Name: **Student Id:**

CMPSCI 377: Operating Systems

Exam 1: Processes, Threads, Scheduling, and Synchronization

October 4, 2016

General instructions:

- This examination booklet has 10 pages.
- Do not forget to put down your name and student number on the exam books.
- The exam is closed book and closed notes.
- Explain your answers clearly and be concise. Do not write long essays.
- You have 75 minutes to complete the exam. Be a smart test taker, if you get stuck on one problem go on to the next. Don't waste your time giving details that the question does not request.
- Show your work. Partial credit is possible, but only if you show intermediate steps.
- Good luck.

---

1. Answer the following questions in brief. (22 pts)

   (a) (4pts) Why is it desirable to overlap I/O and computation on a machine?

(b) (4pts) What is Moore's law? Explain how Moore's law has led to improvements in computing technology over the past several decades.

(c) (5pts) Explain with an example why the OS provides an easier to use virtual machine interface to the user.

(d) (5pts) Draw a picture of the memory hierarchy seen by the CPU.

(e) (4pts) List two operations that may only be performed when the CPU is in kernel mode and explain why in one sentence each.

2. **OS Architecture and System Calls** (27 pts)

(a) (15pts) Consider the following code for the *fork()* system call.

```
main (){
 int i, j;

 i = fork();
 j = fork();

 if (i == 0 OR j == 0)
    printf(" Hello World");
 if ( i==0 AND j == 0)
    printf("Bye World");
}
```

(1) How many processes are created by the above program? Show the parent-child relationship between each process and list the values of i and j within in each process.

(2) What output does the above program produce? Any correct output regardless of order of execution of each process is acceptable.

(b) (5pts) What does the timer instruction of the CPU do? Explain one possible use of the timer instruction by the OS.

(c) (7pts) Briefly explain the modular architecture for designing OS kernels. Explain how the modular approach retains some of advantages of the micro-kernel approach while overcoming its limitations.

3. **Process Management** (27 pts)

(a) (5pts) What information is maintained by the kernel inside the process control block?

(b) (5pts) Classify each of the following scheduling algorithms as preemptive or non-preemptive: FCFS, round-robin, shortest job first, lottery scheduling, MLFQ.

(c) (7pts) Explain how MLFQ approximates the behavior of the SJF scheduler. Also explain how it overcomes the primary limitation of SJF.

(d) (10pts) Compute the completion times and waiting times for the following set of jobs. Assume a zero context switch overhead, a time slice of 1 second, and that all jobs are compute bound.

			Completion Time			**Wait Time**		
Job	length	arrival time	FCFS	RR	SRTF	FCFS	RR	SRTF
1	50	0						
2	40	0						
3	30	0						
4	20	0						
5	10	8						

4. **Threads and synchronization** (25pts)

(a) (6pts) What steps must be performed for context switching between threads of the same process? What additional steps are necessary when switching between threads of different processes?

(b) (15pts) Write psuedo-code for implementing the Acquire and Release methods of a Lock using the *test-and-set* instruction. Comment on how much busy-waiting is involved in your solution and describe in brief how the OS can reduce this busy waiting.

(overflow sheet for your answers)