High level picture of a Unix machine

Amitabha Sanyal

The problem of bootstrapping

- Basic problem: For the computer to be usable, an operating system (OS) must be loaded into the memory (RAM) of the computer.
- Normally, the OS can load a program into memory.
- But the OS has been erased from RAM when you switched off the machine.

- Question: Who loads the OS?

What happens when you switch on a computer

- The processor executes code called BIOS (Basic Input Output System). This code starts at a fixed location in flash memory on the motherboard.
  - Flash memory means that the code is not erased even during power down,
- The program in BIOS does Power on Self Testing (POST) to check the hardware units.
- BIOS does not know the specifics of any particular OS.
- BIOS determines the boot device i.e. the device which contains the “kernel” of an OS that can be loaded. This is normally the hard disk, but can be manually overridden.
- BIOS loads a program called bootloader that would eventually load the OS in memory. This program is called GRUB (GRand Unified Bootloader) in Unix.

GRUB takes over

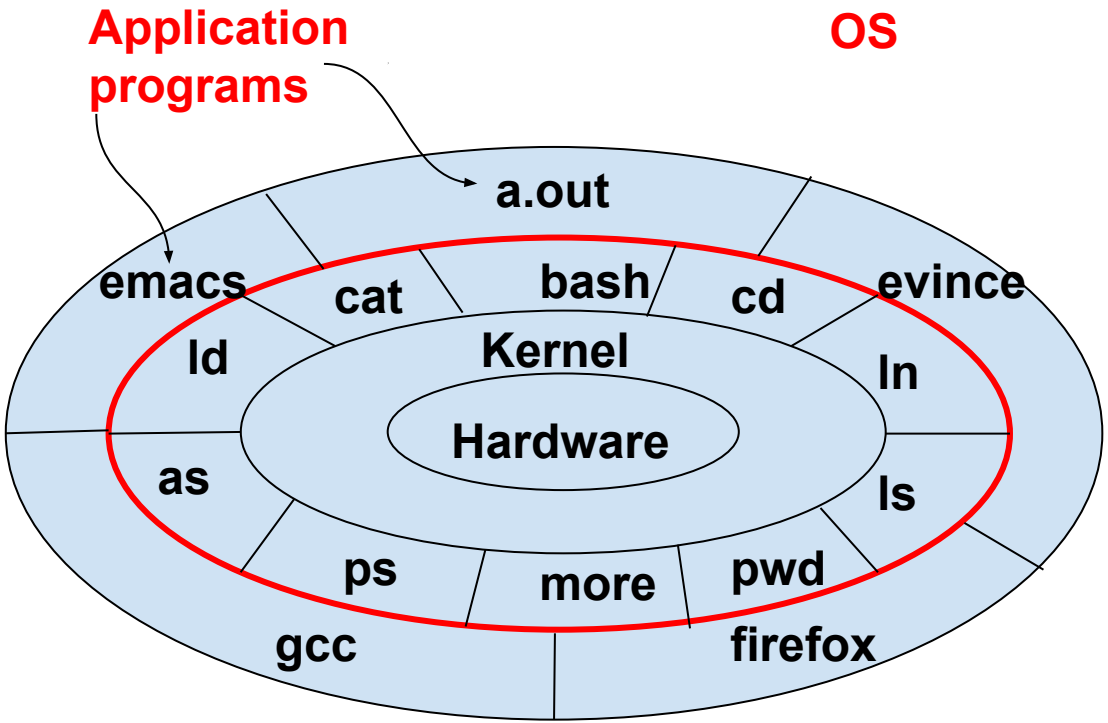
- GRUB has knowledge of the Unix file system.
- GRUB display a list of available OS kernels (16.04, 18.04, Windows)
- Loads the default kernel or the user selected kernel into memory.
- And enters one of the runlevels:
  - 0 - Halt
  - 1 - Single-user text mode
  - 2 - Not used (user-definable)
  - 3 - Full multi-user text mode
  - 4 - Not used (user-definable)
  - 5 - Full multi-user graphical mode (with an X-based login screen)
  - 6 - Reboot
- Use the `runlevel` command to find the current runlevel.

Processes

- After this one can think of the activities of the system in terms of processes.
- A program in execution is called a process.
  - The first process is **`init`**
  - `init can spawn more processes (run “ps aux”) to see this.`
  - `Many processes that run in the background called daemons, processes such as desktop window manager (gdm).`
  - `One of the processes that it runs is a password based authenticator.`
  - `After this it allows you to raise a terminal and puts you in a bash shell.`

At the end of the boot process

- The kernel with some common application programs is what we call “the OS”.
- A special process called bash keeps on listening to the keyboard.
- When you invoke a command, executes the program corresponding to the command.
- The executing program interacts with the hardware through the kernel.



Running a program.

Suppose you wanted to write a program and execute it.

1. **bash** reads your command "**emacs test.c**" and starts the execution of the **emacs** editor.
2. **emacs** reads your editing commands from the keyboard and writes to the console and the filesystem on the disk.
3. Control comes back to **bash**. You type in "**gcc test.c**".
4. Recognizing this bash, sets the execution of the C compiler. Creates a file called **a.out**.
5. Control comes back once again to bash. You type in "**./a.out**".
6. **bash** sets the execution of **a.out**. the result is 120. Control returns to **bash**. You log out and go for chai.