On my honor, I have not given, nor received, nor witnessed any unauthorized assistance on this work.

Print name and sign: ______________________________

Question:	1	2	3	4	5	Total
Points:	4	4	6	8	8	30
Score:						

1. (4 points) You write a Unix shell, but instead of calling `fork` and then `exec` to create a new process, you make a subtle mistake: you first call `exec` and then `fork`. How does this change the functioning of your shell (if it does). Explain your answer.

2. (4 points) Assume we have three jobs which arrive one after the other at time 0 in the following order:

   - Job A which needs 10 seconds of CPU time
   - Job B which needs 15 seconds of CPU time
   - Job C which needs 10 seconds of CPU time

   Assuming a SJF policy, at what time does B finish?

3. When a system call occurs, the hardware will redirect execution to special trap handler code in the OS. The OS must set up a trap table to inform the hardware of the location of all interrupt-handling routines.

   (a) (3 points) Explain when and how the trap table is initialized and who is responsible for the initialization.

   (b) (3 points) When a user process executes a system call, it uses a a special trap machine language instruction (`syscall` on the x86-64 architecture, `trap` on the ARM8). What does this instruction do?

4. Below is a drawing showing a **round robin** scheduler with a 2 second time-slice. Three jobs: A, B, and C arrive in order, at time 0. Each of these jobs run for 6 time units. The scheduler itself takes 1 second to change jobs. This leads to a diagram:

```
SAA SBB SCC SAA SBB SCC SAA SBB SCC
012 345   8   1   1   1   2   2   2
              1   4   7   0   3   6
```

   (a) (4 points) Calculate the average **response time** for this round robin implementation for jobs A, B, and C.

   (b) (4 points) Calculate the average **turnaround** time for this round robin implementation for jobs A, B, and C.

5. Scheduling algorithms can be classified along two axes: preemptive versus non-preemptive and size-based versus non-size-based.

   (a) (4 points) Explain the the terms *preemptive* and *size-based* in the context of scheduling algorithms.

   (b) (4 points) Fill in the table below, placing the five scheduling algorithms you learned about in this sprint (FIFO, SJF, STCF, RR, and MLFQ) into the appropriate quadrant.

	size-based	non-size-based
preemptive		
non-preemptive		