

END TERM EXAMINATION
Fourth semester B tech MAY-JUNE,2014
OPERATING SYSTEM

Time:1 half hrs

Note: question no 1 is compulsory and attempt any two questions from the rest

Q.1.(a) Multiple choice questions:

(i) Spooling is an acronym for-

- (a) Simultaneous peripheral operation online
- (b) Simultaneous process operation online
- © system peripheral operation online
- (D) simultaneous peripheral operation online

(ii) which is the correct definition of a valid process transition within an OS-

- (A) Wakeup: ready->running
- (B) Non-preemptive,preemptive
- (C) Block: ready->blocked
- (D) Timer run out : ready->blocked

(iii) FCFS and RR scheduling are respectively:

- (A) Preemptive,non-preemptive
- (B) Non-preemptive,preemptive
- (C) Deadline,fair-share
- (D) Fair-share,deadline

(iv)To avoid race condition, the number of processes that may be simultaneously inside their critical condition is:

- (A) 0
- (B) 1
- (C) 2
- (D) 4

(v)The simplest directory structure is:

- (A) single level directory
- (B) two level directory
- (C) tree structure directory
- (D) none of the above

ANS:

- (i) Answer is (a) or (d)
- (ii) Answer is (a)
- (iii) Answer is (b)
- (iv) Answer is (b)
- (v) Answer is (a)

Q1(b) What are the difference between a soft real time system and hard realtime system?

Ans: (explain in detail:)

HARD REAL TIME SYSTEMS	SOFT REAL TIME SYSTEMS
1.They are time critical	They are not time critical
2.if timing is not perfect they are destroyed	They are not destroyed if there is some delay in time
3.The critical deadlines are met strictly missing a deadline means failure of the system and OS which support such hardware are called hard real time OS	Occasional missing of deadlines are acceptable and are not considered as failures, it is considered as performance degrade
4. ex: missile,rocket launching Heart pacemaker	Ex:cellphone

Q1(C)define monitor.

ANS. A monitor is an object or module intended to be used safely by more than one thread. The defining characteristic of a monitor is that its methods are executed with mutual exclusion.That is,at each point of time atleast one thread may be executing any of its methods.this mutual exclusion greatly simplifies reasoning about the implementation of monitors compared to reasoning about parallel codes that updates a data structure.

They also provide a mechanism for threads to temporarily giveup exclusive access, in order to wait for some condition to be met, before regaining exclusive access and resuming their task.

Q1(d) Explain belady's anomaly.

ANS. In computer storage, belady's anomaly proves that it is possible to have more page faults when increasing the number of page frames while using the FIFO page replacement algorithm. In common computer memory management, information is loaded in specific sized chunks. Each chunk is referred to as a page. The central processor can only load a limited number of pages at a time. It requires a frame for each page it can load. A page fault occurs when a page is not found, and might need to be loaded from disk into memory

When a page fault occurs and all frames are in use, one must be cleared to make room for the new page. A simple algorithm is FIFO whichever page has been in the frames the longest is the one that is cleared.it is also called FIFO anomaly.in short, sometimes the execution time increases even when more frames are allocated to the process.

Q 1 (e) What are the advantages of multi-programming?

ANS: Multiprogramming is the rapid switching of the CPU between multiple processes in memory...(explain in detail).it is commonly used to keep the CPU busy when one or more process are doing I/O. It makes efficient use of CPU. By overlapping the demands for the CPU and its I/O devices from various users.No matter how much I/O a program does, the CPU will be 100% busy. This ofcourse assumes the major delay is the wait while the data is being copied. A CPU should do work if the I/O were slow for other reasons.

Q1 (f) Briefly explain process control block.

Ans. PCB is the data structure containing certain information about the process. Each process has its own PCB to represent it in the OS.

A PCB may contain several types of the information depending upon the process to which PCB belongs

PCB contains the following..(explain in detail)

1. Process number
2. Priority
3. Process state
4. Program counter
5. CPU register
6. CPU scheduling information
7. Memory management information
8. Accounting
9. I/O status
10. File management

UNIT-I

Q2(a) Compare and contrast between the following types of operating system

- (i) Interacted
- (ii) Network
- (iii) Distributed

Ans. **Interactive System:** in this system the user and the computer interact or communicate with one another.

The user can enter data which is processed immediately or enter commands like 'save' or 'print'. The computer responds through dialog boxes asking the user

Where they would like to save the file or how they would like the file to be printed. This is similar to 'conversation'.

Network system: Most current OS are capable of using the TCP/IP networking protocols. This means that one system can appear on a network of the other are share resources such as files, printers and scanners using either wired or wireless connections. A network OS is most frequently used with LAN and WAN. But could also have applications to larger network systems. The upper 5 layers of the OSI reference Model provides the foundation upon which many network OS are based.

