

The File System

Hands-On UNIX System Administration DeCal

Lecture 2 — 31 January 2011

Last time



Last time

- When you get stuck, **RTFM**: `man` and `apropos` are your friends, and some programs accept the `--help` command line option.
- **Text streams**: `stdin`, `stdout`, `stderr`. Pipe `stdout` to `stdin` (`foo | grep oof`), save them to files (`bar 2>err`, `baz >> log`)...
- **Substitution**: “pipe” `stdout` to command-line arguments (`host `curl whatismyip.org``).

Last time

- This tripped up a few people last time:
`find ~ -name '*.*' 2>/dev/null`
- The apostrophes *escape* the `.*`, keeping your shell from expanding it.
- Compare `echo .*` and `echo '*.*`.

Administrivia

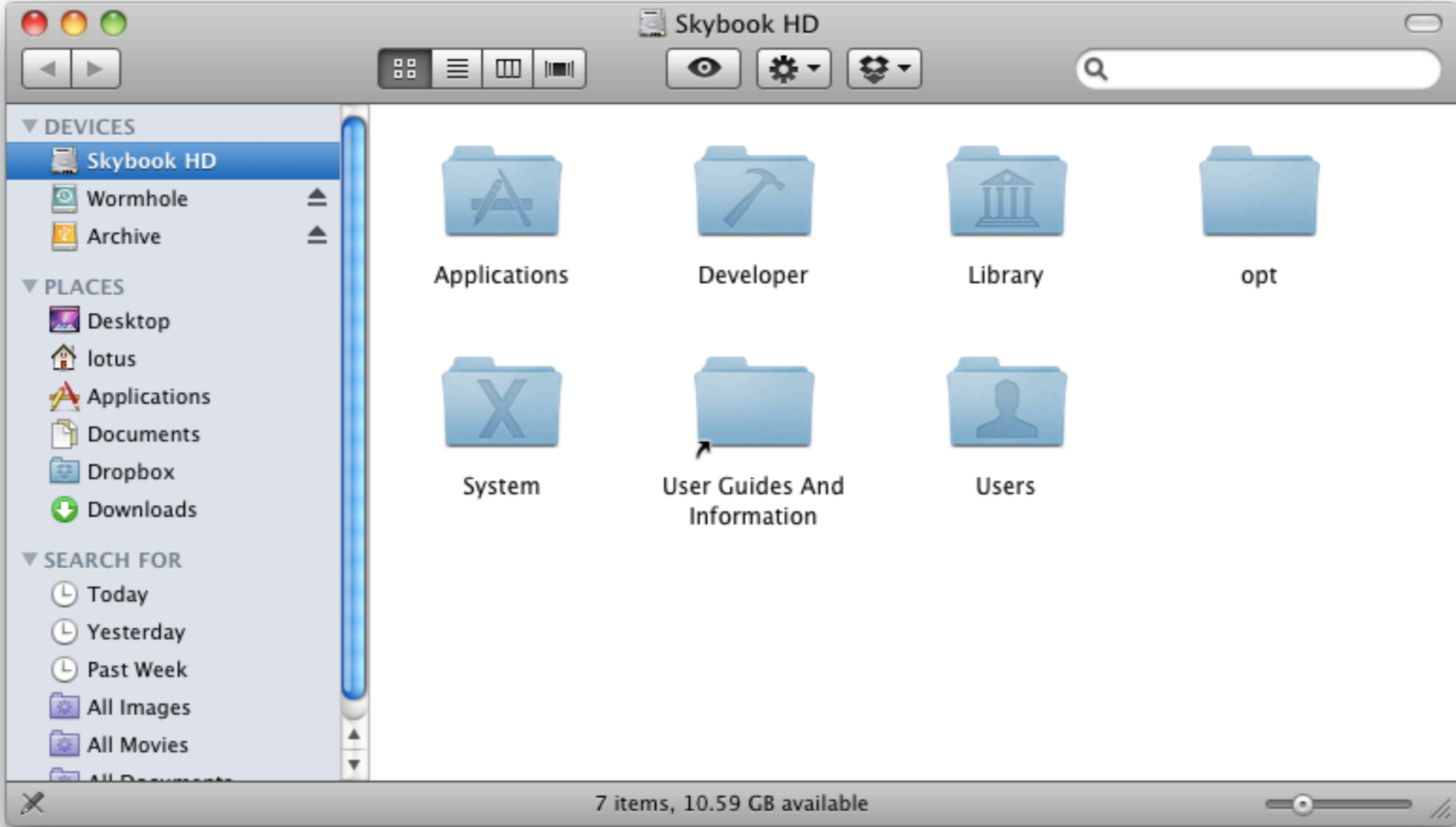
- **Office hours!** If you need help, e-mail me and/or Eugene and we'll figure something out. (Ask questions if you get stuck!)
- **Enrollment.** I've asked the CS dept. to drop last week's no-shows — everyone who signed the roster last week should get into the course.

