/PAG

- International Organization for Standardization
- International Electrotechnical Commission
- Joint Technical Committee 1 (第一合同技術委員会)
ISO/IEC JTC1 /SC22/PAG

- Sub Committee 22
  - Programming languages and, Operating System
ISO/IEC JTC1 /SC22 /PAG

- Sub Committee 22
  - Programming languages and, Operating System
- POSIX Advisory Group
ISO/IEC JTC1 /SC22/PAG

- Sub Committee 22
  - Programming languages and, Operating System
- POSIX Advisory Group
  - Former WG15 (Workgroup–15)
ISO/IEC JTC1/SC22/PAG

You can freely subscribe the Austin Group Mailing Lists!
You can freely subscribe the Austin Group Mailing Lists!

- austin-group-l
  - general discussion
- austin-regexp-l
  - regular expression subgroup
- austin-group-futures-l
  - futures discussion
- austin-group-offtopic-l
  - offtopic discussion
What’s new POSIX:2008
What’s new POSIX:2008

- The Open Group Base Specifications, Issue 7
What’s new POSIX:2008

- The Open Group Base Specifications, Issue 7
  - Former Single UNIX Specifications
What’s new POSIX:2008

- The Open Group Base Specifications, Issue 7
  - The Base Definitions (XBD)
  - The System Interfaces and Headers (XSH)
  - The Commands and Utilities (XCU)
What’s new POSIX:2008

- The Open Group Base Specifications, Issue 7
  - The Base Definitions (XBD)
  - The System Interfaces and Headers (XSH)
  - The Commands and Utilities (XCU)
- Extended API Set, Part 1 – 4
Extended API Set, Part 1
Extended API Set, Part 1

- New APIs derived from Linux Standard Base (LSB)
  - fmemopen, open_w?memstream
  - mbsnrtowcs, wcsnrtombs
  - etc.
What is Linux Standard Base?
What is Linux Standard Base?

- Too many distribution
- Too many differences
- Too difficult to maintain compatibility
What is Linux Standard Base?

- Keep compatibility with several Linux distributions
  - POSIX (CUI and API) compatibility + $\alpha$
  - ABI
  - Filesystem hierarchy Standard (FHS)
  - Runlevel
  - X Window System extensions
  - Printing system
Who made Linux Standard Base?

- History
  - LI18NUX/OpenI18N → LINUX2000
  - Linux Standard Base
  - OpenI18N + LSB = Free Standard Group (FSG)
  - Open Source Development Labs (OSDL)
  - FSG + OSDL = Linux Foundation
  - ISO/IEC JTC1/SC22 Linux Studying Group (LSG) → Extended API Set, Part 1
Implementation status of Extended API Set, Part 1

- Linux
- FreeBSD
- NetBSD – patch provided (by me)
Extended API Set, Part 2
Extended API Set, Part 2

- New system calls with relative pathname
  - fdopendir, fexecve
  - \{open, link, readlink, rename, symlink, unlink\} at
  - etc.
Extended API Set, Part2

- Background
  - TOCTTOU (time of check to time of use)
    races and vulnerabilities (e.g. symlink attack)
  - chdir(2) is not MT–safe
Implementation status of Extended API Set, Part2

- Solaris 9
- Linux kernel 2.6.16
- FreeBSD 8
- NetBSD – patch provided(by ad@)
Extended API Set, Part 3
Extended API Set, Part 3

- New pthread API for robust mutexes
  - pthread_mutex_consistent
  - pthread_mutexattr_{get,set}robust
  - EOWNERDEAD
  - etc.
Extended API Set, Part3

- Background
  - Deadlock possibility
Implementation status of Extended API Set, Part 3

- Solaris 9 later
- Linux/glibc2
Extended API Set, Part 4
Extended API Set, Part4

- New locale APIs for multi-locale
  - locale_t, newlocale, freelocale
  - isw{alnum,alpha,blank,...}_l
  - etc.
Extended API Set, Part 4

- **Background**
  - `setlocale(3)` is not MT-safe
  - C++ `std::locale` requires multi-locale
  - Thread-aware Locale Model (by Urlich Drepper@Redhat)
Problems

- Bad design: too complicated, it's hard to manage the set of locale_t, mbstate_t, wchar_t for programmers
- Require Thread Local Storage support
Implementation status of Extended API Set, Part4

- glibc2
- Microsoft Visual C/C++ 2005
- MacOS X
- NetBSD – patch provided (by me)
What is C1X?
What is C1X?

