Process in Linux

Processes carry out tasks within the operating system.

A program is a set of machine code instructions and data stored in an executable image on disk and is, as such, a passive entity; a process can be thought of as a computer program in action.

States of Process

Running

The process is either running (it is the current process in the system) or it is ready to run (it is waiting to be assigned to one of the system's CPUs).

Waiting

The process is waiting for an event or for a resource. Linux differentiates between two types of waiting process; interruptible and uninterruptible. Interruptible waiting processes can be interrupted by signals whereas uninterruptible waiting processes are waiting directly on hardware conditions and cannot be interrupted under any circumstances.

Stopped

The process has been stopped, usually by receiving a signal. A process that is being debugged can be in a stopped state.

Zombie

This is a halted process which, for some reason, still has a task_struct data structure in the task vector. It is what it sounds like, a dead process.

Initializing a process

A process can be run in two ways:

Foreground Process

Every process when started runs in foreground by default, receives input from the keyboard and sends output to the screen.

When a command/process is running in the foreground and is taking a lot of time, no other processes can be run or started because the prompt would not be available until the program finishes processing and comes out

Background Process

It runs in the background without keyboard input and waits till keyboard input is required. Thus, other processes can be done in parallel with the process running in background since they do not have to wait for the previous process to be completed.

Initializing a process – Continued…

bg %<job_id>

The bg command is used on Linux in order to send a process to the background

fg %<job_id>

The fg command is used in order to send a process to the foreground

NOHUP

A process may not continue to run when you log out or close your terminal.

Use nohup to handle this situation.

nohup <script>.sh &

Appending an ampersand (&) will send the process to the background and allow you to continue using the terminal

Nice value

A nice value of -20 represents highest priority, and a nice value of 19 represent least priority for a process.

Changing Process Priority

Increase the priority

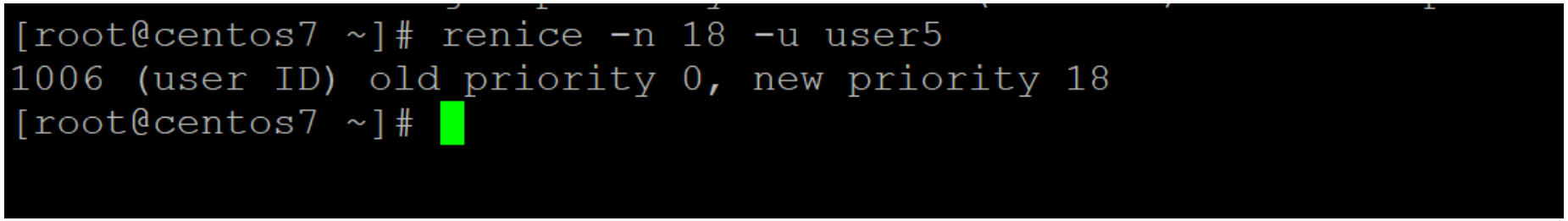
nice -n -5 -p <PID>

Decrease the priority:

nice -n 5 -p <PID>

Change the Priority of All Processes Owned by User

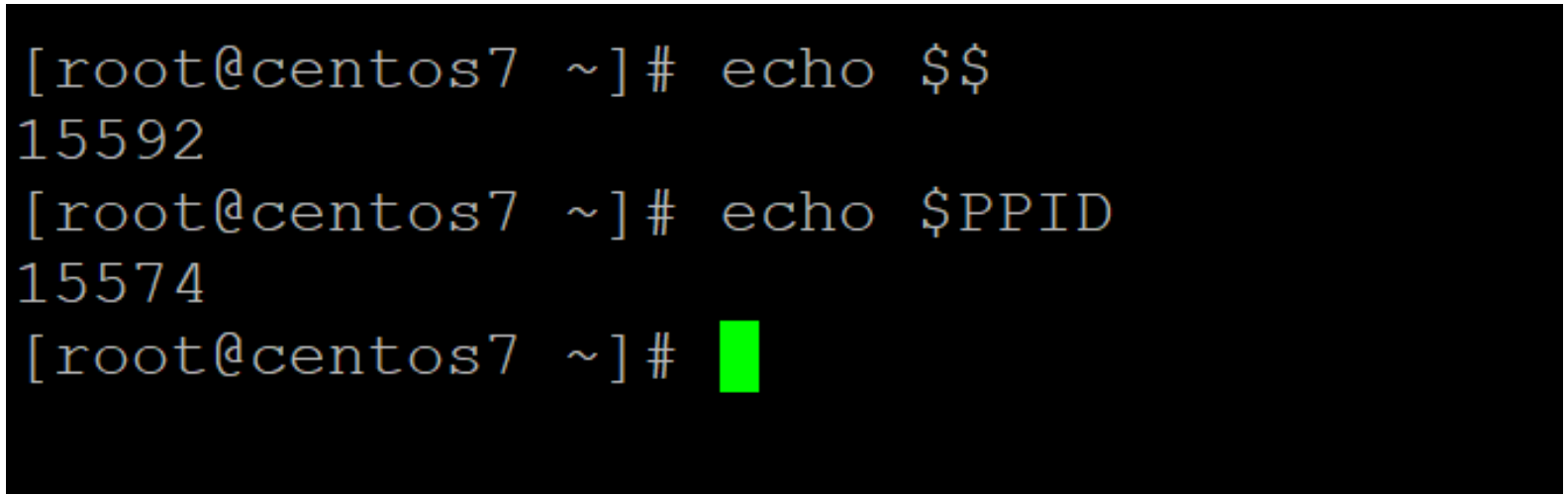
renice -n 5 -u <username>



To find the process ID and parent process ID of the current shell

echo $$

echo $PPID



How to check pid of a process

pidof <process name>

pidof system

```
[root@centos7 ~]# pidof systemd
1
[root@centos7 ~]#
```

Listing Running Processes

ps

ps -f >> full listing

ps -elf >> every running process (-e) and a full listing (-f).

Daemons

These are special types of background processes that start at system startup and keep running forever as a service; they don’t die

Parent and Child process

- Parent processes – these are processes that create other processes during run-time.
- Child processes – these processes are created by other processes during run-time.
- PID --> PID
- PPID --> Parent ID of the process

Scheduling

Preemptive Scheduling

The process can be interrupted, even before the completion.

Non-Preemptive Scheduling

The process is not interrupted until its life cycle is complete.