CS 377: OPERATING SYSTEMS LAB
Assignment 01

9 *January* 2015

This assignment is based on the PROC file system of Linux (in which the OS writes various kinds of information concerning executing processes), process states, interrupts and traps.

Question 1 [10 marks]

Answer the following questions using the proc file system. Also mention the name of the file of proc file system which contains the answer.

a. What is the cache size, address sizes (physical and logical) of any processor?
b. What's the family and model of the processor? How many cores does the processor have?
c. What is the limit on the number of open files of a process? (*Hint:* Execute the program ques4.cpp to find out)
d. Write a 8-10 line paragraph in your own words explaining process-preemption. (Read linux literature to find relevant information)

Question 2 [15 marks]

A process executes in the ***user mode*** most of the time and in the ***kernel mode*** some of the time. A simple explanation of these modes is as follows: A ***privileged instruction*** is an instruction in the CPU which a user process is not allowed to execute. If a process contains some privileged instructions, the kernel executes them on behalf of the process. During this time, the process is said to be in kernel mode. The process also executes in the kernel mode when it makes a *system call.*

Consider the given two programs ques2_1.cpp and ques2_2.cpp. Compile and execute the programs using following commands:

```
g++ ques2_1.cpp -o first
g++ ques2_2.cpp -lm -o second
./first > /dev/null
./second > /dev/null
```

Use proc filesystem to find out the time spent by each of the above two processes in kernel mode and user mode by running both the processes for 30 seconds. Observe the values of the times spent in the kernel and user modes. Study the programs and give a logical explanation about the relative magnitudes of these times.

P.S.: End the process's by (Ctrl+C) after completion.

Question 3 [**20 marks**]

The operating system uses the notion of a state to represent the current activity in a process and changes the state as the activity in the process changes.

You are given a program ques3.cpp. The process that represents its execu- tion spawns three child processes by using the fork command during its execution. The process that spawns other processes is called the parent process. You will notice that the three child processes execute different functions in ques3.cpp.

a. From Linux literature, find the meaning of the various states.
b. Now study the program ques3.cpp and code the functions executed by the three child processes such that one of them will be in zombie state, one will be in sleep state and the third will be in some other state. Submit your program.
c. How can you get your process in traced or stopped (T) state?

Question 4 [**15 marks**]

A ***context switch*** occurs when the OS switches the CPU from execution of one process to execution of another process. A *voluntary context switch* occurs when a process voluntarily yields the CPU (by explicitly calling sleep, or waiting for an event or some resource to become available); whereas an *involuntary context switch* occurs when the OS scheduler decides to execute some other process.

You have been given a program ques4.cpp. Compile the same program twice to produce two executables.

```
g++ ques4.cpp -o first
g++ ques4.cpp -o second
```

Execute these programs with different scheduling priorities as follows.

```
nice -n +5 ./first output_file1
nice -n +11 ./second output_file2
```

a. Count the no of **a**'s in both the files after 30 seconds.
b. Observe the CPU percentage consumed by these processes.
c. Observe the values of voluntary and involuntary context switches. (You can get them from status file of the proc filesystem.)

Change the priority of process **first** as `renice +20 <process PID>`

Answer questions (a)–(c) for the new priorities.

Question 5 [15 marks]

Use the proc file system to find the current load average of the system. Now run the program ques5.c. Use the proc file system to again find the current load average of the system.

Read about load averages. Analyze and reason the difference between the load averages.

Modify program ques5.c so that you obtain a significant difference in load averages. Explain how you achieved it. Also submit the modified program.

Question 6 [25 marks]

You are given an executable ques6.out, which runs a loop, producing a certain interrupt signal every iteration. You have to figure out which interrupt is being raised.

a. Run the executable. Use the proc file system to find the signal number corresponding to the interrupt signal being raised.
b. Which is the interrupt signal corresponding to the same?
c. Explain exactly which file(s) you looked into and how you retrieved it.
d. Can you also find the number of iterations executed? (If yes, explain how)