Joel McKinzie
ME-701
Fall 2015
Dr. Roberts

Homework 1

1. I enjoy using flight simulators. I have used several different flight simulators including the various versions of Microsoft Flight Simulator. In the Software Manager for Linux, I was able to find a flight simulator that may be able to be used as a casual substitute to Microsoft's paid version. Flight gear flight simulator looks to be decent as far as visuals, less detail than Microsoft's flight simulator, but I'm more concerned with realism than visuals. I would imagine, that based on the description, it would remain fairly realistic in its physics.

2. The Command Line
    a. Displaying info about the processor

```
jsmckinzie@Joel-Linux ~ $ lscpu
Architecture:              x86_64
CPU op-mode(s):            32-bit, 64-bit
Byte Order:                Little Endian
CPU(s):                    4  – This is where it lists the number of CPUs
On-line CPU(s) list:       0-3
Thread(s) per core:        2
Core(s) per socket:        2
Socket(s):                 1
NUMA node(s):              1
Vendor ID:                 GenuineIntel
CPU family:                6
Model:                     69
Stepping:                  1
CPU MHz:                   2003.906 – This is where it lists the Processor
speed
BogoMIPS:                  4788.95
Virtualization:            VT-x
L1d cache:                 32K
L1i cache:                 32K
L2 cache:                  256K
L3 cache:                  3072K
NUMA node0 CPU(s):         0-3
```

    b. Displaying info about programs that use the most memory and processing power

```
jsmckinzie@Joel-Linux ~ $ top
```

I used the “top” command to display memory and processing power of programs. You could also use “ps aux --sort -rss” to display these in order from highest to lowest memory usage. I also pasted my data into a spreadsheet because it’s easier to read as opposed to Microsoft Word’s crazy attempt at formatting.

```
top - 08:51:13 up 32 min,  2 users,  load average: 1.19, 1.14, 1.03
Tasks: 193 total,   1 running, 192 sleeping,   0 stopped,   0 zombie
```

```
%Cpu(s):  0.5 us,  0.1 sy,  0.0 ni, 99.2 id,  0.3 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem:   8056988 total,  2188400 used,  5868588 free,    57336 buffers
KiB Swap:  3906556 total,        0 used,  3906556 free.  1131440 cached Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
7	root	20	0	0	0	0	S	1	0	00:15.6	rcu_sched
1257	root	20	0	473328	119156	107536	S	1	1.5	00:19.4	Xorg
25	root	20	0	0	0	0	S	0.7	0	00:13.9	rcuos/2
2672	jsmckin+	20	0	1784432	151508	64644	S	0.7	1.9	00:42.5	cinnamon
72	root	20	0	0	0	0	S	0.3	0	00:00.1	kworker/2:2
1120	avahi	20	0	34832	5724	2824	S	0.3	0.1	00:46.7	avahi-daem+
4518	jsmckin+	20	0	612416	28980	23068	S	0.3	0.4	00:00.8	gnome-term+
4748	jsmckin+	20	0	878608	161444	64932	S	0.3	2	00:02.9	chrome
1	root	20	0	33756	4324	2684	S	0	0.1	00:01.5	init
2	root	20	0	0	0	0	S	0	0	00:00.0	kthreadd
3	root	20	0	0	0	0	S	0	0	00:00.0	ksoftirqd/0
4	root	20	0	0	0	0	S	0	0	00:00.1	kworker/0:0
5	root	0	-20	0	0	0	S	0	0	00:00.0	kworker/0:+
8	root	20	0	0	0	0	S	0	0	00:00.0	rcu_bh
9	root	20	0	0	0	0	S	0	0	00:01.2	rcuos/0
10	root	20	0	0	0	0	S	0	0	00:00.0	rcuob/0
11	root	rt	0	0	0	0	S	0	0	00:00.0	migration/0

3. Simple Shell Scripting

   a. I wrote my bash script to convert temperature from Fahrenheit to Celcius. I did not use the “let” command, I instead just used $(echo “scale=<Desired decimal places>;Math Operation|bc). My file is called Temp.sh

b. I wrote my script for file counts using $(ls -1 | wc -l)

   This takes the output of ls and decreases it by 1. The wc -l portion tells it to count by lines. I The output of this was assigned to a variable and then output using echo. The file is called FileCount.sh

c. This script was a bit trickier. I used a lot of google to figure out how to check for certain things, but essentially I have the following: I read in the file from the user. I then check if the file exists in the working directory. If it exists, I check to see if the “Trash” folder exists. If that exists I use mv to move the file to that folder. If the “Trash” folder does not exist, it gets trickier. My else uses mkdir -p to create the directory. -p was required to allow mkdir to create the folder. Then I had to use the chmod command to give write permissions in that directory. I then move the file using mv. If the file did not exist, I output a message indicating that to the user. The file is called Trash.sh

4. Git and Version Control
   a. Link to GitHub: https://github.com/jsmckinzie/Homework_1
   b. I modified my temperature conversion program and the updated version is located in the repository above.