Processes

The Linux kernel is the core of a Linux installation. The kernel manages memory, provides software with a way to access the hard disk, doles out CPU time, and performs other critical low-level tasks. The kernel is loaded early in the boot process, and it's the kernel that's responsible for managing every other piece of software on a running Linux computer.

More about processes

Normally at boot, one process is started `/sbin/init`. Then everything else is managed by init and are called child processes. (sometimes also referred to as daemons)

Internally, the kernel maintains process information in the process table . Tools such as ps and top (described shortly) enable you to view and manipulate this table.

Process ID numbers

Every process has a PID. Init's id is normally 1. Each process also has a parent process id(PPID).

- `ps aux --forest` can help show parent/child relationships

More on Processes

- kill: find the pid and issue the command
- We can also send it a signal like `kill -9 PID`
- killall: kills by process name instead of pid
- example: `killall -9 testme`

Process related commands

- ps , ps aux
- fg
- bg
- control + z
- jobs
- top
- & # to run in background
- sleep

Process states

code	state
D	uninterruptable sleep (I/O is happening)
R	running
S	interruptable sleep (waiting for event to complete)
T	stopped or paused
Z	defunct (zombie)

Process Signals

Messages sent between processes. A numeric value.

Names:

- interrups (Cntrl + c)

- sleep (Cntrl + z)
- kill
- etc..

Process related files

- `/proc/cpuinfo`
- `/proc/meminfo`