This question paper contains 8 printed pages]

Roll No. | | | | | | | | | | |

S. No. of Question Paper : 49

Unique Paper Code : 32341302 I

Name of the Paper : **Operating Systems**

Name of the Course : **B.Sc. (H) Computer Science**

Semester : III

Duration : 3 Hours Maximum Marks : 75

(Write your Roll No. on the top immediately on receipt of this question paper.)

Question No. 1 of 35 marks is compulsory.

Attempt any *four* questions from Q. Nos. 2 to 7.

1. (*a*) Fill in the following blanks : 5

(*i*) The two modes of execution of an operating system are ..... and .....

(*ii*) ..... provide(s) an interface to the services provided by an operating system.

(*iii*) A necessary condition for a deadlock which states that a resource held by a process cannot be taken away forcibly .....

P.T.O.

(*iv*) ..... scheduling is approximated by predicting the next CPU burst with an exponential average of the measured lengths of previous CPU bursts.

(*v*) The mapping of a logical address to a physical address is done in hardware by the .....

(*b*) What is a file allocation table ? 2

(*c*) Differentiate between : 4

(*i*) binary semaphore and mutex

(*ii*) counting semaphore and binary semaphore ?

(*d*) What is a bootstrap program ? Where is it stored ? 2

(*e*) Given five memory partitions of sizes 750 KB, 575 KB, 225 KB, 510 KB, 300 KB (in order). How would the best-fit algorithm place processes of sizes 450 KB, 540 KB, 200 KB, and 560 KB (in order) ? 2

(*f*) What are the advantages of using loadable kernel modules ? 2

(*g*) Which of the following scheduling algorithms could result in starvation ? Justify your answer : 2

(*i*) First-come, first-served

(*ii*) Shortest job first

(*iii*) Round robin

(*iv*) Priority.

(*h*) Given that the actual pids of the parent and child in the following program fragment are 1500 and 1700 respectively, identify the pids at lines (*a*), (*b*), (*c*) and (*d*) (assume that fork is executed successfully) : 4

```
{
pid_t pid1, pid2;
pid1 = fork();
if (pid1 == 0){
    pid2 = getpid();
    printf("pid1 = %d", pid1);        /* (a) */
    printf("pid2 = %d", pid2);        /* (b) */
    }
else{
    pid2 = getpid();
    printf("pid1 = %d", pid1);        /* (c) */
    printf("pid2 = %d", pid2);        /* (d) */
    wait (NULL);
    }
}
```

P.T.O.

(*i*) What is the use of `pthread_join( )` function ? 2

(*j*) Consider a logical address space of 128 pages with 2 KB page size, mapped onto a physical memory of 64 frames : 3

(*i*) How many bits are required in the logical address ?

(*ii*) How many bits are required in the physical address ?

(*k*) Assume a program has just referenced an address in virtual memory. Which of the following scenarios can occur and which cannot. Justify your answer : 3

(*i*) TLB hit and page fault

(*ii*) TLB miss with no page fault.

(*l*) What is a mount point ? 2

(*m*) Describe the in-memory structures for file system management. 2

2. (*a*) Compare client-server computing and peer to peer computing. 3

(*b*) Consider the following code segment : 4

```
pid_t pid;
pid = fork();
if (pid == 0){
    fork();
    thread_create( . . .);
}
fork();
```

(*i*) How many unique processes are created ?

(*ii*) How many unique threads are created ?

(*c*) What are the three mechanisms for implementing index block for large files in the indexed allocation scheme ? 3

3. (*a*) Differentiate between internal and external fragmentation. Which of the following memory organization schemes suffer from external fragmentation : contiguous memory allocation, paging. Give arguments to support your answer. 2+3

(*b*) Describe the Readers Writers synchronization problem. Suggest the process structures to solve this problem. 5

4. (*a*) Consider the following set of processes, with the length of the CPU burst given in milliseconds. Given that the order of arrival of the processes is P1, P2, P3 and P4 (all at time zero), determine the average waiting time of each process for the following scheduling algorithms : 6

(*i*) SJF (non-preemptive)

(*ii*) RR (quantum = 2)

Process	Burst Time
P1	4
P2	3
P3	9
P4	6

(*b*) How many memory accesses are required in the case of a TLB hit and TLB miss ? Consider a paging system with the page table stored in TLBs. Given that 90 percent of the page references are found in the TLBs, determine the effective memory reference time if a memory reference takes 150 nanoseconds. (Assume that finding a page-table entry in the TLBs takes zero time, if the entry is there.) 4

5. (*a*) Two processes P1 and P2 are simultaneously accessing the following code. Demonstrate the impact of race condition in this scenario : 3

```
x = 1
Func( )
{
    If (x ==0)
        Return;
    X--;
}
```

(*b*) What do you understand by *locality of reference* ? Explain the working set model to avoid thrashing. 3

(*c*) List the different types of directory structures giving one advantage of each. 4

6. (*a*) Explain the following in the context of demand paging : 4

(*i*) Belady's anomaly

(*ii*) copy-on-write.

(*b*) Assuming a 2 KB page size, what are the page numbers and offsets for the following address references provided as decimal numbers (assume that the page numbers begin with zero) : 2

(*i*) 7825

(*ii*) 17239.

P.T.O.

(c) For a 32 bit logical address, calculate the number of bits in the page number and page offset fields given that the page size is 2 KB. 2

(d) Would it be appropriate to have a web server run as a single-threaded process ? Why or why not ? 2

7. (a) Write two methods that implement the `wait( )` and `signal( )` operations for a semaphore *s*. 4

(b) Consider the following page reference string : 4

9, 5, 3, 6, 5, 8, 2, 1, 9, 9, 0, 7.

Assuming demand paging with four frames, how many page faults would occur for the following replacement algorithms ?

- FIFO replacement
- Optimal replacement.

(c) Justify that the mutual exclusion condition is necessary for a deadlock to occur. 2