Welcome to CS140U

This class and all CS courses do contain a lab component that will require the submission of an email to the instructor, or a posting in the discussion board or use the dropbox folder. Do not worry if you see a series of 10 labs in the dropbox, as explained, some labs will be submitted in a different manner and the grades will be kept in my spreadsheet. Some labs may require group participation. For the campus students, the course is labeled in the schedule as CLWEB (classroom meetings with lab component online)

Your labs will count towards 10% of your final grade. The labs are scheduled to be 3 hours long but in many cases you will probably finish the activity earlier. The labs will have due dates and once the submission period ends, you will not be able to submit it.

It is understood that a few students may not be able to be present during a lab session due to several reasons. If this is your case, you will need to submit your lab activity prior to the lab meeting. These lab meetings will be announced via email and in the "New material available Announcement" Topic section in the discussion board. Please make sure to ask questions so you do not miss out on the lab activities.

Answer for Lab 6

On the top of all your documents, always include these items:

your name:
date:
Lab number

Repeat the process that you used to save the log file that will be produced by this lab. You can either use the log feature on putty or the script command at the shell prompt.

Login to our system and follow these steps while recording your session. Once your session has been recorded next to command prompt, describe what you just have done. i.e. :

```
wmorales@syccuxfs01:~> ps              << this command sows the running processes


PID TTY        TIME CMD
1257 pts/5   00:00:00 bash
1596 pts/5   00:00:00 ps

wmorales@syccuxas01:~>
```

>>>>>>>>>>>>>>>> start here <<<<<<<<<<<<<<<<<

```
$ ps ax
```

```
$ ps –ef
```

```
$ ps aux
```

```
$ps –ef f
```

```
$ ps -f -u wmorales

$ ps aux --sort=-pcpu,+pmem

$ ps aux --sort=-pcpu | head -5

$ ps -f -–forest

$ ps -e -o pid,uname,pcpu,pmem,comm

$ ps -e -o pid,uname=USERNAME,pcpu=CPU_USAGE,pmem,comm

$ ps -e -o pid,comm,etime

$ watch -n 1 'ps -e -o pid,uname,cmd,pmem,pcpu --sort=-pmem,-pcpu | head -15'
```

How to use ps command

1. Display all processes

The following command will give a full list of processes

```
$ ps ax
```

Pipe the output to "less" to make it scrollable.

```
PID TTY     STAT   TIME COMMAND

   1 ?       Ss     0:02 /sbin/init showopts

   2 ?       S     0:00 [kthreadd]

   3 ?       S     0:00 [ksoftirqd/0]

   5 ?       S<     0:00 [kworker/0:0H]

   7 ?       S     0:00 [migration/0]

   8 ?       S     0:00 [rcuc/0]

   9 ?       S     0:00 [rcub/0]

   10 ?       S     0:01 [rcu_preempt]

----------- cut ---------------
$ ps -ef

UID       PID PPID C STIME TTY         TIME CMD
```

```
root      1    0 0 Oct30 ?      00:00:02 /sbin/init showopts

root      2    0 0 Oct30 ?      00:00:00 [kthreadd]

root      3    2 0 Oct30 ?      00:00:00 [ksoftirqd/0]

root      5    2 0 Oct30 ?      00:00:00 [kworker/0:0H]

root      7    2 0 Oct30 ?      00:00:00 [migration/0]

root      8    2 0 Oct30 ?      00:00:00 [rcuc/0]

root      9    2 0 Oct30 ?      00:00:00 [rcub/0]

root     10    2 0 Oct30 ?      00:00:01 [rcu_preempt]

root     11    2 0 Oct30 ?      00:00:01 [rcuop/0]

root     12   2 0 Oct30 ?      00:00:01 [rcuop/1]

root     13    2 0 Oct30 ?      00:00:00 [rcu_bh]

root     14    2 0 Oct30 ?      00:00:00 [rcuob/0]

root     15    2 0 Oct30 ?      00:00:00 [rcuob/1]

root     16    2 0 Oct30 ?      00:00:00 [rcu_sched]
```

