Linux — Operating Systems and Design — Freedom

Frank C Langbein
Cardiff University, `https://langbein.org/`, `frank@langbein.org`

Qyber\black, `https://qyber.black/`

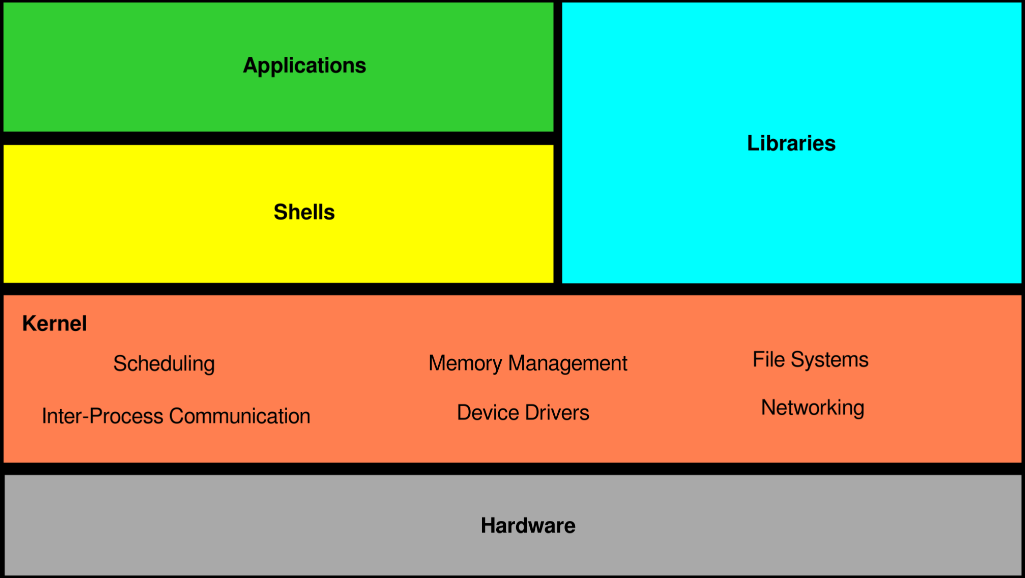
November 2023

OSs – Confusing Concepts

- Manage hardware and software of computers to *simplify operation*
- *Evolving* concepts
  - 1940s... vacuum tubes, machine language
  - 1950s... transistors, assembly language
  - 1960s... integrated circuits, high-level languages
  - 1970s... microprocessors, PCs, GUIs
  - 1990s... ULSI, networking, AI (?) — still working on it
- Not about distributions, user interfaces... poor marketing
- UN*X — *multi-user, multi-task, not real-time*
  - 1969 — AT&T Bell Labs (Ken Thompson, Brian Kernighan, Dennis Ritchie)
  - Clones: AIX, HPUX, SunOS, Xenix, ...
  - Standards: BSD, System V, POSIX
  - 1983... free computer programs can be freely modified and shared
    - 1985... GNU is Not UNIX / Free Software Foundation (Richard Stallman)
    - 1987... Minix (Andy Tanenbaum)
    - 1991... Linux (Linus Torvalds)

    - 1990... GNU MACH / The Hurd (Thomas / Michael Bushnell)

Rough OS Structure

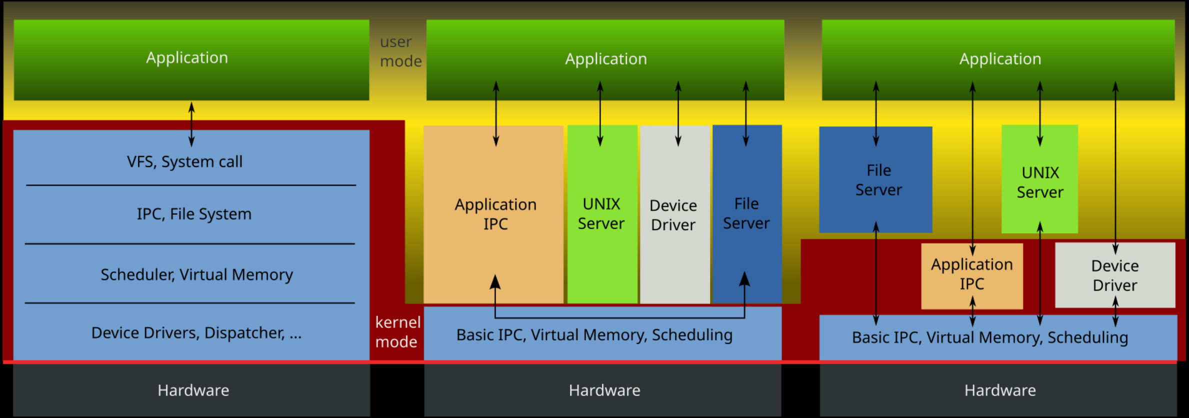


Design Factors

- Freedom
- Performance
- Efficiency
- Security
- Reliability
- Usability
- Control