Some features of network OS:

1. Provide basic OS features such as support for processors, protocols, automatic hardware detection and support multi-processing of the apps.
2. Security features such as authentication, authorization, logon restrictions and access control.
3. Provide name and directory services.
4. Provide file, print, web services, backup and replication services
5. Support internetworking such as routing and WAN ports
6. They provide 2 tier client server architecture

Distributed systems:

It is a method of computer processing in which different parts of a program run simultaneously on two or more computers that are communicating with each other over a network. It is a type of parallel computing.

They provide n tier client server architecture

Types:

1. tightly coupled(explain in detail)
2. peer to peer(explain in detail)

Q2. (b) Explain demand paging. What do you mean by thrashing?

Ans:

when pure demand paging is used, page loading only occurs at the time of the data request, and not before. In particular, when demand pager is used a program usually begins execution with none of its pages pre load in the RAM. The pages are copied in the executable file into RAM the first time the executing code references them, usually in response to the page faults. As a consequence, pages of the executable file containing code not executed during a particular run will never be loaded in the memory. It is a type of swapping in which pages of the data are not copied from the disk into the RAM until they are needed.

Thrashing is a high paging activity. A process is thrashing if it is spending more time with paging rather than execution.

It is caused by:

1. global replacement algorithm
2. under allocation of the minimum number of pages required by the process
3. very high degree of multi programming

it is eliminated by:

1. reducing level or degree of multi programming
2. use local replacement algorithm
3. while allocating check minimum sufficient frame required for a process

Q 3. (a) consider the following sequence of memory reference from 460 word programmies:

10,11,104,170,73,309,185,245,434,438,364

- (i) give the reference string assuming a page size of 100 words.
- (ii) Find the number of page faults for the above reference string assuming 200 words primary memory available to the program using LRU replacement algorithm

Ans: (a) reference string is :0,0,1,1,0,3,1,2,4,4,3

Primary memory 200 bytes

(b) LRU-> 7 page faults

Q 3. (b)Describe the implementation of segmentation with a suitable example.

Ans. Memory segmentation is the division of computer's primary memory into segments or sections.in a computer system using a segmentation, a reference to a memory location included a value that identifies a segment and an offset within that segment. segments or

section are also used in object files of compiled programs when they are linked together into a program image and when the image is loaded into memory. Different segments may be created for different program modules, or for different classes of memory usage such as code and data segments. Certain segments may even be shared between the program.

Unit II

Q4. What are CPU schedulers? Describe various types of schedulers.

Ans. In concurrent programming a critical section is a piece of code that accesses a shared resource that must not be concurrently accessed by more than one thread of execution. A critical section will usually terminate in fixed time to enter it. Some synchronization mechanism is required at the entry and exit of the critical section to ensure exclusive use, for example a semaphore.

Critical section

1. Set of instructions that must be controlled so as to allow exclusive access to one process.
2. Execution of the critical section by processes is mutual exclusive in time.

Three types of scheduling are as below :

1. Short- term (CPU scheduler) (explain in detail)
2. Medium - term (memory manager):(explain in detail)
3. Long- term (job scheduler) (explain in detail)

A primary difference is in the frequency of their execution. The short-term scheduler must select a new process quite often. Long term is used much less often since it handles placing jobs in the system and may wait a while for a job to finish before it admits another one.

Q5 (a) What do you mean by a critical section? Using semaphores , write a solution to readers and writers problem that gives priority to readers.

Ans. Critical section is the execution part of any process. any process can only enter its critical section when any other process is not executing in its critical section. No two processes should execute their critical section simultaneously to avoid deadlock.

1. Deadlock prevention(explain in detail)
2. Deadlock avoidance(explain in detail)
3. Deadlock detection and recovery(explain in detail)

The readers/writers problem:

A data object is to be shared among various concurrent processes some of these may want only to read the content of the object, while others may want to update the shared the shared object the former set of processes is called readers whereas the later is called the writers. The above synchronization is called readers / writers problem. The first solution requires that no reader is kept waiting unless a writer has already have obtained permission to use the shared object. This requirement give priority to reader over writers. The second solution requires that the writer has a priority over readers and do not wait unless other readers or writers have already obtained permission to use the shared object.