Use the "u" option or "-f" option to display detailed information about the processes

```
$ ps aux

USER        PID %CPU %MEM   VSZ   RSS TTY     STAT START   TIME COMMAND

root          1 0.0 0.1 48356 4524 ?       Ss   Oct30   0:02 /sbin/init show

root          2 0.0 0.0     0     0 ?       S    Oct30   0:00 [kthreadd]

root          3 0.0 0.0     0     0 ?       S    Oct30   0:00 [ksoftirqd/0]

root          5 0.0 0.0     0     0 ?       S<    Oct30   0:00 [kworker/0:0H]

root         7 0.0 0.0     0     0 ?       S    Oct30   0:00 [migration/0]

root          8 0.0 0.0     0     0 ?       S    Oct30   0:00 [rcuc/0]

root          9 0.0 0.0     0     0 ?       S    Oct30   0:00 [rcub/0]

root        10 0.0 0.0     0     0 ?       S    Oct30   0:01 [rcu_preempt]

root        11 0.0 0.0     0     0 ?       S    Oct30   0:01 [rcuop/0]

root        12 0.0 0.0     0     0 ?       S    Oct30   0:01 [rcuop/1]

root        13 0.0 0.0     0     0 ?       S    Oct30   0:00 [rcu_bh]
root        14  0.0  0.0      0      0 ?        S    Oct30    0:00 [rcuob/0]

root        15  0.0  0.0      0      0 ?        S    Oct30    0:00 [rcuob/1]

root        16  0.0  0.0      0      0 ?        S    Oct30    0:00 [rcu_sched]

root        17  0.0  0.0      0      0 ?        S    Oct30    0:00 [rcuos/0]

root        18  0.0  0.0      0      0 ?        S    Oct30    0:00 [rcuos/1]

------------------             cut --------------

$ ps -ef f

UID        PID  PPID  C STIME TTY          TIME CMD

root         1     0  0 Oct30 ?        00:00:02 /sbin/init showopts

root         2     0  0 Oct30 ?        00:00:00 [kthreadd]

root         3     2  0 Oct30 ?        00:00:00 [ksoftirqd/0]

root         5     2  0 Oct30 ?        00:00:00 [kworker/0:0H]

root         7    2  0 Oct30 ?        00:00:00 [migration/0]

root         8     2  0 Oct30 ?        00:00:00 [rcuc/0]

root         9     2  0 Oct30 ?        00:00:00 [rcub/0]

root        10    2  0 Oct30 ?        00:00:01 [rcu_preempt]

root        11    2  0 Oct30 ?       00:00:01 [rcuop/0]

root        12    2  0 Oct30 ?        00:00:01 [rcuop/1]

root        13    2  0 Oct30 ?        00:00:00 [rcu_bh]

root        14    2  0 Oct30 ?        00:00:00 [rcuob/0]

root        15    2  0 Oct30 ?        00:00:00 [rcuob/1]

----------------- cut ---------------------
```

Why is the USER column not displaying my username, but showing others like root, www-data etc ? For all usernames (including yours) if the length is greater than 8 characters then ps will fall back to show only the UID instead of username.

2. Display process by user

To filter the processes by the owning user use the "-u" option followed by the username. Multiple usernames can be provided separated by a comma.

```
$ ps -f -u wmorales
```

```
UID        PID PPID C STIME TTY          TIME CMD

wmorales 1256 1253 0 18:05 ?       00:00:00 sshd: wmorales@pts/5

wmorales 1257 1256 0 18:05 pts/5   00:00:00 -bash

wmorales 1600 1257 0 18:24 pts/5   00:00:00 script processexercise

wmorales 1602 1600 0 18:24 pts/5   00:00:00 script processexercise

wmorales 1603 1602 0 18:24 pts/4   00:00:00 bash -i

wmorales 1627 1603 0 18:26 pts/4   00:00:00 ps -f -u wmorales

wmorales 23848     1 0 Oct30 ?       00:00:00 /usr/lib/systemd/systemd --user

wmorales 23850 23848 0 Oct30 ?       00:00:00 (sd-pam)
```

3. Sort process by cpu or memory usage

System administrators often want to find out processes that are consuming lots of memory or CPU. The sort option will sort the process list based on a particular field or parameter.

