**Linux Processes**

A process is a program or application that is currently in execution

An instance of a running application

**Background process**

* Not normally connected to any foreground process
* Doesn’t expect any input
* The process that controls the keyboard or the mouse etc.

**Foreground**

* Browser
* Terminal
* Anything that’s interacted with

**Creating a process**

* Normally done by running a binary
* Can be done using syscalls
  + System(), Fork(), Exec()

**Identifying a process**

* Any process is assigned a PID
  + Unique number that keeps track of what process is which in a multi-user based system
* When an app is first run, it’s considered the parent and will have a PID
* Any process creates by this parent is assigned a PID and a PPID (parent PID)
  + Usually when a parent process is killed, so are the fork processes

**Top level / Mother process**

* Process first started by kernel itself
* This has a PID of 1, in most cases this is the INIT but can also be systemd
* All processes on the system is a child of PID 1 as its job is to manage processes

**States of a process**

* Running
* Waiting
* Stopped
* Zombie/Orphan

**State of Running**

* A process is actively running or waiting for a CPU core to be assigned to it
* Depends on the number of cores the CPU has
* Can only be so many processes running as there are only so many cores a CPU has

**State of Waiting**

* Waiting for one of two things
  + Events to happen like user unput
  + System resources to become available
* 2 types of waiting
  + Interrupted: some hardware-based conditions can cause this
  + Uninterrupted: this process cannot be stopped by any event or signal

**State of Stopped**

* Process has been stopped by some means
* Can be a signal to stop from the user or system
* Can also be in this state during the time a debugger is attached to it, analysing for bugs
  + This is more of a ‘halted’ state

**State of Zombie or Orphan**

* State is no longer alive AKA dead
* The reason its still showing Zombie or ‘z’ is due to it still being on the process entry table in the kernel
  + Keeps track of all processes and IDs so process 1 knows all processes and states
* Don’t want the table to be too backed up or it can cause problems

Pidof

* Gives PID of process
* Pidof firefox
  + Prints PID string

Ps

* Listing of processes
* -aux
  + Will show every process being ran, despite what you have access to

Top

* Provides info4rmation on CPU and memory usage by processes
* You can then order and sift through these items via PID or CPU usage or time etc.

Kill -l

* Brings up different kill options
* Kill 9 2975
  + Kill (option) PID
* Pkill will do it by name
* Killall

**HW – Nice levels**