Semaphore mutex, wrt;

```

Int readcount;
Reader:
Wait(mutex)
Readcount++;
If(readcount== 1) wait(wrt);
//first reader locks semaphore
Signal(mutex);
//Read the data
Wait(mutex);
Readcount--;
If(readcount=0) signal(wrt);
//last reader signals
Signal(mutex);

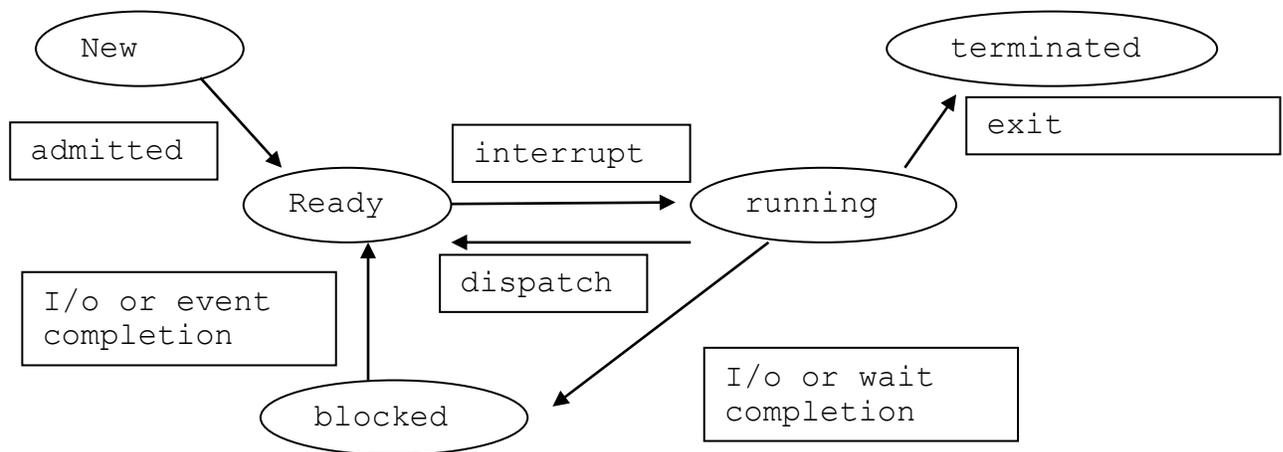
```

```

Writer:
Wait(wrt);
Write the dat;
Signal(wrt);

```

Q5 (b) Explain the process states by using processes state transition diagram
 Ans.



UNIT III:

Q6(a) What is deadlock? List 4 necessary conditions for the occurrence of deadlock. Is it possible to have a deadlock having one process only? Explain your answer.

Ans. A set of processes is in a deadlock state if each process in the set is waiting for an event that can be caused by only another process in the set. In other words, each member of the set of deadlock processes is waiting for a resource that can be released only by a deadlock process. None of the processes can run, none of them can release any resources, & none of them can be awakened. It is important to note that the number of processes & the number & kind of resources possessed & requested are unimportant.

The resource may be either physical or logical.

Necessary conditions for deadlock:

1. Mutual exclusion condition: the resources involved are non-shareable
2. Hold & wait condition: Requesting process holds already resources while waiting for requested resources.
3. No-Preemptive condition: Resources already allocated to a process cannot be preempted.
4. Circular wait condition: The processes in the system form a circular list or chain where each process in the list is waiting for a resource held by the next process in the list.

It is not possible to have a deadlock with 1 process only but it is possible to have a deadlock with multiple threads of a single process.

Q6(b). Explain Banker's Algorithm by using a suitable example.

A. The Banker's Algorithm is a resource allocation & deadlock avoidance algorithm that tests for safety by simulating the allocation of predetermined maximum possible amounts of all resources, & then makes a "safe state" check to test for possible deadlock conditions for all other pending activities, before deciding whether allocation should be allowed to continue. (explain example using the algorithm in detail)

Q7. Explain the following terms in disk scheduling.

(a) Seek time (b) Rotational latency (c) Disk bandwidth

A. The operating system is responsible for using hardware efficiently for the disk drives, this means having a fast access time & disk bandwidth.

-Access time has 2 major components

-Seek time is the time for the disk heads to move to the cylinder containing the desired sector.

-Rotational latency is the additional time waiting for the disk to rotate the desired sector to the disk head.

-Minimize seek time

-seek time \rightarrow seek distance

-disk bandwidth is the total no. of bytes transferred, divided by the total time

between the first request for service & the completion of the last transfer.

Unit 4:

Q8(a) Explain any 3 allocation schemes that exist for allocating secondary storage to files.

A. There are 3 methods: contiguous, chained & indexed.

With contiguous allocation a single contiguous set of blocks is allocated to a file at the time of file creation. Thus, this is a reallocation strategy, using variable-size portions. The file allocation table needs just a single entry for each file, showing the starting block & length of the file. It is best for sequential file. The some problems in this external fragmentation will occur, it will be necessary to perform a compaction algorithm to free up additional space on the disk, & also it is necessary to perform a compaction algorithm to free up additional space on the disk, & also it is necessary to declare the size of the file at the time of the file at the time of creation.

