Lecture 1: The UNIX Time-Sharing System

Mothy Roscoe

August 30, 2005

A warm-up paper; goals:

- Worked example of what we're going to do.
- The *form* of a typical systems paper.
- Defamiliarization – what's *unusual* about Unix.

Features:

- Time-sharing system
- Hierarchical file-system as unifying abstraction, device-independent I/O
- Shell-based interface
- Unstructured files (byte vectors)

History (version 3 UNIX):

- Ran on PDP-11s
- $<$ 50kB kernel, 2 man-years to write
- Written in C – to some extent C written for UNIX
- Reaction to Multics complexity, hence *very* simple.

File system:

- Ordinary files (byte vectors, no records)
- Directories (marked, protected ordinary files)
- "Special" files (for I/O)
- Principle unifying abstractions:
  - Namespace (names as opposed to files)
  - Files, and file descriptors (open files)

Namespace:

- Single root, hierarchical path names
- Multiple names for files (but not directories!)
- Directories are distinguished normal files
- Links to parent directories
- Current working directory (relative names)
- *Mounting* file system over ordinary file

Files:

- Protection: user/world, RWX
- set-user-id bit, super-user is special case of user
- Special files: indirect to devices and pseudo-devices

Uniform I/O model:

- No structure, simply byte streams
- open, close, read, write, seek, etc.

File system implementation:

- i-node table
  - simple, allows efficient recovery
  - hard to extend functionality (e.g. accounting)
- path-name scanning, and the mount table
- the buffer cache, and write-behind

Processes:

- just 3 segments (text, data, stack), one address space
- Text sharing
- Process creation primitives: fork and exec
- Synchronization: wait and exit
- Pipes (unified with files, again)
- Asynchrony: traps and signals

The Shell:

- cmd arg1 . . . argn
- I/O redirection
- filters, pipes, multitasking (backgrounding)
- Just a program, but pervades rest of system . . .
- . . . except the kernel
- Perhaps key systems design issue is the shell/kernel separation