ROLL NO. _____

Code: CT31 Subject: OPERATING SYSTEM

ALCCS - NEW SCHEME

Time: 3 Hours **AUGUST 2013** Max. Marks: 100

PLEASE WRITE YOUR ROLL NO. AT THE SPACE PROVIDED ON EACH PAGE IMMEDIATELY AFTER RECEIVING THE QUESTION PAPER.

NOTE:

- **Question 1 is compulsory and carries 28 marks. Answer any FOUR questions from the rest. Marks are indicated against each question.**
- **Parts of a question should be answered at the same place.**

Q.1 a. Explain concept and working of Virtual machine.

b. What is process control block? Explain.

c. What do you mean by boot strap program? Explain briefly.

d. What are the advantages of multiprocessor system? Explain briefly.

e. Explain the concept of demand paging.

f. Explain the layered architecture of an operating system.

g. Differentiate between preemptive and non preemptive scheduling. (7 × 4)

Q.2 a. Describe reasons for providing an environment that allows process cooperation. Explain the following terms related to IPC:
Naming, Synchronization and Buffering (9)

b. Consider two processes P1 and P2, where p1 = 50, t1 = 25, p2 = 75 and t2 = 30.
(i) Can these two processes be schedules using rate-monotonic scheduling? Illustrate your answer using a Gantt Chart.
(ii) Illustrate the scheduling of these two processes using earliest-deadline-first (EDF) scheduling. (9)

Q.3 a. What aspects of a distributed system would you select for a system running on a totally reliable network? (6)

b. What are the advantages and disadvantages of buffer cache? (6)

c. Why is deadlock detection much more expensive in a distributed environment than in a centralized environment? (6)

Q.4 a. Explain different methods for user authentication for security. (6)

b. What is system call? Explain various types of system call. (6)

c. How does the principle of least privilege aid in the creation of protection systems? (6)

Q.5 a. What is a process? Discuss various states of a process? Give a list of events that are responsible for process state transitions. (6)

b. Explain Banker's algorithm to handle deadlocks. (6)

c. Describe a mechanism by which one segment could belong to the address space of two different processes. (6)

Q.6 a. What are the six basic file operations? Explain in detail. (6)

b. What is a semaphore? Explain the Critical Section (CS) implementation with semaphores. (6)

c. Differentiate between static and dynamic disk scheduling for multimedia. (6)

Q.7 a. What is paging? Explain paging hardware. (6)

b. Compare following CPU scheduling algorithms using a suitable example.

(i) FCFS
(ii) Priority
(iii) Round Robin

Which one is best to use? (8)

c. There are two sets of reasons for high overheads of switching between processes- Intrinsic reasons and OS-related reasons. Briefly describe each. (4)

Time: 3 Hours

AUGUST 2013

Max. Marks: 100

PLEASE WRITE YOUR ROLL NO. AT THE SPACE PROVIDED ON EACH PAGE IMMEDIATELY AFTER RECEIVING THE QUESTION PAPER.

NOTE:

- **Question 1 is compulsory and carries 28 marks. Answer any FOUR questions from the rest. Marks are indicated against each question.**
- **Parts of a question should be answered at the same place.**

Q.1 a. What are the two main functions of an operating systems?

b. Why "disable all interrupts" is implemented only in kernel mode?

c. How does priority scheduling differ from round robin method?

d. What is device independence?

e. What are the different modes in which processes are executed?

f. Give an example of the situation describing deadlock.

g. What is RAM Disk? (7×4)

Q.2 a. Draw process transition diagram and explain the meaning of "ready to swap in" process. (5)

b. Give an example of classical problem of concurrent programming and show how it leads to deadlock? (5)

c. Explain the concept of Least Recently Used memory page replacement method and how it is different from First In First Out (FIFO) page replacement method. (8)

Q.3 a. What is the use of process table in process management? How does OS update a process table? (6)

b. How does memory compaction works? Write an algorithm for the memory compaction. (6)

c. For a given reference strings: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5. Using LIFO algorithm of page replacement, compute number of page faults for number of frame = 3 and number of frame = 4. (6)