  .
  .
  .
- Modularity, Simplicity, Elegance, . . . ???
  - Often a waste of time, yielding poorer practical results
  - It's not that simple: *humans, model of the world, wishful thinking* vs *the machine, physical reality*

Monolithic, Micro, Hybrid Kernels



[adapted from Wikipedia, Hybrid Kernel]

Linux Terminal

- *Shell*: interactive command line interface
  - `sh`, *`bash`*, `tcsh`, `zsh`, `dash`, `...`
  - Auto-complete via TAB and basic editing via cursor keys
  - Also see terminal multiplexers: `screen`, `tmux`
  - Either text *console* or *GUI* ([CTRL-]ALT-Fn switches) or *remote* (ssh; not telnet)
- Commands *act on files*
  - *Single hierarchy* of files and directories
  - Spaces are evil
  - Files are *IO interfaces* to resources, not just (sequential) data dumps
  - Read/write/ioctl — *everything is a file*
  - Current / working directory
- Processes, users, groups, permissions
  - Shell is a *process* that spawns other processes (which have one/multiple threads)
  - Each process has a *user* and a *group*, inherited from the parent (can change if it has permission)
  - Users and groups have *file permissions* (simple UNIX permissions; but also ACLs)

File System

File System Hiararchy Standard

```
lrwxrwxrwx     1  root  root     7  Jan 5 2021     bin -> usr/bin
drwxr-xr-x     1  root  root   500  Nov 10 10:12   boot
drwxr-xr-x    21  root  root  4500  Nov 4 11:28    dev
drwxr-xr-x     1  root  root  5942  Nov 13 11:56   etc
drwxr-xr-x     1  root  root    12  Aug 25 22:25   home
lrwxrwxrwx     1  root  root    29  Nov 10 10:12   initrd.img -> boot/initrd.img-6.5.0-3-amd64
lrwxrwxrwx     1  root  root    29  Oct 30 22:33   initrd.img.old -> boot/initrd.img-6.5.0-1-amd64
lrwxrwxrwx     1  root  root     7  Jan 5 2021     lib -> usr/lib
lrwxrwxrwx     1  root  root     9  Jan 5 2021     lib32 -> usr/lib32
lrwxrwxrwx     1  root  root     9  Jan 5 2021     lib64 -> usr/lib64
drwxr-xr-x     1  root  root    60  Aug 14 20:04   media
drwxr-xr-x     1  root  root     0  Jan 5 2021     mnt
drwxr-xr-x     1  root  root   226  Aug 14 17:00   opt
dr-xr-xr-x  1038  root  root     0  Oct 15 14:13   proc
drwx------     1  root  root   508  Nov 10 10:13   root
drwxr-xr-x    41  root  root  1300  Nov 13 03:25   run
lrwxrwxrwx     1  root  root     8  Jan 5 2021     sbin -> usr/sbin
drwxr-xr-x     1  root  root    34  Aug 26 03:01   srv
dr-xr-xr-x    13  root  root     0  Oct 15 14:13   sys
drwxrwxrwt     1  root  root  8552  Nov 13 13:56   tmp
drwxr-xr-x     1  root  root   104  Sep 25 12:17   usr
drwxr-xr-x     1  root  root   148  Oct 13 02:41   var
lrwxrwxrwx     1  root  root    26  Oct 30 22:33   vmlinuz -> boot/vmlinuz-6.5.0-3-amd64
lrwxrwxrwx     1  root  root    26  Oct 30 22:33   vmlinuz.old -> boot/vmlinuz-6.5.0-1-amd64
```

- `mount`, `df`, `du`

Users, Groups, Permissions

- Each file has *owner* and *group* (and *others*) with permissions
  - *R*ead (4), *W*rite (2), e*X*ecute (1) - triple of 3 bits (octal numbers)
  - `0700: -rwx------`
  - `0750: -rwxr-x---`
  - `0755: -rwxr-xr-x`
  - `0600: -rw-------`
  - `0640: -rw-r-----`
  - `0644: -rw-r--r--`
  - First, if 4 octals, is special; see `chmod`
- `umask` sets *default permission mask* (bits to remove!!!)
  - `umask 0077`
  - `umask 0022 # Poor default!`
- *Change owner/group*: `chown`, `chgrp` (if you have permission)

