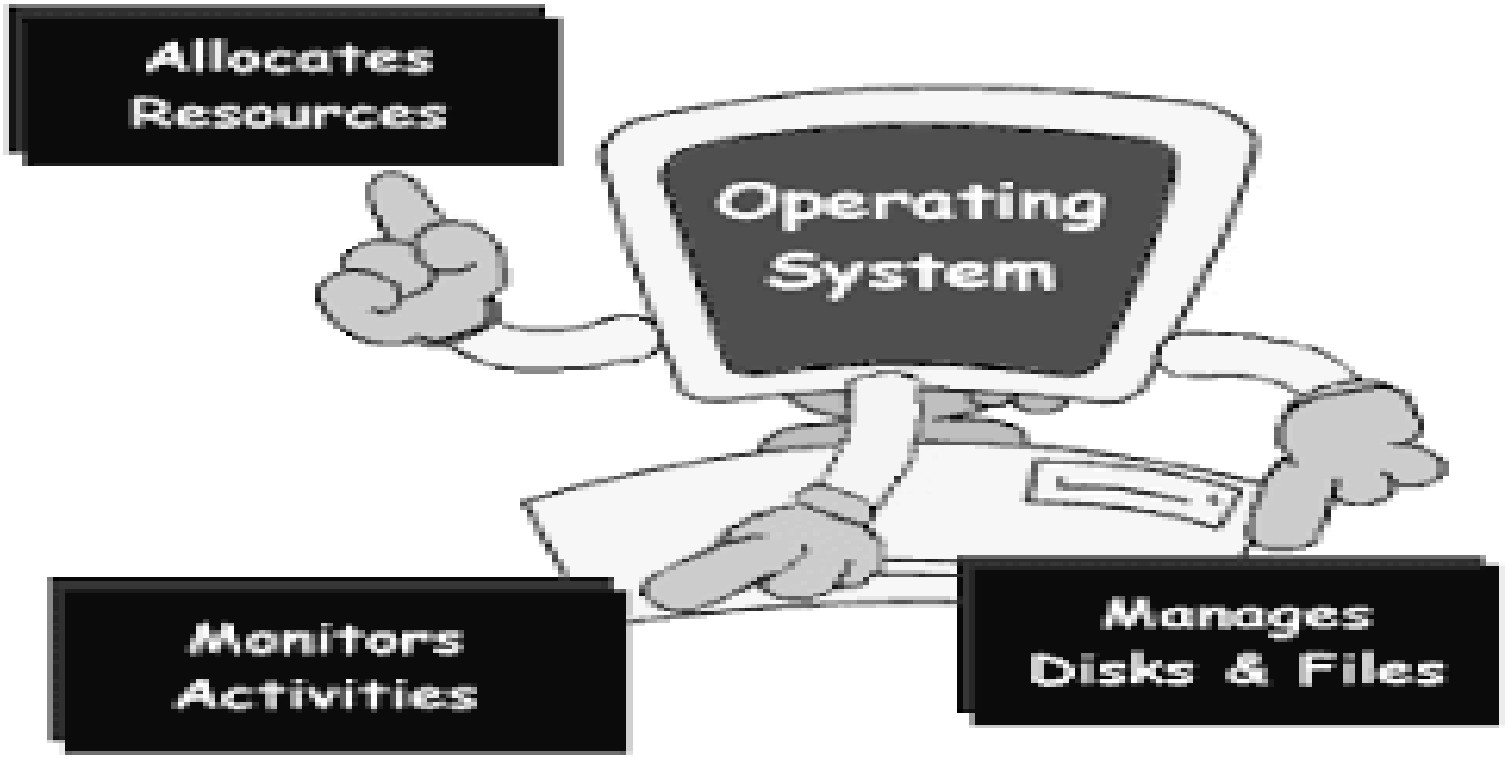
OPERATING SYSTEM



DAYANANDA V N
Lecturer in Computer Science and Engg.
Acharya polytechnic, Bangalore

UNIT 1

INTRODUCTION TO OPERATING SYSTEM

SYLLABUS

1. What Operating Systems Do.
2. Operating System Structure.
3. Operating System Operations.
4. Process Management.
5. Memory Management.
6. Storage Management.
7. Protection and Security.
8. Distributed System.
9. Special Purpose System.
10. Computing Environment.
11. Open-Source Operating Systems.

SYNOPSIS

Objectives: To provide coverage of basic computer system organization.

What is an Operating System.

A program that acts as an intermediate between a user of a computer and the computer hardware Operating system goals are

1. Execute user programs and make solving user problems easier.
2. Make the computer system convenient to use.
3. Use the computer hardware in an efficient manner.

1. What Operating System Do.

Depends on the point of view, users want convenience, **ease of use,** Don't care about **resource utilization.** But shared computer such as **mainframe** or **minicomputer** must keep all users happy. Users of dedicate systems such as **workstations** have dedicated resources but frequently use shared resources from **servers**. Handheld computers are resource poor, optimized for usability and battery life.

2. Computer - System Architecture

Most systems use a single general-purpose processor (PDAs through mainframes). Most systems have special-purpose processors as well. **Multiprocessors** systems growing in use and importance. Also known as **parallel systems, tightly-coupled systems.**

Advantages include: **1. Increased throughput.**

2. Economy of scale.

3. Increased reliability - graceful degradation or fault tolerance

There are Two types:

1. Asymmetric Multiprocessing.

2. Symmetric Multiprocessing.

3. Operating System Structure.

Multiprogramming needed for efficiency single user cannot keep CPU and I/O devices busy at all times. Multiprogramming organizes jobs (code and data) so CPU always has one to execute subset of total jobs in system is kept in memory.