Q.4 a. Discuss the file protection mechanisms incorporated in a Unix file system. (6)

b. Define UFD and MFD. What are the most common schemes for defining the logical structure of a directory? (6)

c. What is Virtual File Systems? How is it implemented? (6)

Q.5 a. Why is it difficult to protect a system in which users are allowed to do their own I/O? Explain how relocation helps for the protection of the data? (6)

b. What are the advantages of Contiguous allocation? What are the drawbacks of contiguous allocation of disk space? (6)

c. A hard disk has 63 sectors per tracks, 10 platters each with 2 recording surfaces and 1000 cylinders. The address of a sector is given as a triple <c, h, s> where c is the cylinder number, h is the surface number and s is the sector number. Thus 0th sector is addressed as <0, 0, 0>, the 1st sector is addressed as <0, 0, 1> and so on. Calculate the sector number whose address is <400, 16, 29>. (6)

Q.6 a. What advantages do threads have over multiple processes? What major disadvantages do they have? (5)

b. What is the time stamping scheme of distributed system for mutual exclusion? Explain the functioning of the scheme through a diagram. (5)

c. Write one method suitable for inter-processor communication in hypercube multiprocessor operating system. (8)

Q.7 Write short notes on any THREE of the followings:

(i) Virtual Memory
(ii) Clock Synchronization
(iii) Interrupt Handling
(iv) Binary semaphore

(6+6+6)

Code: CT31 ROLL NO. _____

Subject: OPERATING SYSTEM

ALCCS - NEW SCHEME

Time: 3 Hours

FEBRUARY 2013

Max. Marks: 100

PLEASE WRITE YOUR ROLL NO. AT THE SPACE PROVIDED ON EACH PAGE IMMEDIATELY AFTER RECEIVING THE QUESTION PAPER.

NOTE:

- **Question 1 is compulsory and carries 28 marks. Answer any FOUR questions from the rest. Marks are indicated against each question.**
- **Parts of a question should be answered at the same place.**

Q.1 a. What is semaphore? What is its use?

b. List the different actions taken by a time sharing scheduler.

c. What is cooperating process? Give reasons for providing an environment that allows process cooperation?

d. List the action taken by event handler when a process makes an I/O request?

e. Why are segmentation and paging sometimes combined into one scheme?

f. Write the functions of Virtual File System (VFS) layer?

g. What are the components of a Linux system?

(7 × 4)

Q.2 a. What is an operating system? Discuss the various functions of operating system. (8)

b. What are the benefits of Multithreaded Programming? (4)

c. State various scheduling criteria that must be kept in mind while choosing different scheduling algorithms. (6)

Q.3 a. What is critical-section problem? Give a classic Peterson's solution to the critical-section problem. (2+6)

b. Consider the following system snapshot using data structures in the Banker's algorithm, with resources A, B, C, and D and process P0 to P4:

	Max				Allocation				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	6	0	1	2	4	0	0	1	3	2	1	1
P1	1	7	5	0	1	1	0	0				
P2	2	3	5	6	1	2	5	4				
P3	1	6	5	3	0	6	3	3				
P4	1	6	5	6	0	2	1	2				

Using Banker's algorithm, answer the following questions:
(i) How many resources of type A, B, C, and D are there?
(ii) What are the contents of the Need matrix?
(iii) Is the system in a safe state? Why?
(iv) If a request from process P4 arrives for additional resources of (1,2,0,0,), Can the Banker's algorithm grant the request immediately? Show the new system state and other criteria. (2+2+2+4)

Q.4 a. Describe the First fit, Best fit and Worst fit allocation algorithms. Given memory partitions of 100K, 500K, 200K, 300K, and 600K (in order), how would each of the First-fit, Best-fit, and Worst-fit algorithms place processes of 212K, 417K, 112K, and 426K (in order)? Which algorithm makes the most efficient use of memory? (4+6)

b. Consider a demand-paging system with the following time-measured utilizations:

CPU utilization	20%
Paging disk	97.7%
Other I/O devices	5%

Which (if any) of the following will (probably) improve CPU utilization? Explain your answer.

