CS330: Operating Systems
Quiz#3

Name:
Roll No.:

1. Consider a process P that has a single CPU burst and no I/O bursts. It is observed that P has a waiting time of 20 units in the ready queue and a turn-around time of 50 units for a certain scheduling algorithm A. When some other scheduling algorithm A' is used, the waiting time of P increases to 30 units. What is the turn-around time of P when A' is used? Ignore context switch, mode switch, timer interrupt handler, and scheduling algorithm's run-time overheads. **(1 point)**

Solution: In the absence of I/O, turn-around time = waiting time in ready queue + sum of CPU bursts. Therefore, sum of CPU bursts $= 50 - 20 = 30$ units. The sum of CPU bursts is a property of a program and is independent of the scheduling algorithm. Therefore, new turn-around time $= 30 + 30 = 60$ units.

Grading policy: No partial credit.

2. Suppose four processes A, B, C, D have priorities 10, 20, 30, 40. They have infinite run-time (i.e., they run forever) and have no I/O. Consider the proportional share-based stride scheduling algorithm with a fixed time quantum i.e., the time quantum does not depend on the priorities. Show the order in which the processes would be scheduled to maintain proportional share (you need to show the schedule for the first fifteen quanta). Assume that any kind of scheduling tie is broken in the priority order e.g., the first four processes to get scheduled are D, C, B, A in that order. **(4 points)**

Solution: The strides are 12, 6, 4, 3 respectively for A, B, C, D. Please refer to lecture slides on stride scheduling to verify that the schedule is D, C, B, A, D, C, D, B, C, D, D, C, B, A, D, ... where ties are broken in priority order.

Grading policy: One mark for correct calculation of strides. Three marks for correct calculation of schedule.

3. Consider two processes with id 0 and 1. The processes execute the following code segments where the array A and the variables x and y are shared between the processes. All elements of the array A are initialized to zero. The variables x and y are initialized to -1 and 10 respectively. What are the possible values of the variable y after both processes complete execution? **(5 points)**

```
Process 0                           Process 1
-----------                         ------------
x = 1;                              x = 0;
A[0] = 1;                           A[1] = 1;
while ((x == 1) && A[1]);           while ((x == 0) && A[0]);
y++;                                y += 2;
A[0] = 0;                           A[1] = 0;
```

Solution: If we consider the variable x to represent `turn` and the array A to represent the `flag` array, the given program segment is a buggy version of Peterson's algorithm where the order of the first two lines is reversed (discussed in class). Therefore, the critical section where `y` is updated is not atomic. The result of P0's computation i.e., `y`=11 and the result of P1's computation i.e., `y`=12 starting from the initial value of 10 are both possible. The result of one process's computation starting from the result of the other process is also possible i.e., `y`=13.

Grading policy: Two marks each for 11 and 12, while one mark for 13.