CS105 Program Assignment #3
Custom ps

Erik Steggall, TA
esteggall@soe.ucsc.edu
Due: 23:59, Wednesday October 23, 2013

Fall 2013

1 Goals

This assignment is designed to introduce you to procfs and understand how to use it for practical purposes. For this assignment you will be duplicating the main functionality of the ps command.

2 Assignment Specifications

The bare *ps* command will return the current processes that you are running, along with the PID of that process.

```
>  ps
    PID TTY          TIME CMD
22403 pts/2    00:00:00 bash
22454 pts/2    00:00:00 ps
```

The *ps -e* will run for all processes on the system (including processes being run by other users).

```
ps -e
  PID TTY          TIME CMD
    1 ?        00:00:06 init
    2 ?        00:00:00 kthreadd
    3 ?        00:21:26 ksoftirqd/0
    6 ?        00:00:01 migration/0
    7 ?        00:00:10 watchdog/0
    8 ?        00:00:01 migration/1
...
```

The *ps -l* will list more detailed information about each process.

```
> ps -l
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000 22403 22402  0  80   0 -  5940 wait   pts/2    00:00:00 bash
0 R  1000 22929 22403  0  80   0 -  2432 -      pts/2    00:00:00 ps
```

The *ps -el* puts the two options together.

3 Grading

You will receive $\frac{1}{4}$ credit for implementing the bare *ps* command, $\frac{1}{4}$ credit for implementing *ps -e*, $\frac{1}{4}$ credit for implementing *ps -l*, and $\frac{1}{4}$ credit for implementing *ps -le*.

4 Hints

- All of the information you need is inside /proc. Check out procfs (click here) for more details.
- For the -e option you will need to access other users process information. You'll need root privileges for this, use setuid().
- As always, break this assignment into smaller pieces and test each piece seperately before adding them to your final draft.
- If you need help, come to section (W/F 12:30-1:40) or email esteggall@soe.ucsc.edu.