

## B.Tech.

Fifth Semester Examination

### Principles of Operating Systems (CSE-301-F)

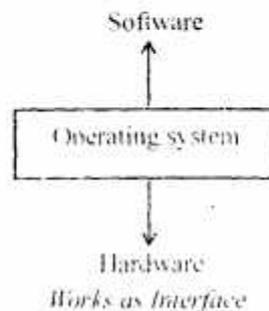
---

Short Questions :

Q. 1. (a) What do you mean by Operating System?

**Ans. Operating System :** An operating system is the most important program in a computer system. This is one program that runs all the time. As long as computer is operational and exists only when the computer is shut down.

Operating systems are programs that act as interface between the user and computer hardware. They sit between the user and hardware of computer providing an operational environment to the users and application programs. For a user, therefore a computer is nothing but the operating system running on it. It is extended machines.



Q. 1. (b) What do you mean by multiprogramming system?

**Ans. Multiprogramming Operating System :** The objective of a multiprogramming operating system is to increase the system utilization efficiency. The batch processing system tries to reduce the CPU idle time through operator interaction. However, it cannot reduce the idle time due to input/output operations. So, when some input/output is being performed by the currently executing job of a batch, the CPU sits idle without any work to do. Thus, the multiprogramming operating system tries to eliminate such idle times by providing multiple computational tasks for the CPU to perform. This is achieved by keeping multiple jobs in the main store.

Some most popular multiprogramming operating system are : Unix, Windows NT etc. A multiprogramming supervisor has a very difficult job of managing all activities that take place simultaneously in the system. He has to monitor many different activities and react to a large number of different situations in course of working.

Q. 1. (c) What are the various services of operating system.

**Ans. Various Services of Operating System :** An operating system provides an environment for the execution of programs. The operating system provides certain services to programs and to the users of those programs. The specific services provided will differ from one operating system to another. These operating system services are provided for convenience of programmer, to make programmer task easier. Some of operating system services are given below :

- (i) Program execution services.
- (ii) Input/output operation services.

- (iii) File system services.
- (iv) Communication services.
- (v) Memory management services.

**Q. 1. (d) Write name of 5 operating system.**

**Ans.** Five operating systems names :

- (i) Windows 95/98/2000/xp/2003
- (ii) MS-DOS
- (iii) Unix/Linux
- (iv) OS/2
- (v) Ubuntu

**Q. 1. (e) What is spooling?**

**Ans. Spooling :** Spooling refers to putting jobs in a buffer, a special area in memory or on a disk where a device can access them when it is ready. Spooling is useful because devices access data at different rates. The buffer provides a waiting station where data can rest while slower devices catches up.

The most common spooling application is print spooling. In print spooling, documents are loaded into a buffer and then the printer pulls them off the buffer as its own rate. Because the documents are in a buffer where they can be accessed by the printer we can perform other operations on computer while the printing takes place in the background spooling also lets and place a number of print jobs on a queue instead of waiting for each one to finish before specifying next one.

**Q. 1. (f) What is race condition in thread or process?**

**Ans. Race Condition in Thread or Process :** A race condition occurs when two threads access a shared variable at the same time. The first thread reads the variable and second thread reads the same value from the variable. Then the first thread and second thread perform their operations on the value and they race to see which thread can write the value last to the shared variable. The value of thread that writes its value last is preserved, because thread is waiting over the value that previous thread wrote.

**Example :**

```
Thread 1
total = total + val 1
Thread 2
total = total - val 2
```

**Q. 1. (g) What do you mean by deadlock?**

**Ans. Deadlock :** A specific condition when two or more processes are each waiting for another to release a resource or more than two processes are waiting for resources in a circular cha... Deadlock is a common problem in multiprocessing where many processes share a specific type of mutually exclusive resource known as a software or soft lock. Computers intended for time sharing and/or real time markets are often equipped with a hardware lock which guarantees exclusive access to processes, forcing serialization. Deadlocks are particularly troubling because there is no general solution to avoid deadlocks.

**Q. 1. (h) Discuss the first fit, best fit and worst fit memory scheme.**

**Ans. First Fit :** Search for the first hole big enough from either and if found, allocate this hole to the process else wait for running processes to finish.