- The second revision of ISO/IEC 9899 “Programming Language C”
- And the first revision in the 21st century
What is C1X?

- The second revision of ISO/IEC 9899 “Programming Language C”
- And the first revision in the 21st century
  - 1990 – C90 (C89)
  - 1995 – C90 AMD1 (C95) [Amendment-1]
  - 201X – C1X [Third Edition]
What's new in C1X?

- C – The C1X Charter (N1250)
- Latest Draft of ISO/IEC 9899:201X (N1425)
- Including “C Secure Coding Guidelines”
DTR 19769: New character type in C (N1040)

- UTF{16,32} hardwired character type
  - char{16,32}_t, u and U prefix
  - mbstrtoc{1632}, c{1632}rtomb
  - etc.
DTR 19769: New character type in C (N1040)

- **Background**
  - C string function (e.g. strcmp, mbrtowc) can't treat UTF{16,32} String (L'\0' != '\0')
  - C99: __STDC_ISO_10646__ and Universal Character Name (e.g. L"\u3042")
  - C++0x: Native Unicode Support
  - Unicode Raw String Literal (N1333)
DTR 19769: New character type in C (N1040)

- Problem
  - `mbrtoc{16,32} / c{16,32}rtomb` is just lesser `iconv(3)`
DTR 19769: New character type in C (N1040)

- Implementation status
  - glibc2 and gcc4 – WIP
  - NetBSD and pcc – NOT YET
TR24731: Extension to the C library
TR24731: Extension to the C library

- Part 1: Bounds-checking interfaces (N1225)
TR24731: Extension to the C library

- Part 1: Bounds-checking interfaces (N1225)
  - strncpy_s, strcat_s, sprintf_s
  - fopen_s, getenv_s
  - etc.
TR24731: Extension to the C library

- Part2: Dynamic allocation functions (N1337)
TR24731: Extension to the C library

- Part2: Dynamic allocation functions (N1337)
  - fmemopen, open_w?memstream
  - {str, wcs}n?dup
  - Etc.
  - → see POSIX:2008 Extended API Set, Part1
Background

- Smashing the Stack for Fun and Profit (by Aleph One)
- No more stack buffer overflow
TR24731-1: Bounds-checking interfaces (N1225)

- Problems
  - gcc already have Stack Smashing Protector (a.k.a Propolice) and libssp's checking function, str{cat,cpy}_chk
  - *BSD already have strl{cat, cpy} and widely used by many applications
  - And more, NetBSD have efun(3)'s estrl{cat, cpy} in util.h
TR24731-1: Bouns-checking interfaces (N1225)

- Implementation status
  - Microsoft Visual C/C++ 2005
    - (_CRT_SECURE_CPP_OVERLOAD_STANDARD_NAMES)
      - OpenWatcom
      - NetBSD – patch provided (by me)
TR24731–2: Dynamic allocation functions (N1337)

- **Background**
  - `strl{cat,cpy}` flame war between Theo de Raadt with Urlich Drepper
  - Drepper wrote “Defensive Programming for RHEL”
Others...

- TR 18037: Embedded C (N1169)
- TR 24732: Decimal floating point (N1312)
- TR 24747: Mathematical special functions
More C1X Proposal

- On The Removal of gets() (N1420)
- Blocks and GC Proposal (N1451)
- Atomic Proposal (N1473)
- Anonymous Member–Structures and –Union (N1406)
- Toward support for attributes in C (N1403)
- Type generic macro (N1405)
More C1X Proposal

- #macro Proposal (N1410)
- Adding Alignment support for C (N1397)
- Namespace Reservation for the C Standard (N1345)
- Thread Local Storage (N1329)
- Adding Threads to the C Standard Library (N1325)