Time sharing (multitasking) is logical extension in which CPU switches jobs so frequently that users can interact with each job while it is running, creating **interactive** computing.

4. Operating System Operations.

Dual-mode operation allows OS to protect itself and other system components are, **User mode** and **kernel mode. Mode bit** provided by hardware provides ability to distinguish when system is running user code or kernel code.

5. Process Management.

A process is a program in execution. It is a unit of work within the system. Program is a passive entity, process is an active entity.

6. Memory Management.

Memory management activities are Keeping track of which parts of memory are currently being used and by who. Deciding which processes (or part there of) and data to move into and out of memory. Allocating and deallocating memory space as needed.

7. Storage Management.

OS provides uniform, logical view of information storage. Abstracts physical properties to logical storage unit is called-file. Each medium is controlled by device (i.e., disk drive, tape drive), Varying properties include access speed, capacity, data-transfer rate, access method (sequential or random).

8. Protection and Security.

Protection - any mechanism for controlling access of processes or users to resources defined by the OS. **Security** - defence of the system against internal and external attacks. Huge range, including denial-of-service, worms, viruses, identity theft, theft of service.

9. Distributed Computing.

Collection of separate, possibly heterogeneous, systems networked together Network is a communications path.

10. Special-Purpose Systems.

Real-time embedded systems-most prevalent form of computers, Vary considerable, special **purpose**, limited purpose OS, **real-time OS.**

Multimedia systems-Streams of data must be delivered according to time restrictions Hand held system - PDAs, smart phones, limited CPU, memory, power Reduced feature set OS, limited I/O

11. Computing Environments.

Client-Server Computing, Dumb terminals supplanted by smart PCs, Many systems now servers, responding to requests generated by **clients.**

Compute-server provides an interface to client to request services (i.e., database)

File-server provides interface for clients to store and retrieve files.

Peer-to-Peer Computing

P2P does not distinguish clients and servers, Instead all nodes are considered peers, May each act as client, server or both, Node must join P2P network, Registers its service with central lookup service on network. Broadcast request for service and respond to requests for service via **discovery protocol.**

Web-Based Computing

Web has become ubiquitous PCs most prevalent devices. More devices becoming networked to allow web access. New category of devices to manage web traffic among similar servers: load balancers.

12. Open-Source Operating Systems

Operating systems made available in source-code format rather than just binary **closed-source.** Counter to the **copy protection** and **Digital Rights Management (DRM) movement.**

5 Marks Questions and Answers

1. Explain OS in user's view and systems view.

Ans. A user view of a computer varies according to the interface being used.

Example: A PC consist of a monitor, keyboard, mouse and system unit. In this case performance is important to the user rather than resource utilization, these system are designed for single user, suppose a user sits at a terminal connected to a main frame or mini computers, the other terminal access the same computer (main frames or mini computer) The OS in such cases is designed to maximize resource utilization i.e availability of CPU time, memory and I/O are used efficiently.

In some cases users sit at workstation connected to networks of other workstations and servers. Here users must share resources such as networking and servers like computer and print servers,. So when designing the OS there should be a compromise b/n of the user and resource utilization.

- Handheld systems are designed for individual users they are connected to network using wire or wireless modem.

- Operating systems are designed for individual users but performance amount of battery life is important.
- In embedded system OS are designed to run without user interception.

Example: In automobiles may have numeric keypads and turn indicator light on or off to show status.

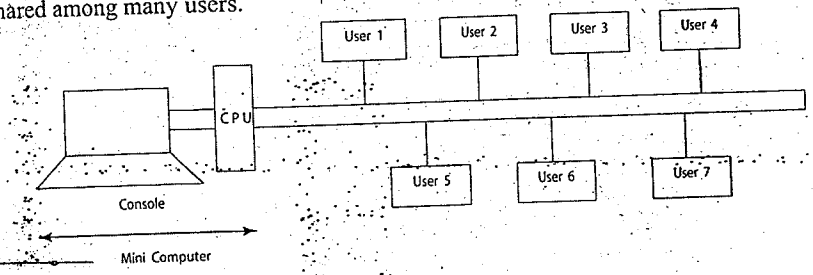
System view of OS

The OS can be viewed as a resource allocator. A computer system has many resources that may be required to solve a problem of CPU time, memory space, "OS is a controlled program which prevent the errors and improper use of the computer". OS is a resource allocator and control programmer.

2. Discuss time sharing system.

Time sharing is also known as logical extension of multiprogramming. It also refers to multiprogramming. The goal of time sharing system is to provide good response time for the user.

- It is developed to provide interactive use of computer system at a reasonable cost. In these system, each user has at least one separate program in memory.
- A time shared OS allows many user to share the computer simultaneously. A little time of CPU is allocated for each user.
- A CPU switches from one user's program to another, so frequently that the user may not be aware of this.
- The user is under the impression, that he has his own computers but actually one CPU is shared among many users.



3. Write a note on protection and security.

Ans. If a computer has multiple users and executes the concurrent execution of multiple process and access to data must be regulated.

Protection is the mechanism for controlling the access or users to the resource defined by a computer system.