**Best Fit :** Allocate the smallest hole big enough for process. The whole of the list of free holes must be

searched unless the holes are maintained in some sorted order.

**Worst Fit :** Allocate the largest hole to process. The whole list of free holes must be searched unless the holes are maintained in some sorted order.

**Q. 1. (i) Write short note on NFS.**

**Ans. NFS :** Network File System, originally developed by Sun Microsystems, is a client/server application that lets a computer user view and optionally store and update file on a remote computer as though they were on the user's own computer. It has the great advantage that it is independent of host operating system and can provide data sharing among different types of systems (heterogenous systems). The user's system needs to have an NFS client and other computer needs the NFS server. Both of them require TCP/IP installed since the NFS server and client use TCP/IP as program that sends the files and updates.

**Q. 1. (j) What is Internal Fragmentation?**

**Ans. Internal Fragmentation :** Internal fragmentation occurs when storage is allocated without ever intending to use it. This space is wasted. While this seems foolish, it is often accepted in return for increased efficiency or simplicity. The term "internal" refers to the fact that the unusable storage is inside the allocated regions.

For example, in many file systems, files always start at the beginning of a sector, because this simplifies organization and makes it easier to grow files. Any space left over between the last byte of the file and first byte of the next sector is internal fragmentation. A program which allocates a single byte of data is often allocated many additional bytes of metadata and alignment. This extra space is also internal fragmentation.

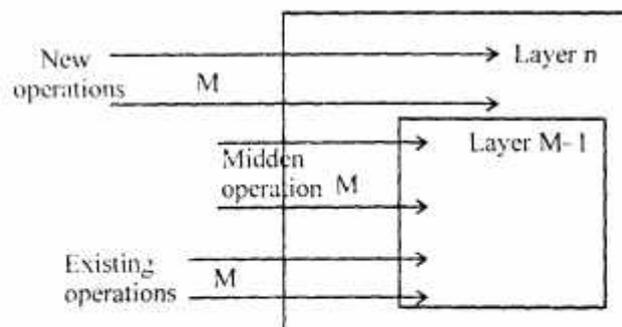
#### Section-A

**Q. 2. (i) Describe Layered Approach operating systems architecture.**

**Ans. Layered Approach of Operating System Architecture :** There are many operating system architecture available but layered approach is more basic and simple approach.

The modulation of a system can be done in many ways : the most appealing is the layered approach, which consists of breaking the operating system into a number of layers (levels), each built on top of lower layers. The bottom layer (layer 0) is the hardware, highest (layer N) is the user interface.

An operating system layer is an implementation of an abstract object that is the encapsulation of data and operations that can manipulate those data. A typical operating system layer-say layer M is depicted in figure given below :



The layer approach to design was first used in the 'THE' operating system. The THE system was defined in six layers.

Layer 5 : User programs
Layer 4 : Buffering for input and output devices
Layer 3 : Operator console device driver
Layer 2 : Memory Management
Layer 1 : CPU scheduling
Layer 0 : Hardware

**Q. 2. (ii) Write down short note on shell.**

**Ans. Shell :** A shell is a piece of software that provides an interface for users (command line interpreter). Typically, the term refers to an operating system shell which provides access to the services of a kernel. However the term is also applied very loosely to applications and may include any software that is "built around" a particular component such as web browsers and email clients that are 'shells' for HTML rendering engines. The name 'shell' originates from shell being an outer layer of interface between the user and inwards of the operating system (the kernel).

Operating system shells generally fall into one of two categories: command line and graphical. Command line shells provide a Command Line Interface (CLI) to the operating system, which graphical shells provide a GUI. The relative merits of CLI and GUI based shells are often debated. CLI components claim that certain operations can be performed much faster under CLI shells than under GUI shells. However GUI proponents advocate the comparative usability and simplicity of GUI shells. The best choice is often determined by the way in which a computer will be used. On a server mainly used for data transfers and processing with expert administration, a CLI is likely to be the best choice.

**Q. 3. (i) What do you mean by context switching and also explain schedules.**

**Ans. Context Switching :** Switching the CPU to another process requires saving the state of old process and loading the saved state for the new process. This task is known as a context switch. Context switch time is pure overhead, because the system does no useful work while switching. Its speed varies from machine to machine, depending on the memory speed, the number of registers which must be copied and existence of special instructions.

