IT5002 Computer Systems and Applications
Tutorial 7

1. How many processes are created in the following program? Include the original process created when the program is first run.

```
for(int i=0; i<10; i++) fork();
```

2. Consider a process P1. What state would P1 be in, in the following circumstances?

Situation	**P1's State**
The OS has just finished creating P1	
P1 is pre-empted by another process.	
P1 is waiting for I/O	
P1's I/O has just completed	
P1 is currently being executed	
P1 is suspended for 2 seconds	

3. Suppose we have a fixed-priority operating system, where each process is given a priority, and when a higher priority process is ready, the OS pre-empts the current process and passes CPU control to the higher priority process.

   a. If there are 255 possible priority levels, how many ways can you assign priorities to *n* processes, if no two processes can have the same priority level?

   b. We consider *periodic processes*. A periodic process is one that runs for C_i CPU cycles every P_i cycles.

      Assuming that a process must finish running before the start of its next period, an intuitive way to assign priorities is by order of process periods; a process with a shorter period would be assigned a higher priority than one with a longer period; this is called *rate monotonic scheduling* or RMS.

So process P_1 might run for 1 cycle every 4 cycles, P_2 might run for 2 cycles every 6 CPU cycles, etc, giving a schedule that looks like this.

Clock Cycle	Process	Ready to Run
0	P1	P1, P2
1	P2	P1 runs first
2	P2	
3		
4	P1	P1
5		
6	P2	P2
7	P2	
8	P1	P1
9		
10		
11		
12	P1	P1, P2
13	P2	P2 delayed by one cycle because of P1
14	P2	
15		
16	P1	
…	…	

Suppose we sort our processes in ascending order of period, so that $\Pr(T_0) > PR(T_1) > PR(T_2) > \cdots > PR(T_n)$, where $\Pr(A) > \Pr(B)$ means that process Consider this algorithm:

1. Sort T by period of each task, if T is not already sorted. We will assume that T_1 has the shortest period, T_2 has the 2nd shortest, etc.
2. For each task $T_i \in T$, recursively compute $S_{i0}, S_{i1}, \ldots$ where:
   a. $S_{i,0} = \sum_{j=1}^{i} C_j$
   b. $S_{i,(x+1)} = C_i + \sum_{j=1}^{i-1} C_j \times \left\lceil \frac{S_{i,x}}{P_j} \right\rceil$

   Stop when $S_{i,(x+1)} = S_{i,x}$. Call this $S_{i,(x+1)}$ the final value $S_{i,F}$. I.e. let $S_{i,F} = S_{i,(x+1)}$ when the iteration terminates.

Prove that $S_{i,F}$ is the worst-case response time for process P_i. The response time is the time between the first time a process runs, and when it finishes running.

4. Our system has one CPU and one I/O device.

   We have a set of 3 processes P1, P2 and P3 where P1 has the highest priority and P3 the lowest. The “execution profile” column is in the format Cw-IOx-Cy-IOz which means that the process runs for “w” cycles, then accesses IO for “x” cycles, then runs on the CPU for “y” cycles, and again performs IO for “z” cycles. All 3 processes are ready to run at time 0.

   When the I/O device is busy the next process will wait for it to be free before using it. If two processes are waiting for I/O, the higher priority process uses it first. However there is no pre-emption for I/O.

Process	Execution Profile
P1	C2-IO3-C3-IO1-C2-IO3-C3
P2	C1-IO2-C1
P3	C3

   a. Show the schedule for the three processes.

   b. What is the response time for each process?

   c. What is the CPU utilization? State your answer to two decimal places.