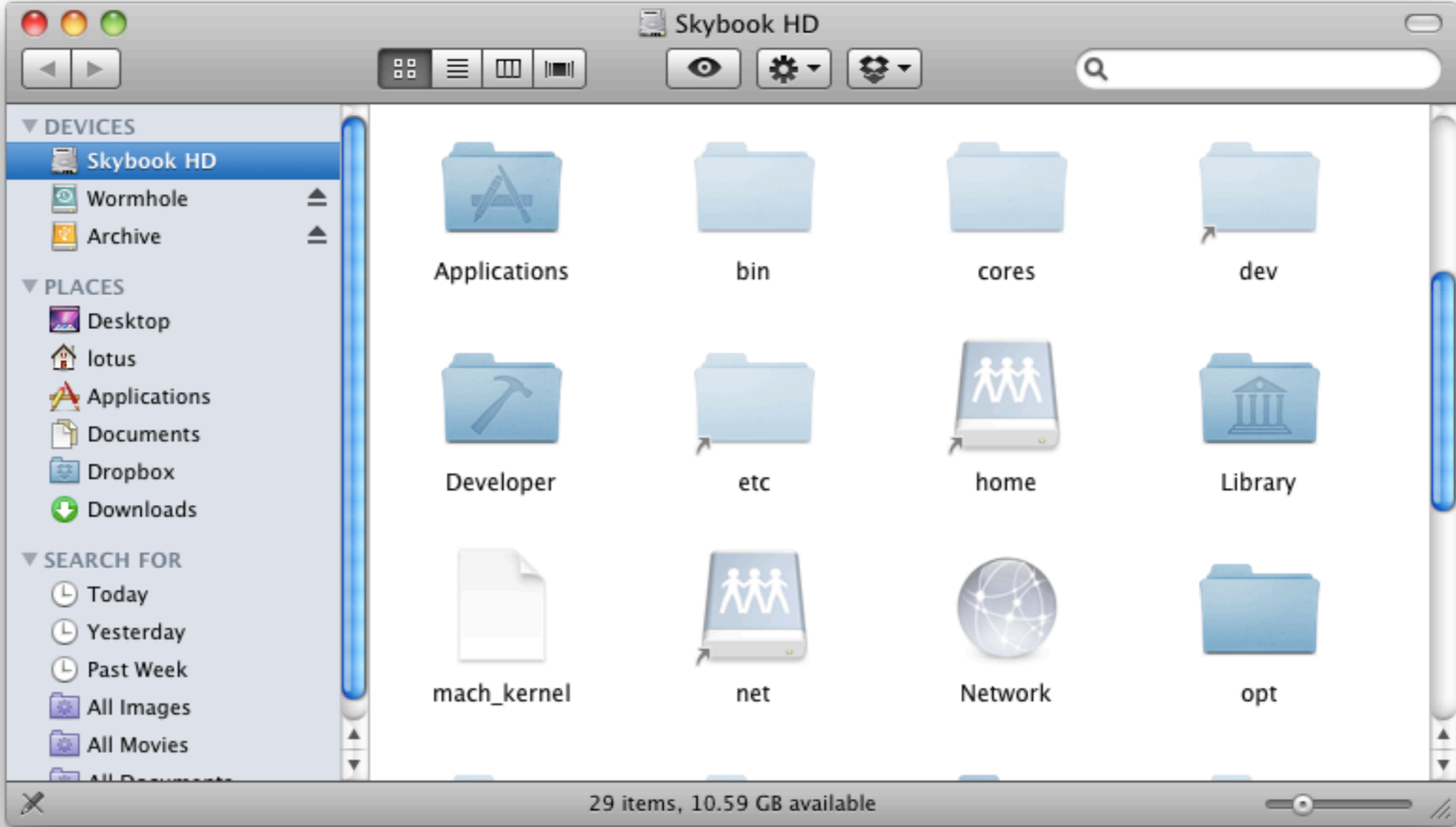
Windows' file system

- Each storage device has a drive letter associated with it (usually). Generally drive A:\ is floppy, C:\ is HDD, D:\ is DVD.
- Core OS files live in C:\Windows, third-party applications in C:\Program Files, and your data in C:\Users\You.

Mac OS X's file system

- Storage devices don't have letters; they appear on the Desktop and live in the hidden folder `/Volumes`. `/` is your OS disk.
- Core OS files live in `/System` and `/Library`, applications in (big surprise) `/Applications`, and your data in `/Users/you`.
- ... Well, on the surface, anyway.





Generic *nix file system

- This varies from platform to platform, but the general idea is the same on all *nixes. For specifics, check manpages (“hier” on BSD and Linux, “filesystem” on Solaris).
