System Programming Overview

Advanced Embedded Linux Development

with **Dan Walkes**

University of Colorado **Boulder**

Learning objectives:

Introduce System Programming

Understand APIs and ABIs

Introduce POSIX

System Programming Overview

- What is System Software?
  - Interfacing with the kernel and C library.
  - GUI, compiler, debugger, web server, database.
- Differences relative to “Application Software”:
  - Doesn’t use higher level libraries.
  - Less OS/hardware details abstracted.

Cornerstones of System Programming

- System Calls (syscalls)
- C Library (libc)
- C Compiler/linker (toolchain)

Cornerstones of System Programming

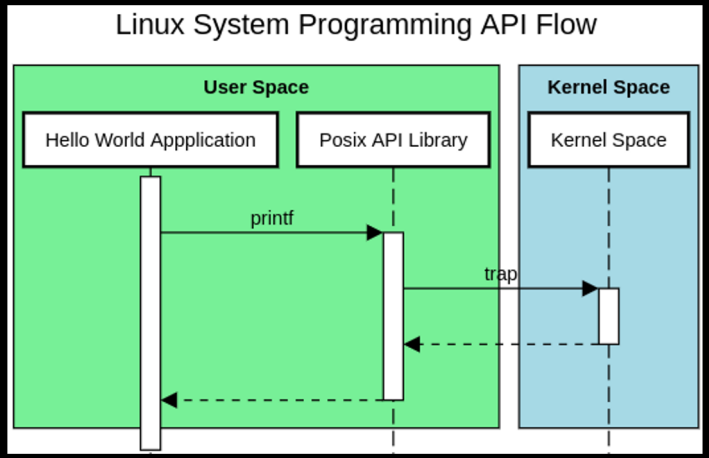
- System Calls (syscalls)
  - Kernel function invocations from user space (read() write()) Roughly 300 total.
  - Shared subset of ~90% implemented by all architectures on Linux.

Cornerstones of System Programming

- C Library (libc)
  - GNU libc (glibc) (gee-lib-see) Wrappers for system calls, threading support and applications
- C Compiler/linker (toolchain)
  - GCC (GNU C compiler)

Invoking System Calls

- Not possible to link your user space application with the Linux kernel.
  - Why not?
    - Not allowed for security/reliability reasons
- How do you invoke system calls?
  - Use a “trap” - typically a software interrupt



Linux System Programming, Robert Love Chapter 1

https://medium.com/the-tech-house/dual-mode-multimode-operation-bcdc93077f7a

API

- Application Programming Interface.
- API is a definition.
- Software which provides an API is the implementation.
- Ensures **source** compatibility.
  - Source can be compiled for different platforms, works in a specific way.

API

- Source code remains portable across different hardware platforms/revisions (ideally)
- Example: C library functions like printf(), strcpy(), etc.
- Exceptions: syscall ~10% architecture differences mentioned in previous slide

ABI

- Application Binary Interface
- Calling conventions, byte ordering, register use
- Ensures **binary** compatibility
- Defined/implemented by Kernel and toolchain
- Defines application interaction with itself, libraries, the kernel.