It can improve reliability by detecting errors it provides a distinguish between authorized and unauthorized usage.

Protection and security requires the system to be able to distinguish among all its users.

Most operating system maintain a list of user name and associated user identification. If the user needs some extra privileges the effective user ID will be provided by the OS.

4. Explain distributed system.

Ans. A distributed system is a collection of physically separate computer system that are networked. These networked computers share the various resources that the system maintains. The desirable properties distinguish systems are.

1. Resource sharing

A user at one side may be able to use resource available at another.

Example: User at site 'A' may access laser printer, which is available only at 'B'.

2. Competition speedup

A distinguish system may distribute. the competition among various sites and those competitions can be done simultaneously.

3. Reliability

If one site fails in an distributed system then the operations of other sites will be affected if same piece of information is available at various sites then failure of one site will not cause the loss of information.

4. Communication

When many sites are connected to by a communication network, the processor at different sites have to exchange information.

Example: File transfer between two system, (email)

5. Write a note client server computing.

Ans.

(i) Client server Computing.

In a client server system the client sends a request to the server and server perform the action and sends back the results to the client.

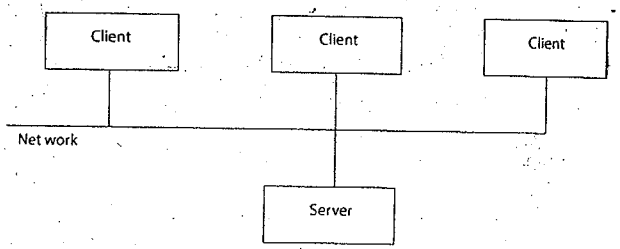


Fig. structure of a client server system

(ii) Peer to Peer network.

Peer is a computer which is a resource provider as well as resource user. In peer to peer network there is no central entity all are peers the resource may be software resources such as files, programs present in a system or a printer attached to the system.

To participate in a peer to peer system a node must first joint the network once a node has joined the network it can act as a service provider or service user.

(iii) Web Based Computing.

The web has become important leading to more access by a wide variety of devices that work on network has faster network connectivity, provided by either improved networking technology, optimized network implementation code.

The implementation of web based computing has given rise to new devices such as "Load balancer" which distributes network connection among a pool of similar server.

6. List the advantages and disadvantages of multi programming system.

Ans. Advantages:

* Increase the CPU utilization and reduce CPU idle time.
* It decreases total read time needed to execute a job as the job are in main memory.

Disadvantages:

* It is fairly sophisticated and more complex as compared to uni-programming system.
* Process can consume more than available memory. In such case system slows down or it may hang some time.

7. Explain the peer-to-peer computing.

Ans peer is a computer which is a resource provider as well as resource user

In peer to peer network there is no central entity all are peers the resource may be software resources such as files, programs presenting a system OR a printer attached to the system.

To participate in a peer to peer system a node must first joint the network once a node has joined the network it can act as a service provider OR service user

10 Marks Questions and Answers

1. What is an OS. Explain the computer system architecture.

Ans. An OS is a collection of software that manages computer hardware resources computer system architecture includes

1. Multi Processor System.

Multi Processor System have more than one processor sharing computer bus, the clock and some memory and peripheral devices. These systems are referred to as tightly coupled system.

There are several reasons for building such systems.

(a) Through Put

- One advantage is that, to increase the through put, more work can be done in shorter period of time. The speed up ratio with 'n' processor is not 'n' but it is less than 'n' even though multiple processor exist if they share code and memory so on.
- It is not possible to achieve 100% utilization.
- Multi processor can also save memory and money compared multiple single system because the processor can share peripheral devices and power suppliers.

(b) Reliability.

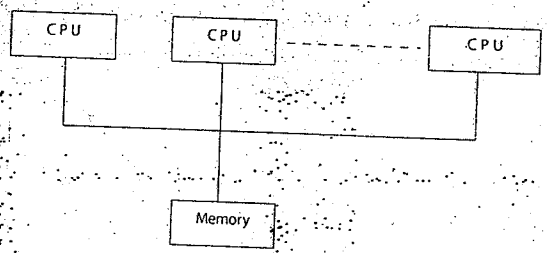
- A failure of one processor will not halt the system.

(c) Fault tolerant system and graceful regradation.

- In a multi processor system if we have 10 process and one fails then each of the remaining process must take the burden of failed process.
- This ability to providing service is proportional to the level of surviving hardware is called graceful degradation.

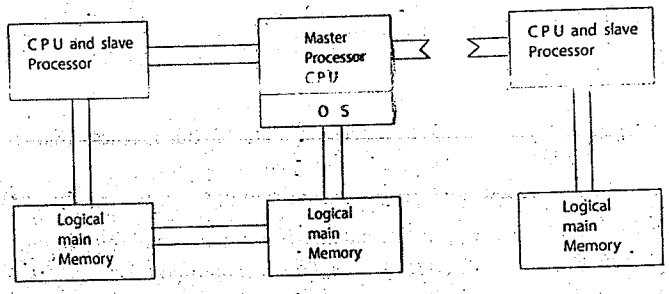
2. Symmetric and Asymmetric Multi Processing System.

- In symmetric multiprocessing system all processor share a common memory and each processor can run identical copy of OS.
- All processor are peers (equality in all aspects) and there is no master slave relationship among multiple processor.



In a symmetric multiprocessing system there exists a single master processor (CPU) which controls and co-ordinates the activities of slave processor.

