Motilal Nehru National Institute of Technology Allahabad
Department of Computer Science & Engineering
End Term Examination 2018-19
Operating Systems (CS 33101), MCA – 3rd Semester

Duration: 3 hours **Max. Marks: 60**

Attempt all questions. Assume if something missing.

1. (a) What are the steps performed by an operating system to create a new process? **(3)**

   (b) Consider a logical address space of 64 pages of 1,024 words each, mapped onto a physical memory of 32 frames. **(3)**

   i. How many bits are there in the logical address?

   ii. How many bits are there in the physical address?

2. (a) What is Belady's anomaly? Show that a page replacement algorithm that possesses the stack property cannot exhibit Belady's anomaly. **(6)**

   (b) A time-sharing system uses swapping as the fundamental memory management technique. It uses the following lists to govern its actions: a scheduling list, a swapped-out list containing processes that are swapped out, a being swapped-out list containing processes to be swapped out, and a being-swapped-in list containing processes to be swapped in. Explain when and why the time-sharing kernel should put processes in the being-swapped-out and being-swapped-in lists. **(6)**

3. We wish to schedule three processes P1, P2 and P3 on a uniprocessor system. The priorities, CPU time requirements and arrival times of the processes are as shown below.

Process	Priority	CPU time required	Arrival time (hh:mm:ss)
P1	10(highest)	20 sec	00:00:05
P2	9	10 sec	00:00:03
P3	8 (lowest)	15 sec	00:00:00

We have a choice of preemptive or non-preemptive scheduling. In preemptive scheduling, a late-arriving higher priority process can preempt a currently running process with lower priority. In non-preemptive scheduling, a late-arriving higher priority process must wait for the currently executing process to complete before it can be scheduled on the processor. Answer the followings: **(6)**

   a. What are the turnaround times (time from arrival till completion) of P2 using preemptive and non-preemptive scheduling respectively.

   b. Compute the average waiting time and average throughput of the system using preemptive and non-preemptive scheduling respectively

4. The first known correct software solution to the critical-section problem for two processes was developed by Dekker. The two processes, P0 and P1, share the following variables:

Scanned by CamScanner

```
boolean flag[2];        /* initially false */
int turn;
do {
flag[i] = TRUE;
while (flag[j]) {
if (turn == j) {
flag [i] = false;
while (turn == j)
; // do nothing
flag [i] = TRUE;
}
}
// critical section
turn=j;
flag [i] = FALSE;
// remainder section
} while (TRUE);
```

Figure 1: The structure of process P_i in Dekker's algorithm.

The structure of process P_i (i == 0 or 1) is shown in above Figure 1; the other process is P1 (j == 1 or 0). Prove that the algorithm satisfies all three requirements for the critical-section problem. **(6)**

5. Consider the traffic deadlock depicted in Figure 2 given below: **(6)**

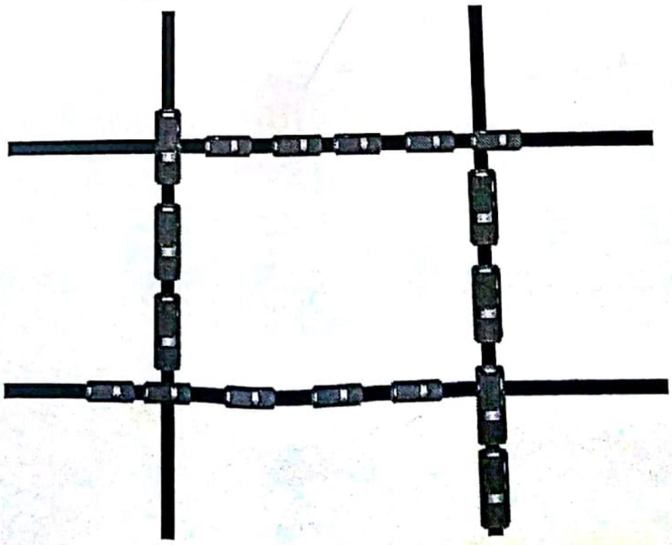

Figure 2: Traffic Deadlock Scenario

Scanned by CamScanner

a. Show that the four necessary conditions for deadlock hold in this example.
b. State a simple rule for avoiding deadlocks in this system.

6. What kind of hardware support operating system need to implement translation look-aside buffer (TLB)? Describe the inverted page table arrangement to handle the TLB? Is it possible to increase TLB size of a computer by upgrading or updating the OS? **(6)**

7. Consider a system with a two-level paging scheme in which a regular memory access takes 150 nanoseconds, and servicing a page fault takes 8 milliseconds. An average instruction takes 100 nanoseconds of CPU time, and two memory accesses. The TLB hit ratio is 90%, and the page fault rate is one in every 10,000 instructions. What is the effective average instruction execution time? (6)

8. Suppose that a disk drive has 5,000 cylinders, numbered 0 to 4999. The drive is currently serving a request at cylinder 192, and the previous request was at cylinder 115. The queue of pending requests, in FIFO order, is:
86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130
Starting from the current head position, how many times will the head change its direction to satisfy all the pending requests for each of the following disk-scheduling algorithms? **(6)**
a. SSTF
b. C-SCAN
c. LOOK

9. A simplified view of thread states is Ready, Running, and Blocked, where a thread is either ready and waiting to be scheduled, is running on the processor, or is blocked (i.e. is waiting for I/0.) This is illustrated in Figure 3. **(6)**

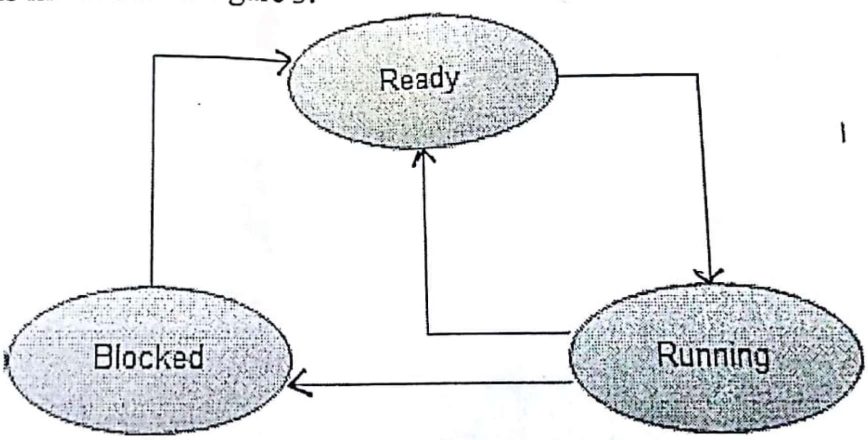

Figure 3: Thread state Diagram

Assuming a thread is in the Running state, answer the following questions: (Be sure to explain your answer.)

a. Will the thread change state if it incurs a page fault? If so, to what new state?
b. Will the thread change state if it generates a TLB miss that is resolved in the page table? If so, to what new state?
c. Will the thread change state if an address reference is resolved in the page table? If so, to what new state?

Scanned by CamScanner