Quiz 2 **MET CS575 Graduate Operating Systems**

Name: Amal Krishna Radhakrishnan **Grade: / 15**

1. Consider a logical address space of 256 pages with a 4-KB page size, mapped onto a physical memory of 64 frames. (2 points)

    a. How many bits are required in the logical address (show work)?

        Ans - 256 Pages = 2^8 pages, 12 + 8 = 20 bits

    b. How many bits are required in the physical address (show work)?

        Ans - 64 Frames = 2^6 Frames, 12 + 6 = 18 bits

2. Explain why implementing synchronization primitives by disabling interrupts is not appropriate in a single-processor system if the synchronization primitives are to be used in user level programs. (1 point)

Ans - If a user-level program is given the ability to disable interrupts, then it can disable the timer interrupt and prevent context switching from taking place, thereby allowing it to use the processor without letting other processes to execute.

3. Assume that a system has multiple processing cores. For each of the following scenarios, describe which is a better locking mechanism — a Spinlock or a Mutex lock where waiting processes sleep while waiting for the lock to become available. (3 points)

Circle the Spinlock or the Mutex lock for each scenario shown below:
• The lock is to be held for a short duration: Spinlock or Mutex lock
• The lock is to be held for a long duration: Spinlock or Mutex lock
• The thread may be put to sleep while holding the lock: Spinlock or Mutex lock

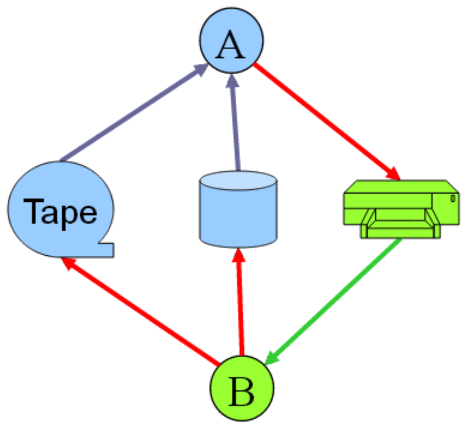
Ans –

The lock is to be held for a short duration - It makes more sense to use a spinlock as it may in fact be faster than using a mutex lock which requires suspending –and awakening - the waiting process.

The lock is to be held for a long duration - a mutex lock is preferable as this allows the other processing core to schedule another process while the locked process waits.

The thread may be put to sleep while holding the lock - a mutex lock is definitely preferable as you wouldn't want the waiting process to be spinning while waiting for the other process to wake up.

4. A utility program performs the following operations: copy a file from tape to disk and print the file on printer. The figure shows resource allocation graph for Processes A and B, and Resources Tape, Disk, and Printer.



a. Is there a potential deadlock? Yes or No. Please explain. (2 points)

Ans - Yes

- A holds tape and disk, then requests for a printer
- B holds printer, then requests for tape and disk

b. Using the symbols in the form of P->R and R->P, list all of the resource allocations and requests shown in the figure. Hint: there are six. (2 points)

A <- Tape (hold)

A <- Disk (hold)

A -> Printer (request)

B -> Tape (request)

B -> Disk (request)

B <- Printer (hold)

5. CPU Scheduling using the non-preemptive SJF scheduling algorithms (Shortest-Job-First Scheduling).

a. Draw the Gantt Charts to illustrate detail execution of the processes using the algorithm mentioned, showing process's name, and arrival/start time according to its burst length. All numbers shown are in milliseconds. (3 points)

Process	Arrive time	CPU burst time
P1	2	5
P2	0	4
P3	2	1
P4	6	1
P5	5	6

0 – P2 – 2 – P3 – 3 – P2 – 5 – P1 – 6 – P4 – 7 – P1 – 11 – P5

Process	Arrival	Burst Time	Start	Wait	Finish	TA
1	2	5	5	4	11	9
2	0	4	0	1	5	5
3	2	1	2	0	3	1
4	6	1	6	0	7	1
5	5	6	11	6	17	12

b. What is the turnaround time of each process? (2 points)
Hint: Use the formula turnaround Time = finishTime - arrivalTime

P1= 11 – 2 = 9

P2= 5 – 0 =5

P3= 3 – 2 = 1

P4= 7 - 6 = 1

P5= 17 – 5 = 12