- Core software lives in /(s)bin; other software in /usr/(s)bin; user data in /home. (Core files: /bin/**ls**, /bin/**cat**... ~~Bloatware~~ nonessential software: /usr/bin/**emacs**...)

In more detail...

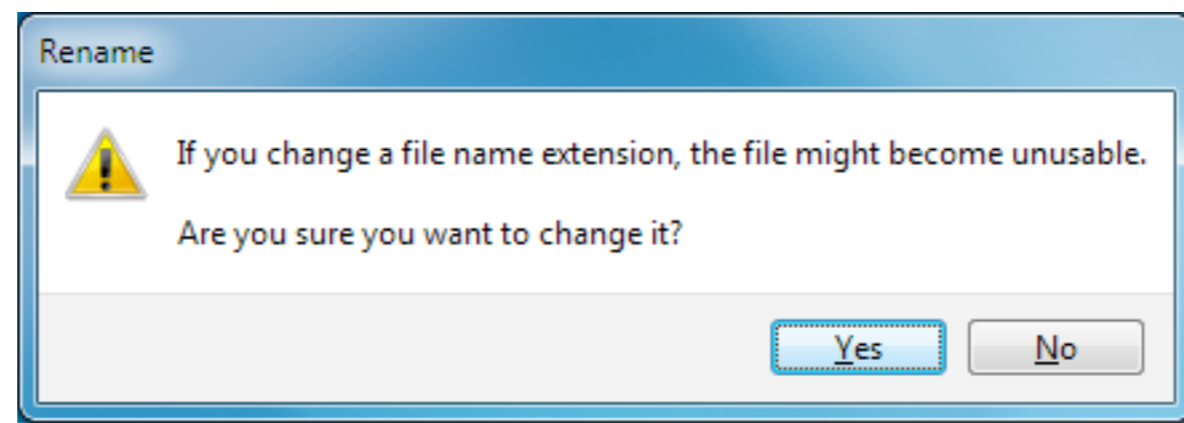
- **/boot** — bootloader files (GRUB)
- **/bin, /sbin** — critical binaries
- **/lib** — critical libraries (think “DLLs”)
- **/usr/bin, /usr/sbin** — normal programs
- **/usr/lib** — normal libraries
- **/proc** — “procfs,” an interface to the kernel

In more detail...

- **/home** — user home directories
- **/root** — root's homedir
- **/etc** — configuration files
- **/var** — /var/log, /var/mail, etc.
- **/dev** — device files (e.g., /dev/sda1)
- **/tmp** — temporary files

Filetypes

Windows: Filetypes are distinguished by (and the OS is wholly reliant on) **filetype extensions** (.exe, .doc...).