The next method is chained allocation. Each block contains pointer to the next block in the chain. The file allocation table needs a single entry for each file, showing the starting block & the length of the file. Although reallocation is possible, it is more common simply to allocate blocks as needed. Any free block can be added to a chain. NO external fragmentation.

Indexed allocation addresses many of the problems of contiguous & chained allocation. In this, the file allocation table contains a separate 1 level index for each file; the index has 1 entry for each portion allocated to the file. Typically, the file indexes are not physically stored as part of the file allocation table, rather in a separate block & the entry for the file in the file allocation table points to that block.

Allocation may be on the basis of either fixed-size blocks or variable-size portions. Allocation by blocks eliminates external fragmentation, whereas allocation by variable size portions improves locality. Indexed allocation supports both sequential & direct access to the file.

Q8(b). What is directory? What are the different ways to implement a directory?

A. A directory is a symbol table, which can be searched for information about the files. Also, it is the fundamental way of organizing files. Usually, a directory is itself a file. A typical directory entry contains information about a file. Directory entries are added as files are created & are removed when files are deleted.

Common directory structures are:

- Single-level: shared by all users
- Two-level: one level for each user
- Tree: arbitrary tree for each user

The files & directories at any level are contained in the directory above them.

To access a file, the names of all the directories above it need to be specified.

The topmost directory in any file is called the root directory. A directory that is below another directory is called a subdirectory. A directory above a subdirectory

is called the parent directory.

Directory implementation:

1. Linear list:

-a linear list is the simplest & easiest directory structure to set up.

-finding a file requires linear search.

-deletions can be done by moving all entries, flagging an entry as deleted or by moving the last entry into the newly vacan position.

-sorting the list makes searches faster, at the expense of more complex insertions & deletions

-a linked list makes insertions & deletions into sorted list easier.

2. Hash table:

-a hash table can also be used to speed up searches

-hash table are generally implemented in addition to a linear or other structure

Q9. Write short notes on the following:

(a)Basic file system

A. The basic file system level works directly with the device drivers in terms of retrieving & storing raw blocks of data, without any consideration for what is in each block.

Depending on the system, blocks may be referred to with a single block number, or with head sector cylinder combinations .

The basic file system needs only to issue generic commands to the appropriate device driver to read & write physical blocks on the disk. Each physical block is identified by its numeric disk address.

(b) Logical file system

A. The logical file system deals with all of the metadata associated with a file i.e. everything about the file except the data itself. This level manages the directory structure to provide the file name.It maintains file structure via file control blocks , FCBs, which contain all of the metadata which includes information about file, including ownership, permissions & location of the file contents as well as block number information for finding the data on the disk.

(c) File access mechanism.

A.

1. Contiguous allocation: This method requires each file to occupy a set of contiguous address on the disk. Disk address defines a linear ordering on the disk. The difficulty with contiguous allocation is finding space for a new file. It also suffers from external fragmentation.

2. Linked allocation: In linked allocation, each file is a linked list of disk blocks.

The directory contains a pointer to the first block of the file. There is no external fragmentation with linked allocation. Any free block is used to satisfy a request. The problem with it is that it is inefficient to support direct-access; it is effective only for sequential-access files.

3. Indexed allocation: This allocation method is the solution to the problem of both the above allocation. This is done by bringing all the pointers together into one location called the index block. Each file has its own index block, which is an array of disk sector addresses. The directory contains the address of the index block of a file.

(d) Free space management.

A. free space management :

another important aspect of disk management is keeping track of and allocating free space bit vector:

1. One simple approach is to use a bit vector, in which each bit represents a disk block, set to 1 if free or 0 if allocated.
2. Fast algorithms exist for quickly finding contiguous blocks of a given size
3. The downside is that a 40 GB disk requires over 5 MB just to store the bitmap

Linked list:

1. A linked list can also be used to keep track of the free blocks.
2. Traversing the list and/or finding a contiguous block of a given size are not easy, but fortunately are not frequently needed operations. Generally the system just adds and removes single blocks from the beginning of the list
3. The FAT table keeps track of the free list as just one more linked list on the table.

Grouping:

A variation on linked list free list is to use links of blocks of indices of the free blocks. If a block holds up to N addresses, then the first block of the linked list contains up to N-1 addresses of free blocks.

Counting:

When there are multiple contiguous blocks of free space then the system can keep track of the starting address of the group and the number of contiguous free blocks. As long as the average length of a contiguous group of free blocks is greater than two this offers a savings in space needed for the free list.