

# CSE 330 Midterm

Olcay Taner YILDIZ

## I. QUESTION (12 POINTS)

Draw the process state diagram and explain states of the process (Example: running) briefly.

## II. QUESTION (18 POINTS)

- Two benefits of multithreaded programming over single-threaded programming are economy and utilization of multiprocessor architectures. Explain these benefits briefly.
- Given disadvantages of many-to-one and one-to-one multithreading models (one disadvantage per model is enough). How does many-to-many model overcome these disadvantages?

## III. QUESTION (20 POINTS)

Given the following arrival time, burst time and priority (1 is the highest priority) information of five processes  $P_1$  to  $P_5$ . Show the Gantt Chart and calculate the average waiting time of the processes for FCFS, preemptive SJF, preemptive priority and round robin scheduling (quantum = 1) algorithms.

Process	Arrival Time	Burst Time	Priority
$P_1$	0	8	5
$P_2$	1	3	3
$P_3$	2	2	2
$P_4$	4	1	4
$P_5$	10	4	1

## IV. QUESTION (12 POINTS)

The next CPU burst is generally predicted as an exponential average of the measured lengths of previous CPU bursts. Let  $t_n$  be the length of the  $n$ th CPU burst, and let  $\tau_{n+1}$  be our predicted value for the next CPU burst. Then, for  $\alpha$ ,  $0 \leq \alpha \leq 1$ , exponential average is defined as

$$\tau_{n+1} = \alpha t_n + (1 - \alpha)\tau_n \quad (1)$$

Given the CPU bursts  $t_0 = 8$ ,  $t_1 = 3$ ,  $t_2 = 2$ ,  $t_3 = 1$ ,  $t_4 = 4$  and  $\alpha = 0.5$ , calculate the predicted values for the next CPU bursts  $\tau_1, \dots, \tau_5$ . Assume  $\tau_0 = 4$ .

## V. QUESTION (20 POINTS)

Given the following C program, draw the process tree.

```
void main(void)
{
    int pid, pid1, pid2, pid3;
    int pid4, pid5, pid6;
    pid = fork();
    if (pid == 0){
        pid2 = fork();
        if (pid2 > 0){
            pid4 = fork();
            if (pid4 > 0)
                pid5 = fork();
        }
    }
    if (pid > 0){
        pid1 = fork();
        if (pid1 == 0){
            pid3 = fork();
            if (pid3 > 0)
                pid6 = fork();
        }
    }
}
```

## VI. QUESTION (18 POINTS)

- Give an example situation where wrong usage of semaphores can result in a deadlock (indefinite blocking).
- Given three processes, ( $P_1, P_2, P_3$ ) and three statements ( $S_1, S_2, S_3$ ), where  $P_i$  will execute  $S_i$ , use semaphores to make the order of execution  $S_1, S_3, S_2$ . Use at most three semaphores. Show the initial semaphore values.