Filetypes

- *nix? Filetype extensions aren't as important. "A rose by any other name smells just as sweet" — UNIX systems can determine filetypes intelligently.
- But wait, there's more! On top of regular files and directories, *nix systems have **character** and **block specials**, **symbolic links**, **sockets**, and **FIFOs**.

Mount points

- On the login server at OCF.Berkeley.EDU, `/` is a 16GB drive — but `/home` contains 670GB of data! How is this possible?
- You can *mount* filesystems at arbitrary *nodes* in the filesystem. The aptly-named `mount` command lists and manages them.
- In this case, `/home` is a 4TB network share.

Types of filesystems

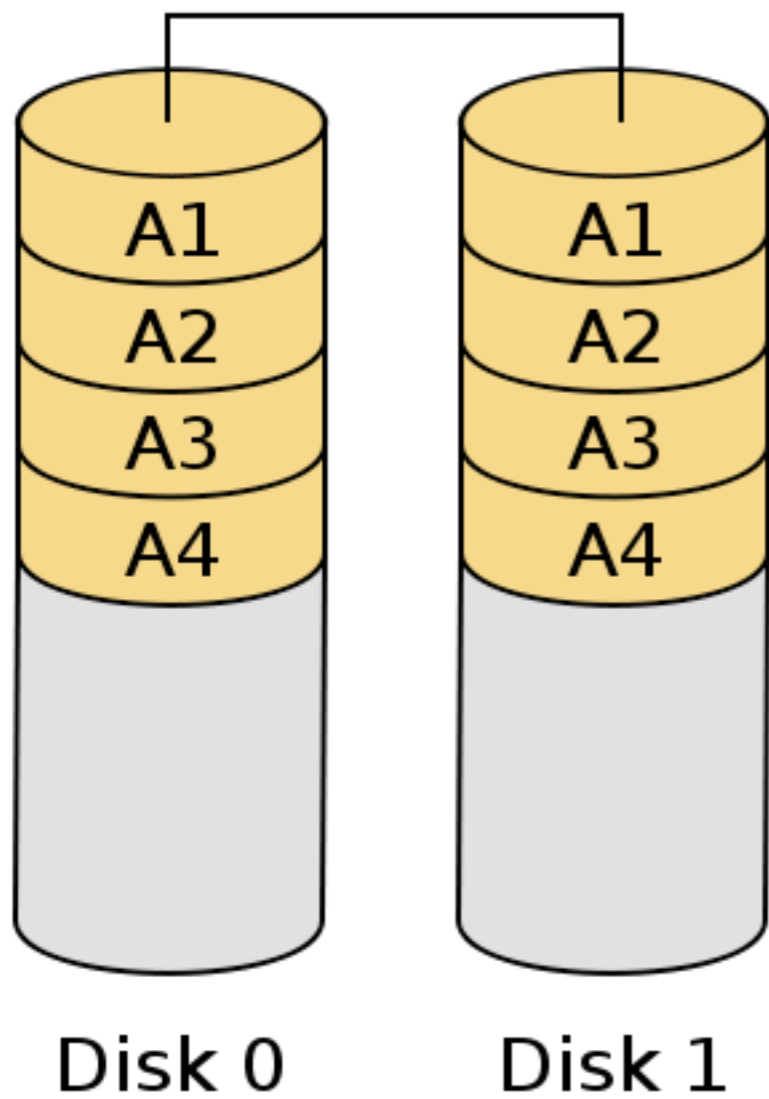
- You've probably heard of the standard DOS/Windows filesystems (**FAT32, NTFS**).
- Mac OS: **HFS** (Mac Classic), **HFS+** (OS X)
- Linux: "extended file system" (**ext3, ext4**)
- Other FSs: **UFS, XFS, ZFS, reiserFS...**