Multiple fields can be specified with the "--sort" option separated by a comma. Additionally the fields can be prefixed with a "-" or "+" symbol indicating descending or ascending sort respectively. There are lots of parameters on which the process list can be sorted. Check the man page for the complete list.

```
$ ps aux --sort=-pcpu,+pmem

USER       PID %CPU %MEM   VSZ   RSS TTY     STAT START   TIME COMMAND

root     2668 0.3 12.0 2131004 488488 ?     Sl   Oct30   6:18 /usr/bin/gnome-

root     2760 0.1 0.4 218884 16388 ?       S    Oct30   2:22 /usr/bin/vmtool

root        2 0.0 0.0     0     0 ?       S    Oct30   0:00 [kthreadd]

root        3 0.0 0.0     0     0 ?       S    Oct30   0:00 [ksoftirqd/0]

root        5 0.0 0.0     0    0 ?       S<   Oct30   0:00 [kworker/0:0H]

root        7 0.0 0.0     0     0 ?       S    Oct30   0:00 [migration/0]

root        8 0.0 0.0     0     0 ?       S    Oct30   0:00 [rcuc/0]

root        9 0.0 0.0     0     0 ?       S    Oct30   0:00 [rcub/0]

root       10 0.0 0.0     0     0 ?       S    Oct30   0:01 [rcu_preempt]

root       11 0.0 0.0     0     0 ?       S    Oct30   0:01 [rcuop/0]

root       12 0.0 0.0     0     0 ?       S    Oct30   0:01 [rcuop/1]

root      13 0.0 0.0     0     0 ?       S    Oct30   0:00 [rcu_bh]
```

```
-------------------------        cut ------------------
```

Display the top 5 processes consuming most of the cpu.

```
$ ps aux --sort=-pcpu | head -5

USER       PID %CPU %MEM   VSZ   RSS TTY     STAT START   TIME COMMAND

root     2668 0.3 12.0 2131004 488488 ?     Sl   Oct30   6:18 /usr/bin/gnome-shell

root     2760 0.1 0.4 218884 16388 ?       S   Oct30   2:22 /usr/bin/vmtoolsd -n vmusr --blockFd 3

root         1 0.0 0.1 48356 4524 ?        Ss   Oct30   0:02 /sbin/init showopts

root         2 0.0 0.0     0     0 ?       S   Oct30   0:00 [kthreadd]
```

4. Display process hierarchy in a tree style

Many processes are actually forked out of some parent process, and knowing this parent child relationship is often helpful. The '--forest' option will construct an ascii art style tree view of the process hierarchy.

The following command will search for processes by the name apache2 and construct a tree and display detailed information.

```
$ ps -f --forest

UID       PID PPID C STIME TTY         TIME CMD

wmorales 1603 1602 0 18:24 pts/4   00:00:00 bash -i

wmorales 1677 1603 0 18:35 pts/4   00:00:00 \_ ps -f –forest
```

Try not to use any sorting with the tree style display, as they both effect the order of display in different ways.

5. Change the columns to display

The ps command can be configured to show a selected list of columns only. There are a large number of columns to to show and the full list is available in the man pages.

The following command shows only the pid, username, cpu, memory and command columns.

```
$ ps -e -o pid,uname,pcpu,pmem,comm

PID USER     %CPU %MEM COMMAND

   1 root     0.0 0.1 systemd

   2 root     0.0 0.0 kthreadd

   3 root     0.0 0.0 ksoftirqd/0
   5 root     0.0 0.0 kworker/0:0H

   7 root     0.0 0.0 migration/0

   8 root     0.0 0.0 rcuc/0

   9 root     0.0 0.0 rcub/0

   10 root     0.0 0.0 rcu_preempt

   11 root     0.0 0.0 rcuop/0

   12 root     0.0 0.0 rcuop/1

   13 root     0.0 0.0 rcu_bh

   14 root     0.0 0.0 rcuob/0

   15 root     0.0 0.0 rcuob/1

   16 root     0.0 0.0 rcu_sched

---------------------         cut ------------------------
```

