CMSC 312 Spring 2020 Assignment #2

Due Date: Sunday 3/1/2020 (Hard deadline)

1) Implement the counting semaphores using binary semaphores. Pseudo-code for the implementation is given in the attached notes on BB: 10x-countingSemUsingBinarySem.pdf
   Use the bounded-buffer producer-consumer problem as uploaded on BB (first code at https://jlmedina123.wordpress.com/2014/04/08/255/). Show the output of this code for BOTH the incorrect and correct implementations. Also, attach your codes (two versions: for incorrect and correct solutions) on BB. (10+10 points)

2) Prove/disprove that SJF is optimal in terms of average turnaround times of the processes. (2 points)

3) For what types of workloads does SJF deliver the same turnaround times as FIFO? (2 points)

4) For what types of workloads and quantum lengths does SJF deliver the same response times as RR? (2 points)

5) What happens to response time with SJF as job lengths increase? (2 points)

6) What happens to response time with RR as quantum lengths increase? Explain in words and write an equation that gives the worst-case response time, given N jobs. (2 points)

7) "Preemptive schedulers are always less efficient than non-preemptive schedulers." Explain how this statement can be true if you define efficiency one way and how it is false if you define efficiency another way. (2 points)

8) What is the priority inversion problem? Does this problem occur if round-robin scheduling is used instead of priority scheduling? Discuss. (2 points)

9) Does Peterson's solution to the mutual exclusion problem work when process scheduling is preemptive? How about when it is non-preemptive? (2 points)

10) Consider the following set of processes, with the length of the CPU burst time given in milliseconds:

Process	Burst-Time	Priority
P1	6	3
P2	1	1
P3	2	5
P4	3	4
P5	5	2

The processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0.

(a) Draw four Gantt charts illustrating the execution of these processes using FCFS, SJF, a nonpreemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1) scheduling.

(b) What is the turnaround time of each process for each of the scheduling algorithms in part (a)?

(c) What is the waiting time of each process for each of the scheduling algorithm s in part (a)?

(d) Which of the schedules in part a results in the minimal average waiting time (over all processes)? (5 points)

11) The aging algorithm with a = 1/2 is being used to predict run times. The previous four runs, from oldest to most recent, are 40, 20, 40, and 15 msec. What is the prediction of the next time? (2 points)

12) Explain what a multi-level feedback scheduler is and why it approximates SRTF.

True or **False** (also give an explanation for your choice): If a user knows the length of a CPU time-slice and can determine precisely how long his process has been running, then he can cheat a multi-level feedback scheduler. (2 points)

13) Consider a system consisting of processes P_1, P_2, ..., P_n, each of which has a unique priority number. Write the pseudo-code of a monitor that allocates three identical line printers to these processes, using the priority numbers for deciding the order of allocation. (5 points)

Start with the following and populate the two functions request_printer() and release_printer():

```
monitor printers {
int num_avail = 3;
int waiting_processes[MAX PROCS];
int num_waiting;
condition c;
```

```
void request_printer(int proc_number) {


}

void release_printer() {

}
}
```

Late Policy:

This assignment has a hard deadline and no extensions will be granted.