Some Commands

`ls`	list contents of directory; long (`-l`); all (`-a`)
`cd`	change directory
`pwd`	print working directory
`cp`	copy file/directory (`-r`)
`mv`	move file
`ln`	link to file; symbolic link (`-s`)
`rm`	remove file; directory (`-r`); forced (`-f`)
`mkdir`	make directroy; hierarchy (`-p`)
`cpio`	copy files to/from archives, including between directories (`-pdvmua`)
`cat`	concatenate files
`sort`	sort data
`find`	find files
`grep`	show matching lines
`sed`	stream editor
`awk`	pattern scanning and processing
`vi`	visual editor (`vim`)
`man`	view manual page

IO

- All *IO via file descriptors* with defaults
  - stdin (1 – keyboard); stdout (2 – display); stderr (3 – display)
  - These are often default input/output files (already open)
- *Redirect* IO
  - `>, >>, <, <<, 2>&1, ...`
  - `ls -la >data.txt`
  - `ls -la / >>data.txt`
  - `sort <data.txt >sorted.txt`
  - Also see bash's `exec`
- *Pipelines*
  - `COMMAND1 | COMMAND2 ...`
  - `COMMAND1 |& COMMAND2 ...`
  - `find ~/all -name "*.txt" | cpio -pdvmua ~/text`
  - `(ls -la; ls -la /) | sort >sorted.txt`

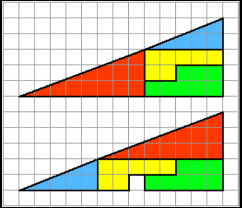
Scripts

```
#!/bin/bash

for l in *.cv; do
  mv "$l" "$(echo $l | sed -e 's/.cv$/.csv/')"
done

grep '^c.*,' *.csv | cut -d, -f1,4 | while read M; do
  mark="`echo $M | cut -d, -f2`"
  student="`echo $M | cut -d, -f1`"
  if [ "$mark" -ge 40 ]; then
    echo $student,pass
  else
    echo $student,fail
  fi
done
```

Processes and Environments

- A *process* is a program running in memory
  - with PID, PPID, TTY, UID, GID
  - `ps auxw`
  - `jobs`
  - `top`
- *Background jobs*: `COMMAND &` or `CTRL-Z + bg / fg`
- *Killing* a process (send a signal)
  - `kill %1`
  - `kill 1 # PID, never 1 - it kills everything :P`
  - `kill -9 1 # Send KILL signal`
- *Environment*:
  Set of variables with special meaning (depends also on shell and commands/prgrams)
  - `$HOME`, `$SHELL`, `$TERM`, `$PATH`, . . .
  - Set in `/etc/propfile`, `~/.profile`, `~/.bash_profile`, `~/.bashrc`, `/etc/environment`, . . .
  - Set with `export` (otherwise not propagated to child processes)

Philosophy



*

- *UN*X philosophy*: programs should
  - do one thing
  - do it well
  - work together
  - process text streams
- *Worse is better*
  - Make it work, not perfect — perfection is usually ill-defined
  - Let the user fix it, even if it is complex — they are not dumb
  - Share the code freely (not fake openly) — at least it can be fixed
- *UIs*
  - Shells have a steep learning curve — at least there is one
  - Using and programming are equivalent — greed says otherwise
  - GUIs can *deceive* (*), *depend on culture* (💾), and mostly *fail*:
    limited interoperability, automation, concurrency, accuracy/precision, choices
- *Over-engineering/abstraction/design* only creates limits — failure of OOP
- *Freedom*: who controls/owns the box? — your (collective) choice

The Future and Further Resources

- *Future developments*
  - Diverse hardware require diverse OS: quantum, biology, chemistry, beyond CMOS, custom ICs
  - UN*X provides somewhat surprising commonality
  - Universality is dead — no need to run doom on your fridge
  - Maybe AI, but not if it produces expected answers instead of reasoning and proofs
- The Linux Documentation Project Guides + HOWTOs: www.tldp.org
  - Bash guide for beginners
  - GNU/Linux command-line tools summary
  - The Linux system administrator's guide
  - Linux filesystem hierarchy
- Books from O'Reilly, like "UNIX power tools" (Powers & Peek)
- https://wikibooks.org/wiki/Learning_the_vi_Editor
- Find more yourself. . .