It is possible to rename the column labels

```
$ ps -e -o pid,uname=USERNAME,pcpu=CPU_USAGE,pmem,comm

PID USERNAME CPU_USAGE %MEM COMMAND

   1 root           0.0 0.1 systemd

   2 root           0.0 0.0 kthreadd

   3 root           0.0 0.0 ksoftirqd/0

   5 root           0.0 0.0 kworker/0:0H

   7 root           0.0 0.0 migration/0

   8 root           0.0 0.0 rcuc/0

   9 root           0.0 0.0 rcub/0

   10 root           0.0 0.0 rcu_preempt

   11 root           0.0 0.0 rcuop/0

   12 root           0.0 0.0 rcuop/1

   13 root           0.0 0.0 rcu_bh

   14 root           0.0 0.0 rcuob/0

   15 root           0.0 0.0 rcuob/1

   16 root           0.0 0.0 rcu_sched

   17 root           0.0 0.0 rcuos/0
```

```
------   cut ------------------------
```

Quite flexible.

6. Display elapsed time of processes

The elapsed time indicates, how long the process has been running for. The column for elapsed time is not shown by default, and has to be brought in using the "-o" option

```
$ ps -e -o pid,comm,etime

PID COMMAND             ELAPSED

   1 systemd         1-11:10:28

   2 kthreadd         1-11:10:28

   3 ksoftirqd/0     1-11:10:28

   5 kworker/0:0H     1-11:10:28

   7 migration/0     1-11:10:28

   8 rcuc/0           1-11:10:28

   9 rcub/0           1-11:10:28

   10 rcu_preempt     1-11:10:28

   11 rcuop/0         1-11:10:28

   12 rcuop/1        1-11:10:28

   13 rcu_bh           1-11:10:28

   14 rcuob/0         1-11:10:28

   15 rcuob/1         1-11:10:28

   16 rcu_sched       1-11:10:28

------------------------         cut -------------------
```

7. Turn ps into an realtime process viewer

As usual, the watch command can be used to turn ps into a realtime process reporter. Simple example is like this

```
$ watch -n 1 'ps -e -o pid,uname,cmd,pmem,pcpu --sort=-pmem,-pcpu | head -15'
```

The output on my desktop is something like this.

```
Every 1.0s: ps -e -o pid,uname,cmd,pmem,pcpu --sort... Fri Oct 31 19:04:57 2014


 PID USER     CMD                             %MEM %CPU
 2668 root     /usr/bin/gnome-shell       12.0 0.3
 1572 vscan   /usr/sbin/clamd             7.6 0.0
12232 root     /usr/lib/YaST2/bin/y2base s 5.4 0.0
11973 root     gnome-control-center --over 1.1 0.0
 2803 root     /usr/lib/evolution-data-ser 1.1 0.0
 1547 root     /usr/bin/Xorg :0 -backgroun 0.9 0.0
 2710 root     /usr/lib/evolution/3.10/evo 0.7 0.0
 2593 root     /usr/lib/gnome-settings-dae 0.6 0.0
12008 root     /usr/lib/YaST2/bin/y2contro 0.5 0.0
 2571 root     /usr/lib64/ibus/ibus-ui-gtk 0.5 0.0
 2644 root     /usr/lib/goa-daemon         0.4 0.0
 2810 root     /usr/lib/libsocialweb-core   0.4 0.0
 2760 root     /usr/bin/vmtoolsd -n vmusr   0.4 0.1
 2482 root     gdm-session-worker [pam/gdm 0.3 0.0
```

The output would be updated every 1 second to refresh the stats. However do not think that this is similar to top.
You would notice that the output of top/htop command changes much more frequently compared to the above ps command.
This is because the top output sorts on a value that is a mix of cpu usage and memory usage. But the above ps command sorts in a more simpler manner, taking 1 column at a time (like school maths). So it would not update rapidly like top.