Assignment for Lab 11
Processes
Computer Science 240

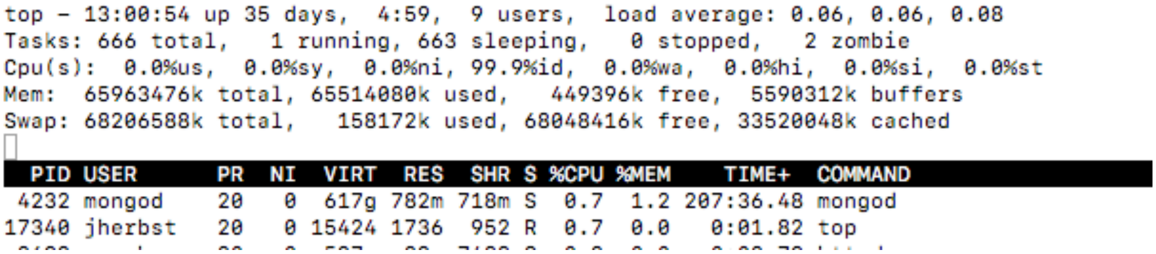
Explore Processes in a Running System

On a lab machine, open a couple applications: use Firefox to view both the CS 240 course web page and at least one additional more dynamic web page, such as Facebook. Also, open at least one other application.

In a Terminal, run the top command to visualize the currently executing processes and the resources they consume by entering the following command at the command prompt:

```
top
```

Examine the output. which looks something like this:



```
top - 13:00:54 up 35 days,  4:59,  9 users,  load average: 0.06, 0.06, 0.08
Tasks: 666 total,   1 running, 663 sleeping,   0 stopped,   2 zombie
Cpu(s):  0.0%us,  0.0%sy,  0.0%ni, 99.9%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:  65963476k total, 65514080k used,   449396k free,  5590312k buffers
Swap: 68206588k total,   158172k used, 68048416k free, 33520048k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM     TIME+  COMMAND
 4232 mongod    20   0  617g 782m 718m S  0.7  1.2 207:36.48 mongod
17340 jherbst   20   0 15424 1736  952 R  0.7  0.0   0:01.82 top
```

Read the “Tasks” line at the very beginning of the output, interpreting “Tasks” as “processes”.

1. How many processes are running? Sleeping? Stopped? Zombie?

Read the “CPU(s)” line, which shows the percentage of the time that the CPU is spending executing user and operating system kernel code, vs. being idle (and a few other categories we will ignore). These levels probably fluctuate at each sample that *top* displays.

2. Also displayed are a list of processes ranked by the percentage of CPU time they have used in the most recent time window. Which processes are using the most CPU time? About how much?

Run the ps command:

```
ps
```

By default, it lists only the processes run under your current login session. (Each terminal window you open actually creates a new login session and runs a shell in it.)

For example:

```
23314 pts/1          00:00:00 bash
30086 pts/1          00:00:00 ps
```

To see all the list of all processes belonging to you, enter:

```
ps ux
```

3.How many have used at least 1 second of CPU time? (see the TIME column, in minutes:seconds form)

To see the list of all processes run by all users on this machine, enter:

```
ps aux
```

To list the contents of the /proc filesystem:

```
ls /proc
```

NOTE: The /proc filesystem is provided by the Linux kernel as an interface to inspect information about process scheduling, individual processes, and other operating system status information

You will see something like this, which is a list of subdirectories:

```
1    12327  171    217    248     29      315     3518    47      8512
10   12328  172    218    24801   290     3152    3519    48      86
100         124    173    219     249     29040   316     352     483     87
101         125    174    22      24946   291     31654   3520    49      88
```

The /proc filesystem has a subdirectory with information about each living process. Each directory is named with the associated PID (Proccess ID) of a process that is currently running.

Examine the interrupts file by entering:

```
cat   /proc/interrupts
```

4.How many interrupts have occurred for scheduling (context-switching)? System calls (traps, labeled “Function Call”)?

Find the PID (Process ID) of *firefox* by entering:

```
top
```

and find *firefox* and its listed PID.

Change into that directory (for example, if the PID of *firefox* is 1179):

```
cd /proc/1179
```

Inspect its status information by showing the contents of the *status* file:

```
cat status
```

5. How many context switches has Firefox experienced? (Look for “switches” or “ctxt switches”.**)**

6. How many child processes has Firefox created? (See the “task” subdirectory or run the pstree -p command to see the hierarchy of process ancestry.)