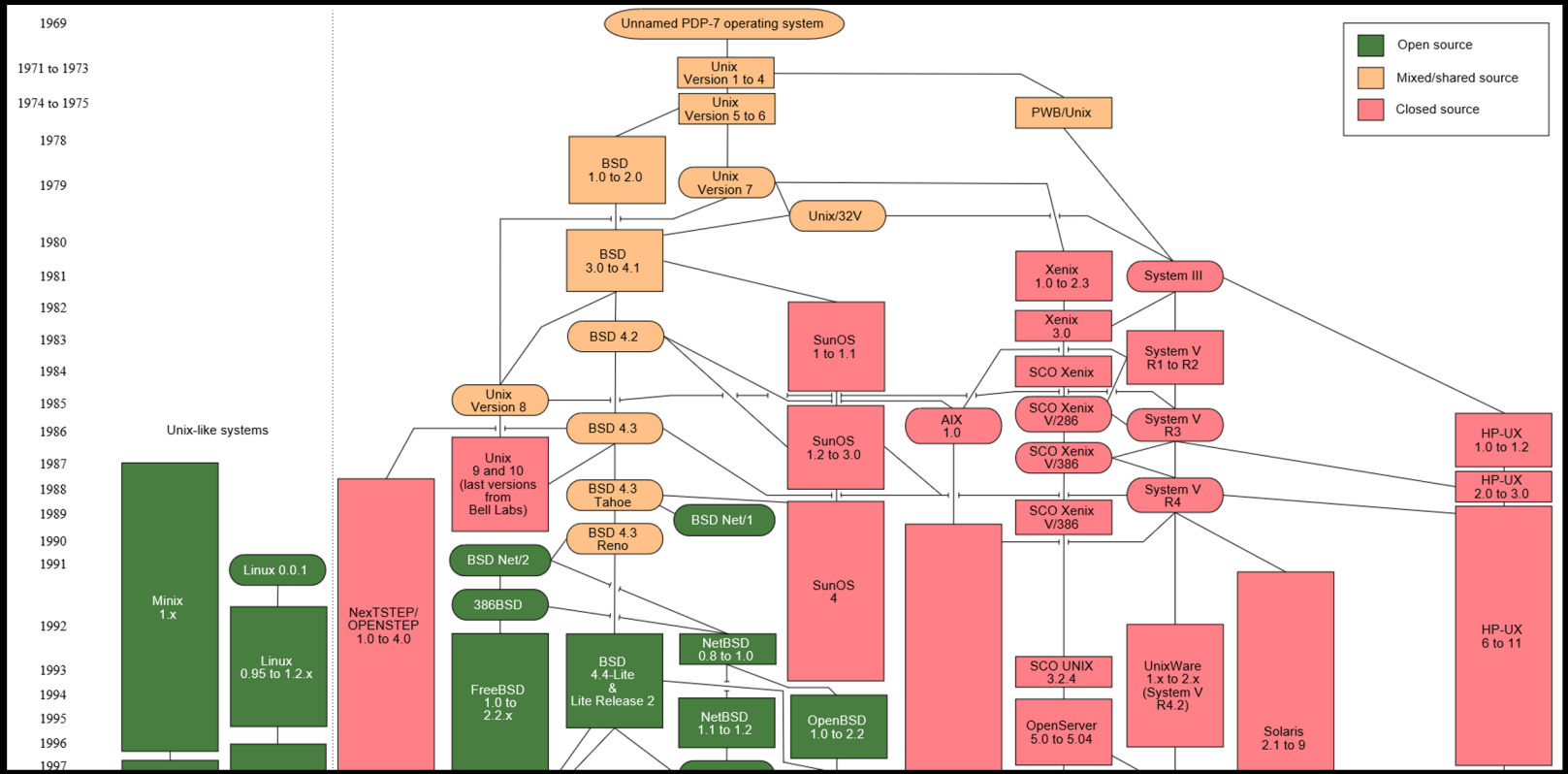
ABI

- Binary generated after compilation/link (by toolchain).
- Byte code specific hardware types, software/compiler/library revisions.
  Requires match:
  - hardware
  - software libraries (especially glibc)
  - compiler

POSIX

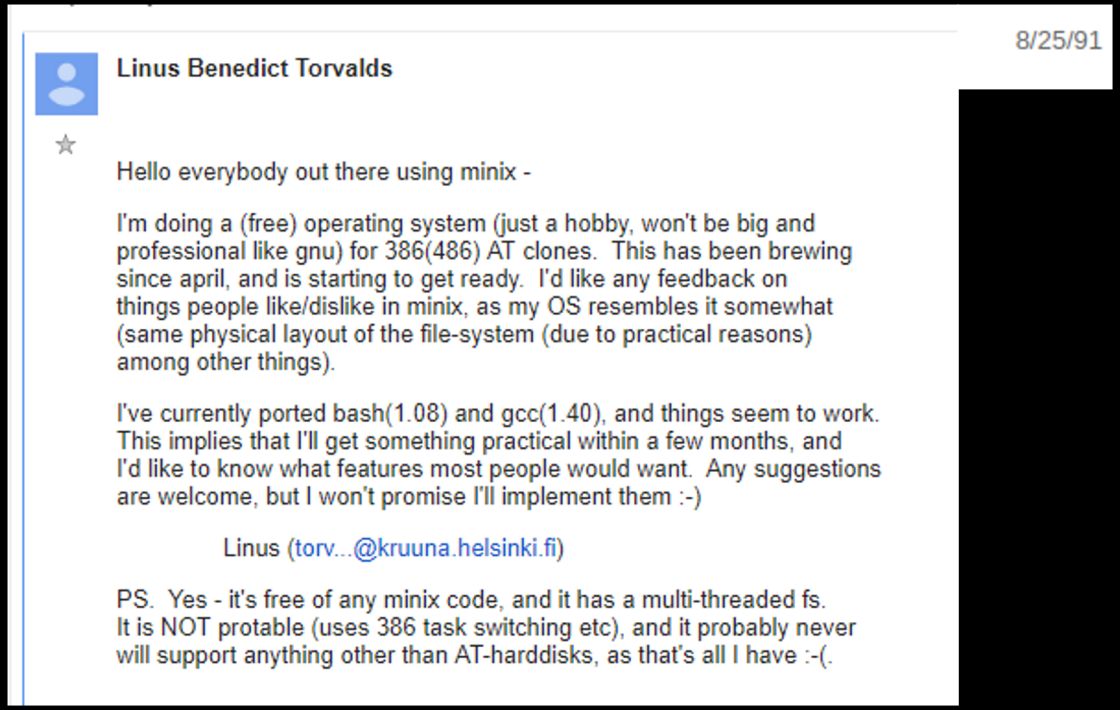
- C APIs for Unix-like operating systems
- Stands for Portable Operating System Interface
  - Pronounced “pahz-icks”
- Started by IEEE in the late 1980s, as a way to coalesce the “Unix Wars”

UNIX Wars



By Eraserhead1, Infinity0, Sav_vas - Levenez Unix History Diagram, Information on the history of IBM's AIX on ibm.com, CC BY-SA 3.0,

Enter Linux



8/25/91

Linus Benedict Torvalds

Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

Linus (torv...@kruuna.helsinki.fi)

PS. Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT protable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-(.

https://groups.google.com/forum/#!msg/comp.os.minix/dlNtH7RRrGA/SwRavCzVE7gJ

POSIX

- Open Software Foundation created the Single Unix Specification (SUS)
  - Specification was free
  - Eventually incorporated the POSIX standard.

POSIX

- POSIX defines C code API for many concepts we'll cover this semester
  - File and directory operations
  - Clocks and timers
  - Semaphores
  - Shared memory
  - Threads

POSIX

- Official list of headers: http://pubs.opengroup.org/onlinepubs/9699919799/idx/head.html
  - Some familiar ones:
  - <stdio.h>
  - <stdint.h>