- All slave processor are assigned a specific task and are executed in parallel.
- Slave processor looks for the instruction from the master processor, so master slave relationship exists in this type.

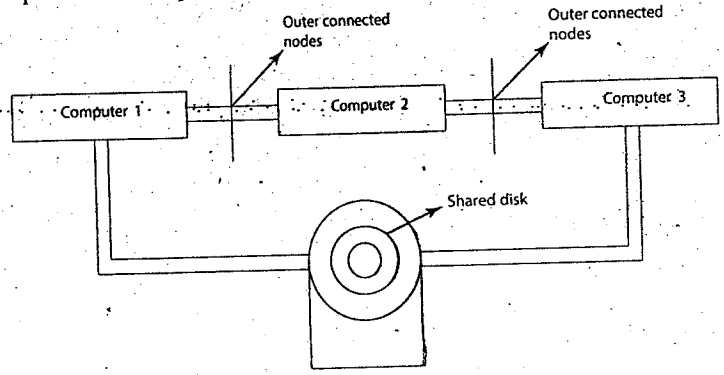


3.Clustered System.

Another type of multiple CPU system is the clustered system. It makes a group of multiple CPU's to accomplish computational work. The generally accepted definition is that clustered computers share, storage and are closely linked through a LAN. Clustering is usually used to provide high availability service. i.e service will continue ever if one or more systems in the cluster fails.

Clustering can be structured asymmetrically or symmetrically.

- In asymmetric clustering one machine is not stand by mode while other is running the applications.
- The not standby host doesn't perform any useful work except for monitoring the active servers but once the active server fails or stop to work for the "Not standby host" machine acts as an active server.
- In symmetric clustering system all systems running applications and monitor each other.
- Clustered technology is so advanced and all machines is a cluster separated by mails.
- These improvements are possible due to storage area network.



2. What are the different operating system operations.

Ans. OS has two separate modes of operation (Dual mode operation)

(i) User mode.

(ii) Kernel mode (Supervisor mode, system mode or privileged mode)

- A Bit called the mode bit added to the hardware of the computer to indicate the current mode i.e Kernel (0) or user (1)
- When the computer system executing on behalf of a user application, the system is in user mode. When a user application, request a service from the system (OS) it must change from user to kernel mode to fulfil the request.
- At the system boot time the hardware starts in kernel mode.
- The OS is then loaded and starts user application in user mode.
- Whenever a trap or interrupt occurs the hardware switches from user mode to kernel mode (mode bit set to 'O')
- The system alway switches to user mode (by setting mode bit to '1') before passing control to a user program.

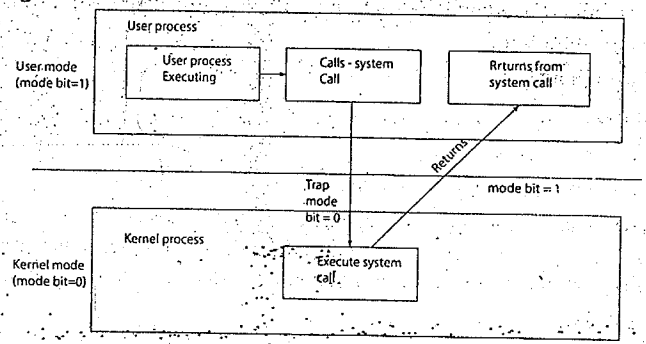


Fig. Transition from user to kernel mode

3. Explain the special - purpose system.

Ans. 1. Real time embedded System

Embedded systems all same everywhere from car engines, robots and microwave ovens.

The operating systems provides limited features.

It just monitors the hardware devices such as automobile engines and robotic arms.

Given now entire house can be computerized, so that a central computer which is either a general purpose computer or a system which can control heating and lighting alarm systems.

Embedded system almost always seen real time operating system.

. Real Time Operating System

A real time system has well defined, fixed time constraints (rules), processing must be done within he defined time otherwise the system will fail.

The real time systems are used in

. Industrial Control System.

. Scientific Experiments.

The control system it should analyse data according to sensor input (eg-temperature control ystem, military system)

. Hard and real time system.

A real time system has well defined fixed time constraints (rules), these are two types of real time ystem.

. Hard real time system.

. Soft real time system.

n Hard real time system a fixed task should be over within predefined time. It is very rigid.

Hard real time system is designed to solve one problem not many so secondary storage is ompletely absent or minimum.

Advanced OS features are.

. Multi programming.

. Multi tasking.

. Virtual reality are not supported by these system.

oft real time systems are priority based depending upon the importance of the task, priority of he task, soft real time system selects and executes the task and completes on time. It supports all dvanced OS features and supports multi programming, virtual memory and soon.

. Multi Media System.

Most OS are designed to handle conventional data such as text files, programs documents and pread sheets. But multi media data consist of audio and video files as well as conventional files.

Multimedia describes a wide range of applications that are in popular use today these includes udio files such as Mp3, DVD movies, video conferencing and news stories.

xample:

movie consists of separate audio tracks. Multimedia application are not only designed for esktop computers, they have been designed for smaller devices such as PDA (Personal digital ssistance) and cellular telephones.

4. Hand held Systems

It includes PDA such as palm Pc's and packet Pc's and cellular telephones which use special purpose embedded OS. The amount of memory in handheld depends upon the device but typically it is somewhere between 512 KB, and 128 MB. So the OS and applications must manage memory efficiently.