(i) Install a faster CPU.
(ii) Install a bigger paging disk.
(iii) Increase the degree of multiprogramming.
(iv) Decrease the degree of multiprogramming.
(v) Install more main memory.
(vi) Install a faster hard disk or multiple controllers with multiple hard disks.
(vii) Add prepaging to the page fetch algorithms.
(viii) Increase the page size. (8)

Q.5 a. Why do some systems keep track of the type of a file, while others leave it to the user or simply do not implement multiple file types? Which system is "better"? (4)

b. Discuss the following RAID levels:
(i) RAID Level 3
(ii) RAID Level 4
(iii) RAID Level 5 (3×3)

c. Write the principles that may be employed to improve the efficiency of I/O operations. (5)

Q.6 a. List the basic issues that the designer of a communication network must address. (5)

b. How cached data are verified for its validity in distributed file system? (5)

c. Explain the different approaches for implementing mutual exclusion in a distributed environment. (8)

Q.7 Write short notes on the following: Subject: OPER

(i) Port Scanning
(ii) Process Migration
(iii) Semaphore and its use
(iv) Types of viruses

Name _____

Roll no. _____

1
2
3
4
Tot

CS 347: Operating Systems
Mid-sem Examination. 13 September 2007

Question-cum-answer paper. Do not detach any pages.

Note: Your answers must be complete in all relevant respects and also self-explanatory.

Q.1 (10 + 10 Marks)

A time sharing system uses a time slice of 100 msec. Each process has a cyclic behavior pattern. In each cycle, it requires 50 msec of CPU time to compute the result of a subrequest and 150 msec to print it on the user's screen. A process receives a new subrequest one second after it has finished printing results of the previous subrequest. The operating system can accommodate 10 processes in memory at any time, however it has enough I/O devices for 25 processes. The swap-in and swap-out times of each process are t_s msec each. Calculate the average throughput of the system over a 10 second period in each of the following cases:

(a) The operating system contains 10 processes.

Let the processes be numbered $P_1 \ldots P_{10}$. Process P_1 uses the CPU for 50 msec and starts an I/O operation that requires 150 msec. Hence it produces a response at the end of 0.2 second.	1 Mark
It receives the next request from the user after 1 second, i.e., at 1.2 seconds.	1 Mark
P_2 produces a response at 0.25 seconds, and so on.	1 Mark
Process P_{10} produces a response at 0.65 seconds.	1 Mark
The system is idle between 0.65 seconds and 1.2 seconds because none of the processes is servicing a request. This pattern repeats every 1.2 seconds.	2 Marks
The ninth such cycle begins at 9.6 seconds.	1 Mark
Processes $P_1 \ldots P_5$ complete processing of their ninth requests at 9.8, 9.85, 9.9, 9.95 and 10.0 seconds.	1 Mark
Thus, 85 requests are processed in 10 seconds, and the throughput is 8.5 requests per second.	2 Marks

(b) The operating system contains 20 processes and t_s is 250 msec.

Process P_1 can be swapped out at 0.2 seconds and P_{11} could be swapped in in its place. This operation would complete at 0.7 seconds, hence P_{11} can be scheduled at 0.7 seconds and it would produce a response at 0.9 seconds. — 2 Marks

If P_2 is replaced by P_{12} through swapping when its I/O operation completes, swapping in of P_{12} would complete at 0.75 seconds, etc. — 1 Mark

After P_{11} produces a response, it would be replaced back by P_1 and this operation would complete at 1.4 seconds. — 1 Mark

Thus, each of the 20 processes would service one request every 1.4 seconds. — 1 Mark

So each process would have serviced 7 requests by 9.8 seconds. — 1 Mark

P_1 starts servicing its eighth request at 9.8 seconds and completes at 10.0 seconds. — 2 Marks

Hence $7 \times 20 + 1$ requests would be serviced in 10 seconds, hence the throughput is 14.1 requests per second. — 2 Marks

Note: If you have assumed that swapping occurs in paralle with I/O, and if the rest of your answer is correct, you have been given marks out of 7 instead of out of 10.