Context switch times are highly dependent on hardware support for instance, some processors provide multiple sets of registers. A context switch simply includes changing the pointer to the current register set.

**Scheduler :** A process migrates between the various scheduling queues throughout its lifetime. The operating system must select, for scheduling purposes, processes from these queue in some fashion. The selection process is carried out by appropriate scheduler. Scheduler is a mechanism that carries out of scheduling activities.

**Q. 3. (ii) Explain Non-preemptive scheduling in details.**

**Ans. Non-preemptive Scheduling :** These scheduling mechanism do not force the process to be deallocate from CPU to release the resource as and when required. The processes release the resources only when it has finished executing.

There are many non-preemptive scheduling algorithm :

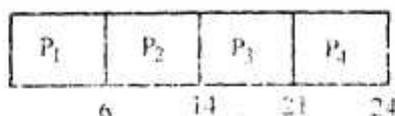
- (i) FCFS (First Come First Serve)
- (ii) SJF (Shortest Job First)
- (iii) Priority scheduling.

**FCFS :** First come first served algorithm. In this scheduling algorithm, which process comes first in ready queue, executes first.

**Example:**

Process	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>
Burst time	6	5	7	3

**Gantt Chart:**



P<sub>1</sub> comes first, then P<sub>1</sub> executes first after that P<sub>2</sub>, P<sub>3</sub> and P<sub>4</sub> executes. When one process complete, then next process comes and executes.

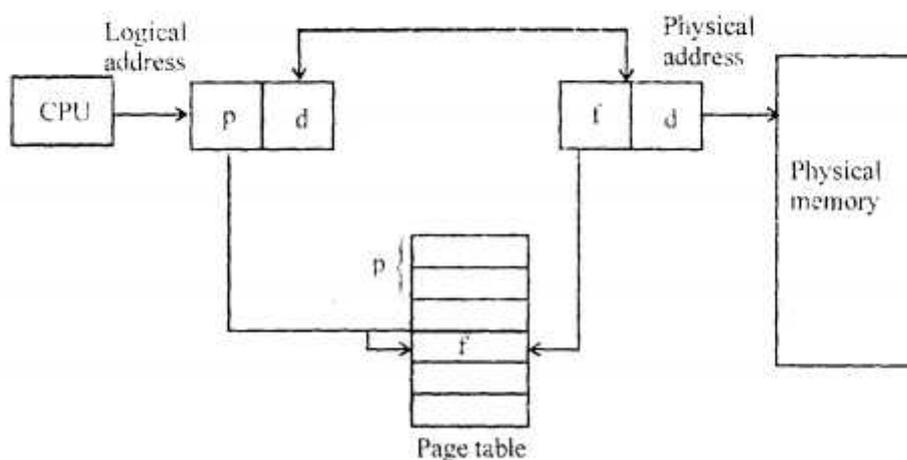
**Section-B**

**Q. 4. Explain paging in detail?**

**Ans. Paging :** A possible solution to the external fragmentation problem is to permit the logical address space of a process to be non-contiguous thus allowing a process to allocated physical memory whenever the latter is available.

One way of implementing this solution is through the use of a paging. Paging avoids the considerable problem of fitting the varying sized memory chunks onto the backing store, from which most of previous memory management schemes suffered when some code fragments or data residing in the main memory need to be swapped out, space must be found on backing store. Paging in its various forms is commonly used in many operating systems.

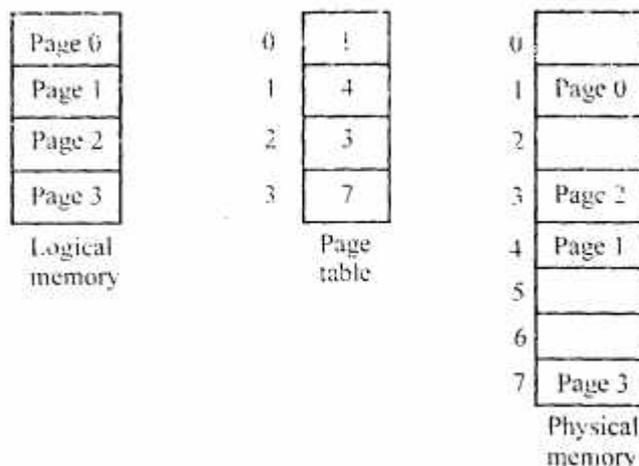
Physical memory is broken into fixed sized blocks called frames. Logical memory is also broken into blocks of same size called pages. When a process is to be executed, its pages are loaded into any available memory frames from the backing store. The backing store is divided into fixed sized blocks that are the same size as memory frames.