The second point is related to the speed of the processor used in the devices, most hand held devices use smaller, slower processors that consumes less power.

That last point to be considered is the usage of small physical space, small keyboards, and touch screen.

Limitations

- Small Physical space but they can be portable.

Merits

- Network connections are possible.
- Other options such as cameras, Mp3 players.

4. Compare process management and memory management.

Ans.

Process Management	Memory Management
1. An OS is responsible for the following tasks with regards to process management.	1. OS is responsible for keeping track of which blocks of memory are currently in use and by which process.
2. Creating and deleting both user and system process.	2. Determining which blocks of code and data to more into and out of memory and when.
3. Suspending and resuming process	3. Allocating and deallocating memory as needed eg: new, malloc.
4. Process synchronization and communication.	4. All data in memory before and after processing
5. Deadlock handling	5. All instruction in memory in order to execute
6. Ensuring that each process receives its necessary resource, without interfering with other process.	

5. Different activities in connection with process management, memory and storage management.

Ans. Operating system is responsible for following activities in connection with management of memory.

1. Allocation and deallocation of memory as and when needed.
2. Keeping track of used and unused memory space.
3. Deciding what process to be loaded into memory in case space becomes available.

Storage Management.

1. Swap space and free space management.
2. Disk scheduling.
3. Allocating space to the data and programs onto the secondary storage device.

Process Management.

1. Creation, deletion of both user and system process.
2. Handling process synchronization.
3. Deadlock handling.

6. Explain clustered system.

Ans. Another type of multiple CPU system is the clustered system. It makes a group of multiple CPU's to accomplish computational work. The generally accepted definition is that clustered computers share, storage and are closely linked through a LAN. Clustering is usually used to provide high availability service. i.e service will continue ever if one or more system in the cluster fails

Clustering can be structured asymmetrically or symmetrically.

- In a symmetric clustering one machine is hot stand by mode while other is running the applications
- The not stand by host doesn't perform any useful work except for monitoring the active servers but once the active server fails or stop to work for the "not standly host" machine acts as an active server.
- In symmetric clustering system all systems running applications and monitor each other.
- Clustered technology is so advanced and all machines is a cluster separated by mails
- These improvements are possible due to storage area network

UNIT 2

PROCESS MANAGEMENT

SYLLABUS

1. Process Concept.
2. Process Scheduling.
3. Operations on Processes.
4. Interprocess Communication.

Process Scheduling

1. Basic Concepts
2. Scheduling Criteria.
3. Scheduling Algorithms.

SYNOPSIS

Objectives:

To introduce the notion of a process, a program in execution, which forms the basis of all computation. To describe the various features of processes including scheduling, creation and termination, and communication.

1. Process Concept

An operating system execute a variety of program are. Batch system-jobs, Time-shared systems-user programs or tasks.

Process is a program in execution; process execution must progress in sequential fashion. A process includes program counter, stack, data section.

1.1 Process State

As a process executes. it changes state are

1. **new:** The process is being created.
2. **running:** Instructions are being executed.
3. **waiting:** The process is waiting for some event to occur.
4. **ready:** The process is waiting to be assigned to a processor.
5. **terminated:** The process has finished execution.

1.2 Process Control Block (PCB)

Information associated with each process are

a. Process state.
b. Program counter.
c. CPU registers.
d. CPU scheduling information.

e. Memory-management information.
f. Accounting information.
g. I/O status information.

2. Process Scheduling

Process scheduler selects among available processes for next execution on CPU. Maintain scheduling queues of processes.

i. **Job queue** - set of all processes in the system.
ii. **Ready queue** - set of all processes residing in main memory, ready and waiting to execute.
iii. **Device queues** - set of processes waiting for an I/O device processes migrate among the various queues.

2.1 Schedulers

1. Long-term scheduler (or job scheduler) - selects which processes should be brought into the ready queue.

2. Short-term scheduler (or CPU scheduler) - selects which process should be executed next and allocates CPU.

2.3 Context Switch

When CPU switches to another process the system must save the state of the old process and load the saved state for the new process via a **context switch Context** of a process represented in the PCB.

Process Creation

Parent process create **children** processes. Which, in turn create other processes. forming a tree of processes. Generally process identified and managed via a **process identifier (pid)**

Process Termination

Process executes last statement and asks the operating system to delete it **(exit)**. Output data from child to parent (via **wait**). Process resources are deallocated by operating system.

4. Interprocess Communication

Cooperating processes need interprocess communication (IPC). Two models of IPC.
1. Shared memory.
2. Message passing.

5. Basic Concepts

Maximum CPU utilization obtained with multiprogramming. CPU-I/O Burst Cycle-Process execution consists of a cycle of CPU execution and I/O wait. CPU burst distribution.

4.1 Dispatcher

Dispatcher module gives control of the CPU to the process selected by the short-term scheduler; this involves: switching context, switching to user mode jumping to the proper location in the user program to restart that program.

5. Scheduling Criteria

a. **CPU utilization** - keep the CPU as busy as possible
b. **Throughput** - processes that complete their execution per time unit.
c. **Turnaround time** - amount of time to execute a particular process.

d. **Waiting time** - amount of time a process has been waiting in the ready queue.
e. **Response time** - amount of time it takes from when a request was submitted until the first response is produced, not output (for time-sharing environment).