Q.2 (10 + 10 Marks)

(a) A signal named SIGX is defined in an OS. An application contains two processes named A and B. Process A wishes process B to take a specific action when it sends the signal SIGX to it.

(i) Explain what provision process B must make for this purpose.

B must code the signal handling action as a routine — 1 Mark
It must make a system call to install the signal handling action — 1 Mark

(ii) Describe what arrangement the kernel must set-up for this purpose.

The kernel must define/build a signal vector — 1 Mark
The signal handler should be accessible from the PCB of a process — 1 Mark
At a "install signal handler" call, it must enter address of the signal handling routine in the appropriate entry of the signal vector — 1 Mark

(iii) Describe actions of the kernel when process A sends signal SIGX to process B.

The kernel obtains id of process B from the "send signal" system call — 1 Mark
It locates the signal vector of B from its PCB — 1 Mark
It finds address of the signal handler installed by B — 1 Mark
It pushes the contents of the PC field of B's PSW on B's stack — 1 Mark
It puts address of the signal handler in the PC field of B's PSW — 1 Mark

(b) An application using threads is to be developed for implementation on single-CPU computer systems as well as multiple-CPU computer systems.

(i) Explain what thread model you would use for implementing it on a **single-CPU computer system**. Give detailed justifications and mention whether the OS should provide any facility in addition to the model of threads chosen by you.

Parallelism of the threads is not important; only concurrency is important	1 Mark
Low thread switching overhead is important	1 Mark
Because switching is performed by the thread library, thread switching overhead is low	2 Marks
Hence use user-level threads	1 Mark
A library to perform certain actions without blocking a thread is essential	2 Marks

(ii) Explain what thread model you would use for implementing it on a **multiple-CPU computer system**. Give detailed justifications and mention whether the OS should provide any facility in addition to the model of threads chosen by you.

Parallelism between threads is essential	1 Mark
Hence kernel-level threads	1 Mark
Thread switching overhead is not so important	1 Mark
No special provisions needed in the OS	1 Mark

Note: Max marks = 3.

Q.3 (4*4 Marks)

(a) Is the following statement true or false? Give a justification for your answer. (Note: No credit unless a matching justification is given.)

If all requests arrive at the same time instant in a system using the shortest request next (SRN) scheduling policy and the system completes execution of these requests in the sequence $r_1, r_2, \ldots r_n$, then weighted turnaround of $r_i >$ weighted turnaround of r_j if $i > j$.

A shorter process is scheduled before a longer one — 1 Mark
However, weighted turnaround depends on both the turnaround time and the service time of a process — 1 Mark
Hence there is no guarantee that it would be less for a process that finishes earlier. — 1 Mark
An example or an explanation in support of the above statement — 1 Mark

(b) Is the following statement true or false? Give a justification for your answer. (Note: No credit unless a matching justification is given.)

If some requests arrive at the same time instant in a system using the round-robin scheduling policy with time slicing, the response ratios of these requests are approximately equal when they complete.

Response ratio is the ratio of (time since arrival + service time) to service time of a process. Response ratio when a process completes is the ratio of (turn around time + service time) / service time — 1 Mark

A process that does not start an I/O operation before exhausting its time slice receives a service time = time slice δ in one round of the round-robin — 1 Mark
A process that performs I/O may receive more service time than δ in one round of the round-robin — 1 Mark
Hence response ratios may not be equal at completion — 1 Mark

(c) Is the following statement true or false? Give a justification for your answer. (Note: No credit unless a matching justification is given.)

When both CPU-bound and I/O-bound requests are present, the least completed next (LCN) scheduling policy provides better turnaround times for I/O-bound requests than provided by the round-robin scheduling policy with time slicing (RR).

