

# Processes

- Start a browser on your desktop
- Login as centos-x to https://85.149.72.170:5800

If you ever need a second login session, simply open a new page or a new tab.

## 1. Parents and Children.

Run the following command: **ps**

```
[centos@centos11 ~]$ ps
PID TTY          TIME CMD
3922 pts/0        00:00:00 bash
3974 pts/0        00:00:00 ps
[centos@centos11 ~]$
```

What is the PID of the bash command?

Make a note of that PID. (3922)

Run the following command: **ps -ef | grep <PID\_of\_bash\_command>**.

```
[centos@centos11 ~]$ ps -ef | grep 3922
centos    3922  3921  0 06:08 pts/0    00:00:00 -bash
centos    3975  3922  0 06:22 pts/0    00:00:00 ps -f
centos    3976  3922  0 06:22 pts/0    00:00:00 grep --color=auto
3922
[centos@centos11 ~]$
```

What is the PPID of your bash shell?

Make a note of that PID. (3921)

Run the following command: **ps -ef | grep <PPID>**

```
[centos@centos11 ~]$ ps -ef | grep 3921
centos    3921  3918  0 06:08 ?        00:00:00 sshd: centos@pts/0
centos    3922  3921  0 06:08 pts/0    00:00:00 -bash
centos    3982  3922  0 06:35 pts/0    00:00:00 grep --color=auto
3921
[centos@centos11 ~]$
```

What is the command of that process?

Use the pstree command to print the process tree of that process.

```
[centos@centos11 ~]$  
[centos@centos11 ~]$ ps tree 3921  
sshd───bash───pstree  
[centos@centos11 ~]$
```

## 2. Jobs

Run the sleep 10000 command, and use <CTRL+Z> to stop the process.

```
[centos@centos11 ~]$ sleep 10000  
^Z  
[1]+  Stopped                  sleep 10000
```

Run the following command: **jobs**

```
[centos@centos11 ~]$ jobs  
[1]+  Stopped                  sleep 10000  
[centos@centos11 ~]$
```

What is the job number?

Start the job in the background.

```
[centos@centos11 ~]$ bg %1  
[1]+ sleep 10000 &  
[centos@centos11 ~]$
```

Check the status with the jobs command: **jobs**

```
[centos@centos11 ~]$ jobs  
[1]+  Running                  sleep 10000 &  
[centos@centos11 ~]$
```

Run the following command: **ps -o pid,ppid,stat,cmd**

```
centos@centos11 ~]$ ps -o pid,ppid,stat,cmd  
PID  PPID  STAT  CMD  
3922  3921  Ss    -bash
```

```
3986 3922 S    sleep 10000
4016 3922 R+   ps -o pid,ppid,stat,cmd
[centos@centos11 ~]$
```

Find out what the different values in the *STAT* column represent.

Send signal 15 to the sleep process.

```
[centos@centos11 ~]$ kill -15 %1
```

List the status of your jobs.

```
[centos@centos11 ~]$ jobs
[1]+  Terminated                  sleep 10000
[centos@centos11 ~]$
```

Start two sleep processes in the background and log out.

```
[centos@centos11 ~]$ sleep 20000&
[2] 4189
[centos@centos11 ~]$ sleep 30000&
[3] 4190

[centos@centos11 ~]$ jobs
[1]+  Stopped                      fb
[2]   Running                      sleep 20000 &
[3]-  Running                      sleep 30000 &
[centos@centos11 ~]$ exit
```

Log in to your system again and list the sleep processes.

```
[centos@centos11 ~]$ ps -eo pid,ppid,cmd | grep [s]leep
4189      1  sleep 20000
4190      1  sleep 30000
[centos@centos11 ~]$
```

What is the PPID of both sleep

```
[centos@centos11 ~]$ ps -eo pid,ppid,cmd | grep -E "[s]leep|PPID"
PID      PPID  CMD
4189      1    sleep 20000
4190      1    sleep 30000
4233     4196  grep --color=auto -E [s]leep|PPID
[centos@centos11 ~]$
```

### 3. Priorities

Using your favorite editor, create the following script, call it 'divide' and make it executable.

```
[centos@centos11 ~]$ cat divide
while :
do
    ((232345/434))
done

[centos@centos11 ~]$
[centos@centos11 ~]$ chmod +x divide
[centos@centos11 ~]$
```

Start the divide script in the background, twice.

```
[centos@centos11 ~]$ ./divide &
[1] 2185
[centos@centos11 ~]$ ./divide &
[2] 2186
```

Use the top command to see that the two background processes get an equal share of the available cpu-cycles.

```
top - 07:32:10 up 3 min, 1 user, load average: 1.93, 0.82, 0.31
(snipped)


| PID  | USER   | PR | NI | VIRT   | RES  | SHR  | S | %CPU | %MEM | TIME+   | COMMAND |
|------|--------|----|----|--------|------|------|---|------|------|---------|---------|
| 2186 | centos | 20 | 0  | 115384 | 1008 | 624  | R | 49.8 | 0.1  | 1:12.14 | bash    |
| 2185 | centos | 20 | 0  | 115384 | 1008 | 624  | R | 49.5 | 0.1  | 1:13.67 | bash    |
| 1    | root   | 20 | 0  | 128092 | 6700 | 3948 | S | 0.0  | 0.7  | 0:00.43 | systemd |


(snipped)
```

Note the PIDs of the two top processes.

Use **<CTRL+C>** to stop the top process.

Use the renice command to lower the priority of one of the background processes.

```
[centos@centos11 ~]$ renice 10 2186
2186 (process ID) old priority 0, new priority 10
[centos@centos11 ~]$
```

Run the top command again and watch the difference in percentage of cpu priority.

```
top - 07:32:10 up 3 min, 1 user, load average: 1.93, 0.82, 0.31
(snipped)


| PID  | USER   | PR | NI | VIRT   | RES  | SHR  | S | %CPU | %MEM | TIME+   | COMMAND |
|------|--------|----|----|--------|------|------|---|------|------|---------|---------|
| 2185 | centos | 20 | 0  | 115384 | 1008 | 624  | R | 90.0 | 0.1  | 7:31.64 | bash    |
| 2186 | centos | 30 | 10 | 115384 | 1008 | 624  | R | 9.6  | 0.1  | 3:51.40 | bash    |
| 1    | root   | 20 | 0  | 128092 | 6700 | 3948 | S | 0.0  | 0.7  | 0:00.43 | systemd |


(snipped)
```

Use **<CTRL+C>** to stop the top process.

Try to set the priority back to 0.

```
[centos@centos11 ~]$ renice 0 2186
renice: failed to set priority for 2186 (process ID): Permission denied
[centos@centos11 ~]$
```

Why is this not allowed? Can you fix this?