*Paging hardware*

The hardware support for paging is illustrated in figure every address generated by the CPU is divided

into two parts : a page number (P) and a page offset (d). The page number is used as an index into a page table. The page table contains the base address of each page in physical memory. This base address combined with page offset to define the physical memory address that is sent to the memory unit.



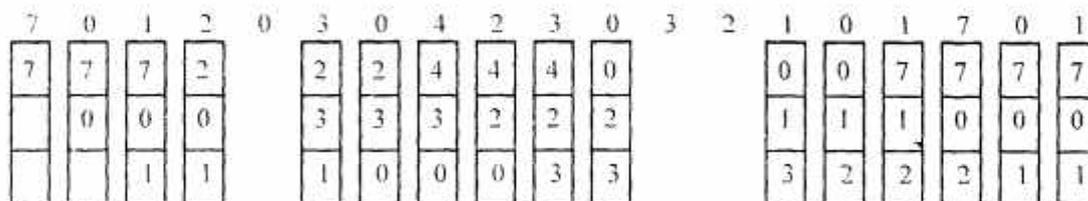
*Paging Model of Logical and Physical Memory*

**Q. 5. Discuss FIFO algorithm of page replacement technique and what Belady's Anomalies?**

**Ans.** The simplest page-replacement algorithm is a FIFO algorithm. A FIFO replacement algorithm associates with each page the time when that page was brought into memory. When a page must be replaced, the oldest page is selected. We can create a FIFO queue to record the time when a page is brought into the memory. We can create a FIFO queue to hold all pages in memory. We replace the page at the head of queue. When a page is brought into memory, we insert it at the tail of the queue.

Reference string,

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1



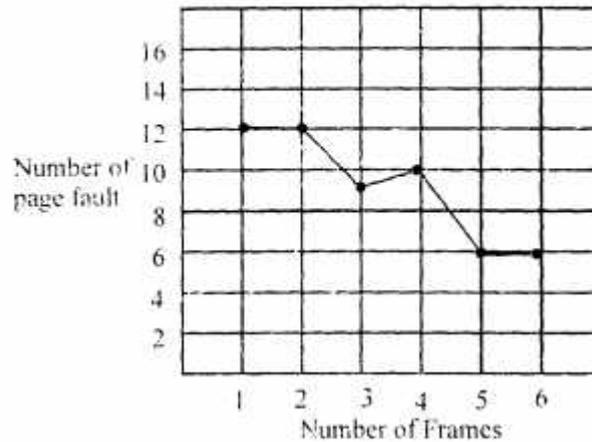
*FIFO Page Replacement Algorithm*

**Belady's Anomalies :** To illustrate the problems that are possible with a FIFO page-replacement algorithm, let us consider the reference string.

1, 2, 3, 4, 1, 5, 1, 2, 3, 4, 5

Figure shows the curve of page faults versus the number of available frames. We notice that the number of faults for 4 frames (= 10) is greater than number of faults for three frames (=9)! This result is most unexpected and is known as belady's anomaly. Belady's Anomaly reflects the fact that, for some page-replacement

algorithm, the page fault rate may increase as the number of allocated frame increases.



### Section-C

**Q. 6. Explain the file system in detail. Also define the various kind of file systems.**

**Ans. File System :** File system is a method for storing and organizing computer files and the data they contain to make it easy to find and access them. File systems may use a storage device such as a hard disk or CD-ROM and involve maintaining the physical location of the files, they might provide access to data on a file server by acting as clients for a network protocol (e.g., NFS, SMB or IP clients) or they may be virtual and exist only as an access method for virtual data.

**Types of File Systems :** File system types can be classified into disk file systems, network file systems and special purpose file systems.

