
IT 1100 : Introduction to Operating Systems

Chapter 10 Processes

Processes

Modern operating systems are usually multitasking, meaning that they create the illusion of doing more than one thing at once by rapidly switching from one executing program to another. The Linux kernel manages this through the use of processes. Processes are how Linux organizes the different programs waiting for their turn at the CPU.

Processes

init - Receives PID of 1. program that runs when Linux starts. init will launch several processes and call init scripts.

Many of these services are called daemons... or programs that run in the background without having any user interface.

A process can launch another process - the new process is said to be a *child* process of the parent.

Processes

Each process is assigned an ID number so that the kernel can keep track of the process. (PID or Process ID). Init always get PID of 1. PIDs also have an associated UID.

Processes (Process Related Commands)

- ps - list processes for the current user on the current terminal (columns)
 - pstree - lists the processes in a tree structure identifying parent processes
 - top - dynamically lists the top processes running
 - jobs - lists the jobs run from this terminal
 - bg - put a program in the background
 - fg - put a program in the foreground
 - kill - kills a process based on the PID
 - killall - kills a process based on the name
-

Process States

A process can be in one of several states:

- *R* = running
 - *S* = Sleeping; process is waiting for an event to occur (keystroke)
 - *D* = Uninterruptible sleep; process waiting for i/o
 - *T* = stopped
 - *Z* = zombie; child has been terminated but not cleaned up by parent
-

Process key commands

- ctrl-c kills a process
 - ctrl-z puts a process in the background without killing it
-

ps options

- a - list all processes
 - x - list the BSD style
 - u - list processes for all users
 - o - customize the process list
 - `ps xao pid,ppid,comm`
-

Killing

- `kill -9 pid #` will kill a process ID
 - `killall xclock #` will kill the xclock process
-

shutdown options

System Shutdown and Restart commands

```
shutdown -r time message
```

Example: `sudo shutdown -r +5 "Server will restart in 5 minutes. Please save your work."` Example: `sudo shutdown -h now`

shutdown options

- `halt` - leaves the machine powered on

Example: `sudo halt` Example: `sudo reboot`