

**CS240 Operating Systems**  
**Coversheet**  
**Assignment 1**

**Name:** \_\_\_\_\_

**Number of cores:** Your computer \_\_\_\_\_ CS-240a/b \_\_\_\_\_

**Due date:** 9/16/2019.

**Deliverables:** use this page as cover sheet -- for all else see below.

**Objective:** Investigation of processes and the resources they use. Our focus is on multi-core processors. Usage of *ps* and *top*.

**Preparation:** We want to investigate the impact of running CPU and I/O intensive jobs on multi-core processors. For this **use your** own computer and only then the CS computers – [cs-240a.cs.uidaho.edu](http://cs-240a.cs.uidaho.edu) or [cs-240b.cs.uidaho.edu](http://cs-240b.cs.uidaho.edu). If you don't own a computer, you can use any other computer on campus (running linux).

Write two small C programs that contain loops and run for 5 minutes. The first program, called *CPU-bound.c*, should just perform simple arithmetic calculations, e.g. a counter. The second program, called *IO-bound.c*, should perform an I/O operation, e.g., open a file, write a character into the file and close it. Again, this should be done in a loop that will terminate after 5 minutes. Make sure the programs really terminate as we want to avoid having endless processes hogging resources.

Read about *top* and *ps* and check out what information they provide, e.g., use the *man* pages to see what options are available.

**Experiment 1:** (using your computer)

1) Run *top* and see what processes the system is using. Try to understand what processes are running and interpret the information displayed for each process, i.e. interpret the columns.

2) Now start executing a single instance of your *CPU-bound* process in the background. Recall that a process can be executed in the background by using the "&", e.g. *CPU-bound* &. Find out how to start a process, stop it, and start it again, run it in the foreground and send it to the background.

3) Next investigate the impact of the *CPU-bound* program using *ps* and *top*.

4) Now start the *IO-bound* in the background and again check using *ps* and *top* for its impact. At this point you should have a single instance of both programs executing. What do you notice about the processes' CPU utilization?

5) Repeat the experiments with multiple instances of the processes:

1. Start multiple instances of your *CPU-bound* program to get the cores working.
  - a. What do you notice now about your *CPU-bound* processes?
  - b. How many instances of your CPU program did you run and what CPU utilization did you observe?
2. Start multiple instances of your *UI-bound* program.
  - a. What do you notice now about your mix of processes?
  - b. How many instances of your IO program should you (did you) run and what CPU utilization did you observe?

6) Once you are sure you can document your findings, **kill the processes**. How do you find out which processes you should kill? By now you should have figured out how to get the process ID (e.g., using *top*, *ps*) and use the man pages on the *kill* command.

**IMPORTANT:** Verify that you actually killed your programs!

**Experiment 2: Do this only after you are done with Experiment 1, unless you do not own a computer. Gather the information from the experiment on the CS computers.**

Document differences between the two systems, i.e., did you observe any measurable differences between the CPU and I/O utilization between the target systems?

**Deliverables:** You need to turn in:

- 1) **All source code** of your programs and a **working Makefile** to build your programs. The source code and the Makefile must be bundled together in **one .tar file**. This means you should use the unix *tar* command to create the tar file named **Asgn1.tar**. This tar file is what you need to turn in using *cscheckin*. Since you are turning in two programs, your Makefile must contain two targets: one named CPU-bound and one named UI-bound. Your Makefile must provide all necessary command-line arguments to build your programs from source.
- 2) A **hardcopy** of 1) the assignment sheet as cover page, 2) your source code and 3) the report that shows and explains your findings, i.e. the impact of your executions and how they differ (or not) between the two target computers. Make sure you include screen shots of your programs in execution using *ps* and *top*. Your documentation should **identify in the screen shots where the values of interest are**, e.g., by highlighting, underlining. Make sure you clearly identify the supporting evidence. This report is a required deliverable and is very important.
- 3) **AGAIN: Use this assignment sheet as the cover-sheet for the hardcopy!**