6. Multilevel Queue

Ready queue is partitioned into separate queues, eg: foreground (interactive), background (batch).

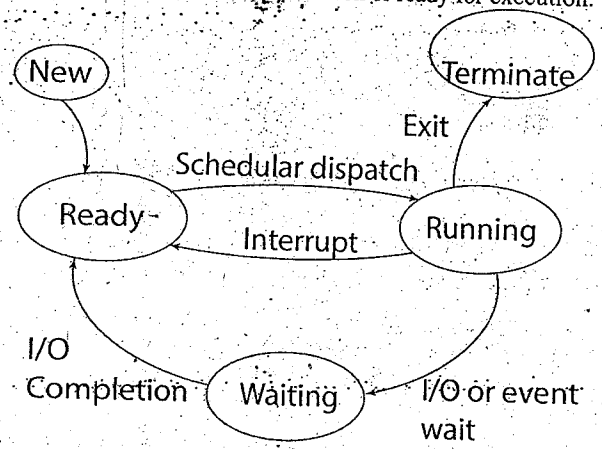
6.1. Multilevel Feedback Queue

Multilevel - feedback-queue scheduler defined by the following parameters:
1. Number of queues.
2. Scheduling algorithms for each queue.
3. Method used to determine when to upgrade a process.
4. Method used to determine when to demote a process.

5 Marks Questions and Answers

1. What is a process. Draw and explain the state transition diagram of a process.

Ans. A process is a program, which is under execution or ready for execution.



State transition diagram

When a process executes, during execution it changes state. The current activity of a process defines the state (new, ready, waiting and so on) of process. Each process may be in one of the following states.
1. New- The process is being created.
2. Running - The instructions are being executed.
3. Waiting - **The** process is waiting for some event. to occur (for ex- I/O operation completion).
4. Ready - The process is waiting for the processor for execution.
5. Terminated - The process has finished execution.

2. Explain the contents of PCB with neat diagram.

Ans. Each process is represented in OS by a process control block (PCB). It is shown below

Process State
Process Number
Program Counter
Register
Memory Limits
List of open files

A PCB contains many pieces of information associated with a specific process are,

i. Process state - A state may be new, ready running, waiting and so on.

ii. Program Counter - It holds the address of the next instruction to be executed.

iii. CPU registers - The registers vary in numbers depending on the computer architecture.

iv Memory management information - This information may include a value of a base and limit registers, page tables and soon.

v. Accounting information - This information includes the account of CPU time use, process numbers and soon.

vi. I/O status information - This information includes the list of I/O devices allocate to the process and list of open files etc.

3. Compare Long-term, Short-term, Medium-term scheduler.

Ans.

Long-term scheduler	Short-term scheduler	Medium-term scheduler
1. Also known as "job scheduler".	1. Also known as CPU scheduler.	1. Is a part of swapping.
2. Selects a set of jobs from hard disk and loads it to main memory for execution,	2. Selects a process from memory and assign it to CPU.	2. It removes the process from the memory.
3. Less frequently executed	Frequency of execution is more.	3. It reduces the degree of multi programming.
4. Selects a good mix of CPU and I/O bound process	4. CPU bound process gets more CPU time and I/O bound process gets less CPU time.	4. MTS is incharge of handling the swapped out process.
		5.Running process may become suspended if it makes an I/O request.

4. Explain the different scheduling criteria.

Ans. Many criteria are used for comparing scheduling algorithms.

1. CPU utilization: We want to keep the CPU as busy as possible. CPU utilization can range from 40% (for light loaded system) to 90% (for heavily used system).

2. Through put: The number of processes that are completed per time unit, called through out for long process this rate may be one process per hour. For short transactions it may be 10 processes per hour.

3. Turn around time: The interval from the time of submission to the time of completion of a process is known as turn around time. Turn around time is the "sum of the periods spent waiting, to get into memory, waiting in the ready queue, executing on the CPU and doing I/O.

4. Waiting time: The waiting time is the sum of the periods spent waiting in the ready queue.

5. Response time: The time from the submission of a request until the first response is produced. This measure is called response time.

10 Marks Questions and Answers

1. Explain the different operations on process.

Ans. There are two operation that can be performed on process.

* Process Creation.
* Process Termination.

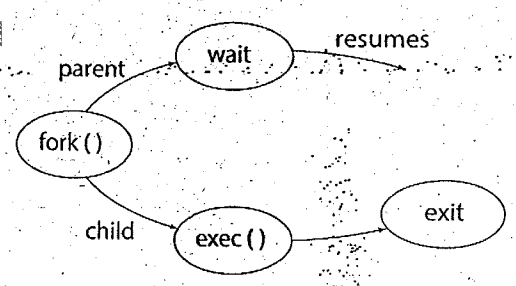
- **Process Creation.**

A process may create several new process through a "create process" system call during execution. The new process are called children of that process may in-turn create other process forming a tree of process.

When a process creates new process two possibilities exist in terms of execution.

i. The parent continues to execute concurrently with its children.

ii. The parent waits until some or all of its have terminated.



Here fork is a system call in UNIX, which creates child process.

- Process termination.

Process executes last statement and as the OS to delete it (exit).

* Output data from child to parent.

* Process resources are deallocated by OS.

Parent may terminate execution of children process abort.

* Child has exceeded allocated resources.
* Task assigned to child is now longer required.

If parent is exiting

* Some OS do not allow child to continue if its parents terminate.
* All children terminated - cascading termination.