In round-robin scheduling, an I/O bound request gets only one opportunity to initiate an I/O operation in one "round" of the scheduler. — 1 Mark

The LCN policy schedules the request that has received the least amount of CPU time — 1 Mark

Hence if the time taken by an I/O bound request is $<$ the time taken for one round of the RR scheduler, an I/O bound request would get more opportunities to initiate an I/O operation in the LCN scheduler than in the RR scheduler. Hence it may finish earlier. — 2 Marks

(d) Which of the following scheduling policies will provide the least turnaround time for a CPU-bound process if both I/O-bound and CPU-bound requests are present in the system? Justify your answer.

(i) RR, (ii) LCN, (iii) multilevel adaptive scheduling.

(Note: RR and LCN have been defined in part (c) above.)

In round-robin scheduling, a CPU-bound request gets δ CPU seconds in one round of the RR scheduler. An I/O-bound request does not consume much CPU time in one round of scheduling. — 1 Mark

In LCN scheduling, I/O-bound requests and short requests can starve the CPU-bound requests that are long — 1 Mark

In multilevel adaptive scheduling, CPU-bound requests receive low priorities — 1 Mark

Hence, RR policy will provide the best turnaround time for CPU-bound requests — 1 Mark

Q.4 (10+34 Marks)

(a) Processes in a producer–consumer system having multiple producers, multiple consumers and multiple buffers use the code give below:

```
var     turn : 1 .. 2;
begin
        turn := 1;
Parbegin
    repeat
        while turn = 2
        do { nothing };
        { Produce if empty buffer exists }
        turn := 2;
        { Remainder of the cycle }
    forever;
Parend;
end.
```

Producer process

```
repeat
    while turn = 1
    do { nothing };
    { Consume if full buffer exists }
    turn := 1;
    { Remainder of the cycle }
forever;
```

Consumer process

Comment on how well this implementation fulfills the requirements of the producers–consumers problem.

Any number of producers may be able to get in when turn =1. (Similarly for consumers. Give full marks if the student mentions any one of these.)	2 Marks
Busy waits would arise if more producers get in than the number of empty buffers. (Similarly for consumers. Give full marks if the student mentions any one of these.)	2 Marks
Race conditions on free buffers could arise is many producers get in (Similarly for consumers. Give full marks if the student mentions any one of these.)	2 Marks
Producers may be blocked even when empty buffers exist (Similarly for consumers. Give full marks if the student mentions any one of these.)	2 Marks
Some buffers may never get used	2 Marks

(b) A 4-way road intersection has cars approaching it from the east, west, north and south directions. Each car is required to follow these rules at the intersection:

(i) A car can begin to cross the intersection if some other car(s) immediately ahead of it and travelling in the same direction is/are crossing it, or

(ii) A car can begin to cross the intersection if the road on its right does not have any cars waiting to cross the intersection. (Thus, a car approaching the intersection from the east should check whether any cars have already reached the intersection on the road from north and are waiting to cross.)

Write a concurrent program using **either semaphores or monitors** for controlling traffic. (*Note:* Do not worry about whether these rules are "good"/"bad" or dangerous.)

Distribution of marks:

A separate semaphore for each direction and one common semaphore : 3 Marks

Pseudocode : 22 Marks

Incrementing/decrementing semaphores correctly : 6 Marks

No busy wait in signaling : 3 Marks.

Common mistakes/important points in random order:

(a) Code not self-explanatory AND comments are missing.

(b) The last car to cross in a specific direction should give a signal to car(s) on its left. However, if semaphores are used and a signal is given when no cars are present to its left, it can lead to problems in future.

Also, this signaling should be done inside a critical section; otherwise, race conditions can arise. See Note (d) before the pseudocode, on next page.

(c) Binary semaphores alone are not adequate.

(d) Rule 1 should be checked before rule 2.