**Disk File Systems :** A disk file system is a file system designed for the storage of files on a data storage device, most commonly a disk drive, which might be directly or indirectly connected to the computer. Examples of disk file system includes FAT, NTFS, HFS and UDF.

**Database File System :** A new concept for file management is the concept of a database based file system. Instead of hierarchical structured management, files are identified by their characteristics like type of file, topic author or similar metadata.

**Network File Systems :** A NFS is a file system that acts as a client for a remote file access protocol, providing access to files on a server. Examples of NFS include clients for NFS, SMB, AFP and IP protocols and file system like clients for FTP and Web DAV.

**Special Purpose File Systems :** A special purpose file system is basically any file system that is not a disk file system or network file system. This includes systems where the files are arranged dynamically by software, intended for such purposes as communication between computer process or temporary file space.

**Q. 7. (i) What do you mean by deadlock? And also discuss all necessary conditions for deadlock.**

**Ans. Deadlock :** A specific condition when two or more processes are, each waiting for another to release a resource or more than two processes are waiting for resources in a circular chain. Deadlock is a common problem in multiprocessing where many processes share a specific type of mutually exclusive resource known as a software or soft lock.

**Necessary Conditions for Deadlock :**

- (i) **Mutual Exclusion Conditions :** A resource is either assigned to one process or it is available.
- (ii) **Hold and Wait Condition :** Processes already holding resources may request new resources.
- (iii) **No preemptive Condition :** Only a process holding a resource may release it.
- (iv) **Circular Wait Condition :** Two or more processes form a circular chain where each process waits for a resource that the next process in chain holds.

Deadlock only occurs in systems where all 4-conditions happen

**Q. 7. (ii) Explain Reader/Writers problem with its solution by using semaphores.**

**Ans. Readers/Writers Problem :** Two kinds of processes-readers and writers share a database. Readers execute transactions that examine database records. Write transactions both examine and update database. The database is assumed initially to be in a consistent state. Each transaction, if executed in isolation, transforms the database from one consistent state to another.

**Formal Readers and Writers Solution Using Semaphores :**

**Semaphore Invariant :** For semaphores, let  $nP$  be number of completed P operations and let  $nV$  be number of completed V operations. If  $init$  is initial value of S, then in all visible program states,  $nP \leq nV + init$

Consequently execution of a P operation potentially delays until an adequate number of V operations have been executed.

By using this definition of semaphores the readers and writers problem can be solved in the following ways :

```
integer n Readers := 0
semaphore semReaders := 1
Semaphore semWriters := 1
Reader [ i = 1 ... m ] :: do true :
    P (sem Readers)
    nReaders := nReaders + 1
    if nReaders = 1 - P (semwriters) fi
    V (semReaders)
    read database
    P (Sem Readers)
    nReaders := nReaders - 1
    if nReaders = 0 - V (SemWriters) fi
    V (SemReaders)
and
writer [ j : 1 ... n ] :: do true ->
P (semwriters)
write the database
V (semWriters)
```

od.

**Section-D**

**Q. 8. (i) Explain blocking and non-blocking input/output.**

**Ans. Blocking and Non-Blocking Input/Output :** Asynchronous input/output or non-blocking input/output, is a form of input/output processing that permits other processing to continue before the transmission has finished input and output operations on a computer can be extremely slow compared to the processing of data. An input/output device incorporate mechanical devices which moves physically, such as a hard drive seeking a track to read or write; extremely slow. For example, during a disk operation that takes ten milliseconds to perform, a process that is clocked at one gigahertz could have performed ten million instruction processing cycles. A simple approach to input/output would be start the access and then wait for it to complete.

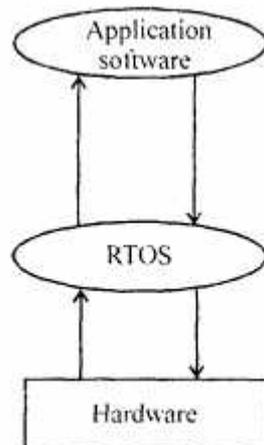
But such approach (called synchronous input/output or blocking input/output) would block the progress of a program while the communication is in progress leaving system resources idle. When a program makes many input/output operations, this means that the processor can spend almost all of its time idle waiting for input/output operations to complete.