3. Consider the following set of process with the length of the CPU burst time given in milliseconds.

Process	Burst Time	Priority
P1	10	3
P2	13	1
P3	3	3
P4	8	4

The process are assumed to have arrived in the order P1, P2, P3, and P4 all at time 0.

(a) Draw the Gantl chart for FCFS and PRIORITY scheduling algorithms.

(b) What is the waiting time and turn around tiem of each process for FCFS and PRIORITY scheduling algorithms.

(c) Calculate the average waiting time and average turn around time for FCFS and PRIORITY scheduling algorithms.

FCFS Gantt Chart

P1	P2	P3	P4

0 10 23 26 34

Process	Waiting Time	Turn around Time
P1	0	10
P2	10	23
P3	23	26
P4	26	34

Priority scheduling Gantt chart

P2	P1	P3	P4

0 13 23 26 34

Process	Waiting Time	Turn around Time
P1	13	23
P2	0	13
P3	23	26
P4	26	34

Average waiting time $= \frac{0+10+23+26}{4} = 14.7$

Average waiting time $= \frac{13+0+23+26}{4} = 15.5$

Average turn around time $= \frac{10+23+26+34}{4} = 23.25$

Average waiting time $= \frac{23+13+26+26+34}{4} = 24$

2. What is inter - Process communication (IPC). Explain the two model of IPC with neat diagram.

Ans. Co-operating processes require an inter-process communication (IPC) mechanism that will allow them to exchange data and information. There are two fundamental model of inter-process communication.

* Shared memory model.
* Message passing model.

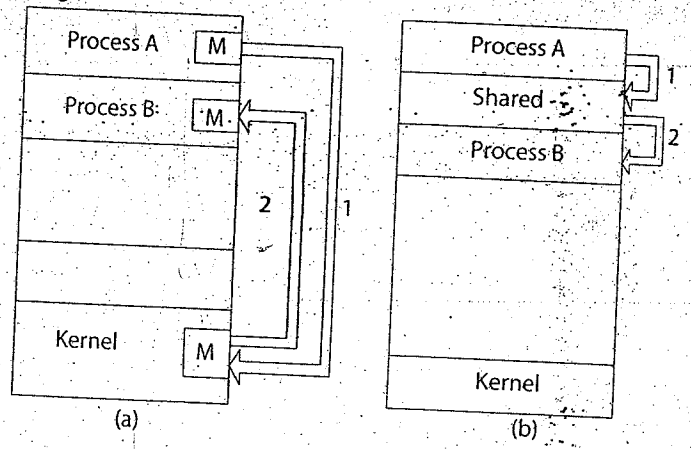


Fig. Communication models

(a). Message Passing.
(b). Shared Memory.

(i) Shared Memory Model.

In a shared memory model, a region of memory that is shared by co-operating processes is created. Processes can then exchange information by reading and writing data to the shared region.

A shared-memory region resides in the address space of the process creating the shared memory segment. Other processes that wish to communicate using this shared-memory segment must attach it to their address space.

Normally, operating system presents one process from accessing another process's memory. Shared memory requires that two or more processes agree to remove this restriction. They can

ıen exchange information by reading and writing data in the shared areas. The form of the data nd the location are determined by these processes and not under the operating system's control. 'he processes are responsible for ensuring that they are not writing to the same location imultaneously.

ii) Message-Passing System

n message-passing system, communication takes place by means of messages exchanged between he co-operating processes.

Aessage-passing provides a mechanism to allow process to communicate and to synchronize heir actions without sharing the same address space and is particularly useful in a distributed nvironment, where the communicating processes may reside on different computers connected)y a network.

A message-passing facility provides at least two operations:

i). Send (message)

ii). Receive (message)

Message sent by a process can be of either fixed size or variable size.

UNIT 3

SYNCHRONIZATION & DEADLOCKS

SYLLABUS

1. Background.
2. The Critical-Section Problem.
3. Semaphores.
4. System Model.
5. Deadlock Characterization.
6. Methods for Handling Deadlocks.
7. Deadlock Prevention.
8. Deadlock Avoidance.
9. Deadlock Detection.
10. Recovery from Deadlock.

SYNOPSIS

1. Background

Concurrent access to shared data may result in data inconsistency. Maintaining data consistency requires mechanisms to ensure the orderly execution of cooperating processes. Suppose that we wanted to provide a solution to the consumer-producer problem that fills all the buffers.

2. Critical Section Problem

Each process has **critical section** segment of code. Process may be changing common variables, updating table, writing file, etc. When one process in critical section, no other may be in its critical section.

3. Semaphore

Synchronization tool that does not require busy waiting. Semaphore S-integer variable. Two standard operations modify S: wait ( ) and signal ( ). Less complicated.

4. System Model

Each process utilizes a resource as request, use release.

5. Deadlock Characterization

1. Mutual exclusion: only one process at a time can use a resource.

2. Hold and wait: a process holding at least one resource is waiting to acquire additional resources held by other processes.

3. No preemption

A resource can be released only voluntarily by the process holding it, after that process has completed its task.

4. Circular wait: there exists a set $\{P_0, P_1, \ldots P_n\}$ of waiting processes such that P_0 is waiting for resource that is held by P_1, P_1 is waiting for a resource that is held by.