Pseudocode for a car approaching the intersection from the East:

Notes:

(a) This is "a" pseudocode. Alternatives that are equivalent have been given due marks.

(b) Sem_E is the semaphore used to decide whether the car can cross.

(c) Sem_S is the semaphore used to signal car(s) that approached from the South and are waiting.

(d) The complete body of end_crossing_E() should be inside a critical section. 3 Marks for it.

(e) The critical sections of all the begin_crossing() and end_crossing() functions should be implemented on the same semaphore so that race conditions do not arise while checking counter values.

(f) Full marks have been given for end_crossing_E() even if only one car waiting on the left has been activated. Bonus marks will be awarded to answers that activate all waiting cars. **You must come forward to claim this bonus.**

Code for the process:

```
If car ahead is already crossing                          5 Marks
    begin_crossing_E()
    { Cross }
    end_crossing_E()
Else                                                      3 Marks
    If (no cars waiting on right)
        begin_crossing_E()
        { Cross }
        end_crossing_E()
Else
    wait (sem_E) { Cars exist on the right }              5 Marks
    begin_crossing_E()
    { Cross }
    end_crossing_E()
```

Code for the functions:

```
begin_crossing_E()                                        3 Marks
    Increment count (This is in a CS)
    return

end_crossing_E()                                          3 Marks
    Decrement count
    If count = 0 then
        { Wake up all waiting cars on its left
        through Signal(sem_S) operations }
    return
```

MAHATMA GANDHI UNIVERSITY
M.C.A DEGREE EXAMINATION
MODEL QUESTION PAPER
(2011 Revised Syllabi)
Second Semester
MCA 204 Operating Systems

Time: 3 hrs Max: 75 Marks

Part A

Answer any ten questions.
Each question carries 3 marks.

1. Explain about process state transition diagram.
2. Explain about system calls.
3. Explain critical section.
4. Discuss the necessary conditions of deadlock.
5. Explain about swapping.
6. Explain the terms first-fit, best fit, and worst fit.
7. Explain about overlays.
8. What is meant by virtual memory?
9. Explain about Thrashing.
10. Explain about disk structure.
11. Explain about file attributes.
12. Differentiate between sequential and direct file access methods.

(10 x 3 = 30 marks)

Part B

All questions carry equal marks.

13. a. Explain the various scheduling algorithms.

OR

b. Explain about different types of operating system.

14. a. Discuss busy-wait implementation of semaphore

OR

b. Explain deadlock avoidance with Banker's algorithm.

15. a. Explain about paging.

OR

b. Explain about segmentation

16. a. Explain Demand Paging.

OR

b. Explain various Disk Scheduling algorithms.

17. a. Explain all file allocation methods

OR

b. Discuss file system implementation.

(5 X 9 = 45 marks)

4 Operating Systems (RMM)

(*a*) What is stored in a *directory entry* and what is stored in a *file entry*? [2 marks]

(*b*) Consider a file system that uses a fixed size 128 byte directory entry structure, a fixed size 32 byte file entry structure, and manages storage via the "chaining in a map" technique, on a disk that uses 4 kByte blocks. Assume that the map of blocks is permanently in memory and need not be read from disk. Assume further that the file system contains 2^{20} files.

(*i*) Assume this file system is structured as a *file system with a single level directory*. Explain how many blocks must be read in order to open and read a 16 byte file named `/a/b/c/d/e`. [3 marks]

(*ii*) Instead, assume this file system is structured as *a simple hierarchical file system with directories that can contain up to 256 sub-directories and 1024 files*. Give upper and lower bounds on how many blocks must be read in order to open and read a 16 byte file named `/a/b/c/d/e` where `/` denotes the path separator. State any assumptions you make in computing your bounds. [10 marks]

(*c*) Unix uses a more complex file entry structure than assumed above, known as an *i-node*. In addition to file metadata, the i-node stores 12 direct blocks, plus single, double and triple indirect blocks (each of which has 512 entries). Explain what purpose is served by each of these four types of block. [3 marks]

(*d*) Using Unix i-nodes as above, compute the maximum size file that can be stored. Assume a block is 4 kBytes in size. [2 marks]

3 Operating Systems (RMM)

Consider a simple operating system where live processes are either *running*, *ready* to run, or *blocked* on an event, before they *exit*.