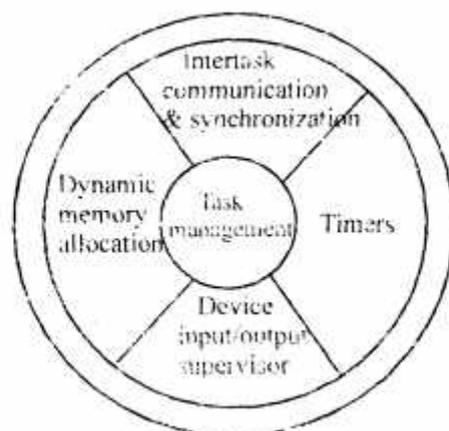
Alternatively it is possible, but more complicated to start the communication and then perform processing that does not require that input/output has completed. This approach is called asynchronous input/output.

**Q. 8. (ii) Define kernel in Real time operating system.**

**Ans. Kernel :** The kernel is the central component of most computer operating systems. Its responsibilities include managing the systems resources and the communication between hardware and software components. As a basic component of an operating system, a kernel provides the lowest level abstraction layer for the resources (especially memory, processors and input/output devices) that applications must control to perform their function. It typically makes these facilities available to application processes through interprocess communication mechanisms and system calls.

**Real Time Operating System Kernel :** The kernel of a real time operating system ("RTOS") provides an "abstraction layer" that hides from application software the hardware details of the processor (or set of processors) upon which application software will run.



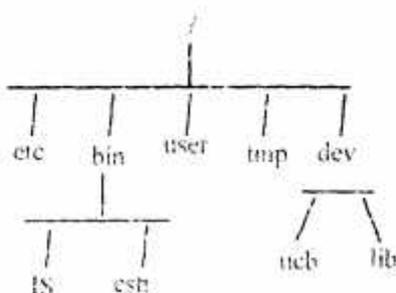


*Basic Services Provided By a Real Time Operating System Kernel*

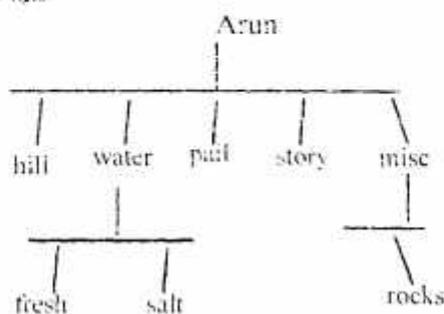
**Q. 9. Explain Unix file system, And also define some unix command.**

**Ans. Unix File System :** The Unix File System (UFS) is a file system used by many unix and unix like operating systems. It is also called Berkeley Fast File System, the BSD fast file system of FFS. It is a distant, descendant of original file system used by version 7 unix.

In unix, the files are organized into a tree structure with a root named by the character '%'. The first few levels of the tree look like this.



User files form a subtree of this tree. For example in many systems the user files are subdirectories of a directory named 'home' within 'usr'. if we had users Ajay and Arun, for example Ajay's home directory would be 'user/home/ajay' and all his files would be within that subtree, and analogous statement would hold for Arun. Suppose Arun's director look like this



File names can be given either in relative terms or with full path names. Look at the file 'salt' above. If we are in the directory 'water', we can refer to this file simply as salt.

If we are in directory above i.e., the one named 'aron' then we must write.

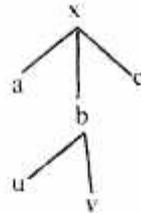
water/salt

**File Types :** There are 4 types of files in the UFS :

- (i) Ordinary files
- (ii) Directory files
- (iii) Device files
- (iv) Link files.

(i) **Ordinary Files :** An ordinary file may contain text, a program or other data.

(ii) **Directory Files :**



If directory structure above like this, then suppose if we talking about b, b is directory contain files u and v and b also has information about u and v.

(iii) **Device Files :** In unix, physical devices (printers, terminals etc.) are represented by "Files."

read( ) → for read file

write( ) → for write file

(iv) **Link Files :**

**Example :**

**In XY :** If we run ln, it will appear that a new file, Y has been created, as a copy of X, as if we had typed CP XY but CP does not create new file.