RAID

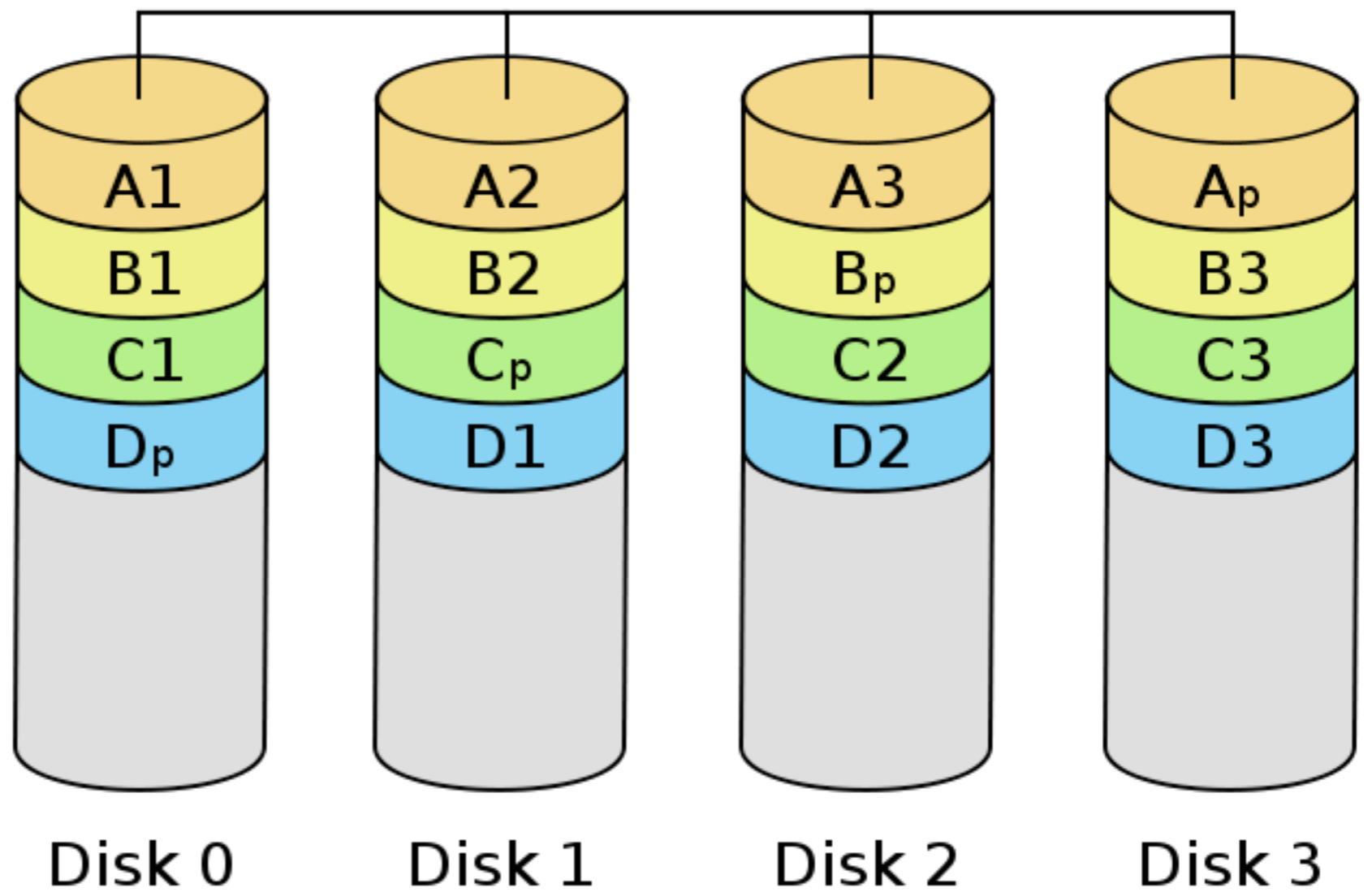
- **Redundant Array of Independent Disks.**
Invented at Cal by, among others, Dr. Patterson (he's teaching CS 61C). Lets you combine multiple hard drives to increase speed, capacity, and reliability.
- Standard **RAID levels:** RAID1 (mirror), RAID5 (popular multi-drive solution), RAID0 (not redundant). Can be nested.

RAID

RAID 1



RAID 5





Not just for servers

<http://macguild.org/raid.html>