(*a*) State four conditions under which the operating system will try to schedule processes. [4 marks]

(*b*) OS schedulers are said to be *pre-emptive* or *non-preemptive*. State the principal problem with non-preemptive schedulers. Explain how pre-emptive schedulers solve this problem. [2 marks]

(*c*) Explain why pre-emptive schedulers are more complex to implement. [2 marks]

(*d*) Consider four CPU-bound processes arriving to a Shortest Remaining Time First (SRTF) scheduler as follows:

Process	Arrival Time	Duration
P_1	0	8
P_2	3	3
P_3	5	1
P_4	9	4

(*i*) Give a schedule computed by the SRTF scheduler, and compute the average waiting time across all four processes. Your answer should be explicit about the state of each process and the *ready* queue at all times. Clearly state any assumptions you make. [6 marks]

(*ii*) Explain why SRTF is difficult to implement in practice, and propose how to address this difficulty. [2 marks]

(*iii*) Assume that jobs are no longer CPU-bound but also perform blocking and non-blocking IO. Discuss how this can affect the effectiveness and fairness of the SRTF scheduler. [4 marks]

SATHYABAMA UNIVERSITY

(Established under section 3 of UGC Act,1956)

Course & Branch :B.E/B.Tech - CSE/DCS/IT
Title of the Paper :Operating Systems
Sub. Code :6C0063
Date :04/05/2010

Max. Marks :80
Time : 3 Hours
Session :FN

PART - A (10 x 2 = 20)

Answer ALL the Questions

1. Define thread.
2. List the functions of an operating system.
3. What is process abstraction?
4. Compare preemptive and non-preemptive scheduling.
5. What is synchronization primitive? Give examples.
6. What do you mean by deadlock?
7. Define paging.
8. What is file naming? Give some typical file extensions.
9. List any four shell operation commands.
10. Give the use of telnet.

PART – B

(5 x 12 = 60)

Answer All the Questions

11. (a) Explain the basic structure of an operating system. (8)

(b) Describe the factors that influence in operating system design. (4)

(or)

12. Discuss in detail about the implementation consideration in an operating system.

13. (a) Explain how to create and terminate a process?

(b) Write note on process scheduling.

(or)

14. How do you solve the producer-consumer problem using semaphore? Explain.

15. Write a detailed note on Inter Process communication.

(or)

16. Write the conditions for the occurrence of deadlock. Explain the Banker's algorithm with an example.

17. (a) Explain the dynamic paging algorithm.

(b) What is Segmentation? Explain.

(or)

18. (a) Explain the file system layout. Discuss the issues in implementing file storage. (8)

(b) Explain briefly about protection mechanism. (4)

19. (a) Explain the Linux file structure.

(b) Describe about the internet services.

(or)

20. Write a shell program for the following:

(a) to find the sum of 'n' natural numbers. (4)

(b) to find the biggest among three numbers. (8)

Institute of Technology and Management Universe, Vadodara
Computer Science Engineering
B.E Semester: 4

Subject Code: 140702
Subject Name: OPERATING SYSTEM

Unit-1: Introduction

Contents
What is an OS?, Evolution Of OS, OS Services, Types Of OS, Concepts of OS, Different Views Of OS, Concepts of OS

Q1. What is operating System? Explain the abstract view of the components of a computer system. [June 2011]

Q2 Explain briefly different types of operating systems. [Dec 2013]

or

List the types of operating systems and explain any one in detail [June 2013]

or

Give the features of Batch Operating System. [Dec 2011]

or

Give the advantages of Distributed Operating System. [Dec 2011, Dec 2012, Dec 2013]

or

Explain the features of Real Time Operating System. [Dec 2011,12, Dec 2013]

