Department of Computer Science & Engineering
Indian Institute of Technology Kharagpur
End-semester Examination, Spring 2016
CS30002: Operating Systems

Full Marks: 60 **Time: 3 Hours**

Instructions: Answer all the questions. Answers must be brief and to the point. Take and state suitable assumptions, if necessary

1. (a) Consider a memory management unit implementing multilevel paging. Assuming a page size of 4 Kbytes and that a page table entry takes 4 bytes, how many levels of page table would be required to map a 64 bit address space, if the top level (outermost) page table fits into a single page?
(b) "LRU page replacement algorithm is free from Belady's anomaly" – Justify
(c) A machine has a 32 bit logical address space and 8KB page size. The page table is entirely in hardware, with one 32 bit word per entry. Once a process starts, the page table is copied to the hardware from memory, at one word every 100 nsec. If each process runs for 100 msec (including the time to load the page table), what fraction of the CPU time is devoted in loading the page tables.
(d) Consider a 1-level paging system with a TLB access time of 10 nanonseconds and memory access time of 50 nanoseconds. The page fault handling time is 1 milliseconds. If the TLB hit rate is 0.99, what can be the maximum page fault rate such that the effective memory access time is at most 100 nanoseconds?

[2.5+2.5+2.5+2.5]

2. (a) A virtual memory implements "LRU approximation algorithm" with the help of reference bit. Consider a small process P1 of size four pages. At the first clock tick, the Reference bits are 0111 (page 0 is 0, the rest are 1). At the subsequent clock ticks, the values are 1011, 1010, 1101, 0010. If the page replacement algorithm is used with an 8 bit counter, give the values of the four counters after the last tick.
(b) Explain clearly how does a page fault handler distinguish between the page fault and illegal memory reference?
(c) Can a local page replacement policy applied to the page frames allocated to a process P affect the performance of any other process in the system? Justify your answer clearly.
(d) What is an inverted page table? Will two processes be able to share a page in a system using an inverted page table for address mapping? Propose a solution.

[3+2+2+3]

3. (a) Consider the organization of a Unix file system represented by the inode. Assume there are direct block pointers and a singly, doubly and triply indirect pointer in each inode. Disk block size is of 8KB. Consider that this file system has been created in a partition of a disk where each seek takes 20msec per track movement, rotation speed is 3600rpm and each track holds 1 MB of data. The inodes are stored in track 0 and it first gets loaded in the memory before any data block access.
(i) If the disk block pointer is 32 bits what is the maximum size file supported by this system? Considering that inode is already in the main memory, how many disk accesses are required to access the byte in position 13,423,956?
(ii) In the above system, a file "sample.txt" of size 50KB is stored. The i^{th} entry of the inode (i starts from 0) points to a data block stored on track $(i \times 30)\%n$ where n=100 are the total number

of tracks (track numbers: 0-99). If SSTF disk scheduling algorithm is used, compute the total time required to read the complete file (assume that the file is already open). Assume that the disk head is initially on track 83 and disk controller overhead is 0.1ms per head movement.

(b) Consider a memory management scheme implementing segmentation with paging. The maximum size of each segment is of 4GB and the maximum number of segments per process is 16. The frame size is of 4KB. If each page table entry takes 2 bytes, compute the size of the page table. At most how many page tables are required per process?

(c) "TLB hit may result in a page fault"—justify the statement.

[(2+4)+2.5+1.5]

4. In the sleeping Barber's problem, consider the following code snippet of Barber and Customer

(a)

```
void barber(void)
{
      while (TRUE) {
            down(&customers);
            down(&mutex);
            waiting = waiting - 1;
            up(&barbers);
            up(&mutex);
            cut_hair( );
      }
}
```

```
void customer(void)
{
      down(&mutex);
      if (waiting < CHAIRS) {
            waiting = waiting + 1;
            up(&customers);
            up(&mutex);
            down(&barbers);
            get_haircut( );
      } else {
            up(&mutex);
      }
}
```

(i) In the barber code, explain the impact of exchanging down(&mutex) with down(&customer)

(ii) In the customer code, explain the impact of exchanging up(&mutex) with down(&barber)

(b) "The proposed final solution of the dining philosopher problem is free from deadlock and starvation"---justify the statement. Feel free to use the code snippet of the solution.

(c) Operating system may need to access a data block from disk for two different purposes (a) reading a regular file (b) load the required page from disk in case of page fault. Explain, from the view of the File management module, how do these two accesses differ.

(d) A disk supports FAT16 File system. If each block size is of 32KB, compute the maximum size of the disk.

[4+3+1.5+1.5]

5. (a) An operating system uses the Banker's algorithm for deadlock avoidance when managing the allocation of three resource types X, Y, and Z to three processes P0, P1, and P2. The following table presents the current system state. Here, the Allocation matrix shows the current number of resources of each type allocated to each process and the Max matrix shows the maximum number of resources of each type required by each process during its execution.

	Allocation			**Max**		
	X	Y	Z	X	Y	Z
P0	0	0	1	8	4	3
P1	3	2	0	6	2	0
P2	2	1	1	3	3	3

There are 3 units of type X, 2 units of type Y and 2 units of type Z still available. The system is currently in a safe state. Consider the following independent requests for additional resources in the current state:
REQ1: P0 requests 2 units of Z
REQ2: P1 requests 2 units of X

Explain which requests will be granted by the deadlock avoidance module.

(b) Consider a system consisting of a number of processes and resources. Assume that we have single instance of each resource type. Illustrate three independent examples where the system moves from (i) safe state to unsafe state (ii) unsafe state to deadlock state (ii) unsafe state to safe state. Show the corresponding wait-for graph.

(c) A computer has six tape drives with n processes competing for them. Each process needs two drives. What is the maximum value of n to keep the system deadlock free?

(d) Justify the rationale behind the improvement of C-SCAN algorithm over SCAN.

[4+2+2+2]

6.(a) Prof. X proposes the following solution of the producer consumer problem. Justify the correctness of the solution.

```
Semaphore n=0;
Semaphore s=1;
void producer()
{
        while(true)
        {
              Produce_item();
              wait(s);
              insert_buffer();
              signal(s);
              signal(n);
        }
}
```

```
void consumer()
{
        while(true)
        {
              wait(s);
              wait(n);
              remove_buffer();
              signal(s);
              consume();
}
```

(b) Consider in a Unix system, a file test.txt is stored at the location **/usr/www/test.txt**. A process P1 opens the file and performs read operation. Specify all the steps showing access of the corresponding i-node and the subsequent updation in the System Wide Open File Table, Per-Process Open File Table etc.

(c) "Linked file allocation requires less access time compared to indexed node (i-node) scheme" Comment with suitable argument.

(d) Consider a 20MB disk has blocks of size 1KB with 16 bit disk block address. What is the maximum number of blocks needed to implement (a) grouping and (b) bitmap implementation of the free space manager?

[3+2.5+1.5+3]