Explain the features of Time Sharing System. [Dec, 2011]

Write different operating system services. [Dec, 2012]

Explain different types of OS and also Explain different types of tasks done by OS. [June 2012, Dec 2013]

Unit -6 :Input Output Management Principles Of Input/Output H/W

I/O Devices, Device Controllers, Direct Memory Access, Principles Of Input/Output S/W: Goals Of The I/O S/W, Interrupt Handler, Device Driver,Device Independent I/O Software Disks : RAID levels, Disks Arm Scheduling Algorithm, Error Handling

Q1. Explain Goals of I/O Software. [June 2010] [June, 2010]

or

What is Device Controller, Device drivers? [June 2012] [June 2013]

or

What are the uses of device driver and controller in OS? [Dec 2013]

Q2. Explain RAID level system. [June 2011, Dec 2013]

or

Short note on RAID. [June 2013]

Q3. Explain Direct Memory Access (DMA). [June 2010]

Q4. Explain Device Independent I/O software. [Dec 2012]

Q5. Explain any Three Disk Arm Scheduling Algorithms with illustration. [Dec 2011]

or

Explain SSTF and LOOK disk scheduling algorithms. [June 2011]

Numerical

Q1. Disk requests come in to the disk for cylinders 10, 22, 20, 2, 40, 6 and 38. A seek takes 6 msec per cylinder move. How much seek time is for Closest cylinder next algorithm? Initially arm is at cylinder 20. [June 2013]

Unit-3 :Interprocess Communication

Race Conditions, Critical Section, Mutual Exclusion, Hardware Solution, Strict Alternation , Peterson's Solution, The Producer Consumer Problem, Semaphores, Event Counters, Monitors, Message Passing, Classical IPC Problems: Reader's & Writer Problem, Dinning Philosopher Problem etc., Scheduling , Scheduling Algorithms.

Q1. What is Semaphore? [Dec 2011]

Q2. What is Semaphore? Give the implementation of Readers-Writers Problem using Semaphore. [Dec, 2011, Dec 2013]

Q3. What is Semaphore? How can we achieve the synchronization using semaphore for producer – consumer problem? [June 2013]

Q4. Explain IPC Problem –Dining Philosopher Problem. [June 2012]

Explain IPC Problem – Readers & Writers Problem.

or

Q5. What is priority inversion problem in inter process communication? How to solve it with semaphore? [Dec 2010]

Q6. Give the implementation of Bounded Buffer Producer Consumer Problem using Semaphore. [June 2010]

Q7. Define: Race Condition, Mutual Exclusion, Throughput [Dec 2013]

Q8. Discuss the Peterson's solution for the race condition with algorithm. [June 2013]

Q9. What is advantage of using Monitor? Give the implementation of Bounded Buffer Producer Consumer Problem using "Monitor". [June 2010]

Q10 What is monitor? Give the implementation of Bounded Buffer Producer- Consumer Problem using monitor. [Dec 2011]

Unit -2: Process Management:

Contents
Process, Process Control Block, Process States, Threads, Types of Threads, Multithreading.

Q1. What is Process State? Explain different states of a process with various queues generated at each stage. [June, 2010]

or

Draw process state diagram for THREE states and explain all states. [June 2013]

or

Explain "5 State" Process State Transition Diagram with illustration. [Dec 2013]

Q2. What is Process? Give the difference between a process and a program. [June 2010]

or

What is process? What are the different types of states Of any process? Explain different data structures to handle process management. [June, 2012]

Q3. Define Process. List the major events for creation of a process and explain them. [June 2013]

Q4. What is thread? Explain thread structure. [Dec 2010 ,June 2011]

Q5. Explain the classical thread model with its implementation strategies [June 2013]

Q6. Explain PCB. Discuss its major fields. [Dec, 2010] [June 2013]

or

What is Process Control Block? Explain various entries of it. [Dec 2013]

Q7. What is scheduler? Explain queuing diagram representation of process scheduler with figure. [June 2012].