6. Methods for Handling Deadlocks

Ensure that the system will never enter a deadlock state. Allow the system to enter a deadlock state and then recover. Ignore the problem and pretend that deadlocks never occur in the system; used by most operating systems, including UNIX.

7. Deadlock Prevention

1. Mutual Exclusion - Not required for shareable resources; must hold for non-shareable resources.

2. Hold and Wait - Must guarantee that whenever a process requests a resource, it does not hold any other resources.

3. No Preemption - If a process that is holding some resource requests another resource that cannot be immediately allocated to it, then all resources currently being held are released.

4. Circular Wait - Impose a total ordering of all resource types, and require that each process requests resources in an increasing order of enumeration.

8. Deadlock Avoidance

The deadlock - avoidance algorithm dynamically examines the resource-allocation state to ensure that there can never be a circular-wait condition. Resource-allocation state is defined by the number of available and allocated resources, and the maximum demands of the processes.

9. Deadlock Detection

1. Allow system to enter deadlock state.
2. Detection algorithm.
3. Recovery scheme.

10. Recovery from Deadlock

Abort all deadlocked processes. Abort one process at a time until the deadlock cycle is eliminated.

5 Marks Questions and Answers

1. Write a note on semaphores.

Ans. The hardware based solutions to the critical section problem are complicated to use for application programmers. To overcome this difficulty we can use a synchronization tool called a semaphore.

A semaphore 'S' contains an integer variable, is accessed only through two stand and atomic operations Wait( ) and Signal ( ).

The definition of Wait( ) is as follows

```
Wait (S)
{
        While       S <= 0
                    ;       Loop.
              S--;
}.
```

The definition signal ( ) is as follows

```
Signal (S) {
        S + +;
        }.
```

For example, consider two concurrently running processes, P1 with statement S1 and P2 with statement S2.

Suppose P1 and P2 are sharing common variable "synch" then the following statements can be seen,

S1;

Signal (synch) in process P1 and the statements.

Wait (synch) in process P2, S2;

P2 will execute S2 only after P1 has invoked signal (synch), which is after statement S1 has been executed.

2. Explain the 3 requirement for the solution to critical-section problem.

Ans. 1. Mutual exclusion: If process Pi is executing in its critical section, then no other processes can be executing in their critical sections.

2. Progress: If no process is executing in its critical sections, then only those processes that are not executing in their remainder sections can participate in deciding which will enter its critical section next.

3. Bounded waiting: There exists a bounded limit on number of times that other processes are allowed to enter their critical sections after a process has made a request to enter its critical section.

3. Write a note on critical section problem and the solution to solve it.

Ans. Each process has a segment of code called a critical section, in which the process may be changing common variables, updating a table, writing a file and so on.

The important feature of the system is that, when one process is executing in its critical section, no other process is allowed to execute in its critical section.

The main purpose is to design a protocol which does not allow two processes to execute in their critical section.

The structure of the critical section is

```
do {
    entry section
        Critical Section
    exit section
        Remainder Section.
    } While (TRUE);
```

The section of code implementing this request is the entry section. The critical section may be followed by an exit section, the remaining was the remainder section.

A solution to the critical section problem must satisfy the following 3 requirements.

Ref: question number (2) for solution to solve it.

4. What is deadlock. what are the necessary condition for dead lock.

Ans.-Process request the resource during the course of the execution, if the resources are not available at that time the process will enter in to waiting state. Sometimes a waiting process never able to change its state, because the requested resources are held by other process, this situation is called as deadlocks.

Necessary conditions are.

1. Mutual exclusion: Only one process at a time can use the resources if another process request the resources, the requesting process must be delayed until the resource has been released.

2. Hold and wait: A process must be holding at least only one resource, at waiting for additional resources that are currently being held by other processes.

3. No pre-emption: Resource can be released only voluntarily but the process holding it, after that process has completed its task.

4. Circular wait: A set of [P0,P1,P2 --------Pn] of waiting process must exist, such that P0 is waiting for a resource held by P1, P1 is waiting for a resource held by P2 and soon. Pn-1 is waiting for a resource held by P0.

All these 4 conditions must hold for a deadlock to occur.

5. How deadlock can be prevented. Explain.

Ans. There are four conditions to prevent the deadlock.

1. Mutual exclusion: The mutual exclusion condition must holds good for non shareable resource, i.e system should be built without any non shareable resources. So that the deadlock will not occurs but a system cannot be built without any non shareable resource. Hence mutual exclusion cannot be used to present deadlock.

2. Hold and wait: To ensure that hold and wait condition never occur in the system. we must guarantee that when ever a process request a resource. Then it should not hold any other resource.

The major problem with this approach to present deadlock is

1. The resource utilization will be poor, since resources may not be altered and unused for longer time.

2. There is a possibility of resource starvation as a process may need several resource simultaneously.

3. No preemption: The third necessary condition for deadlock is, there must be no preemption of resource that have already been allocated.

To ensure this condition, does not holds good to prevent the deadlock, the protocols must be used

If a process holding some resource and requesting for other resource that cannot be immediatly granted then all the resources allocated to the process must be preempted immediately.

this suituation the process must be reallocated with the older resources that were eempted and the new resources, that the process was waiting in order to continue the ecution this leads to longer turn around time, which in turn reduces the efficiency of e system.

Circular wait: The fourth necessary condition for deadlock is circular wait condition. e way to ensure this condition never holds good is to impose the total ordering of all ource type and each process must requst for the resource type in an increasing order.

How deadlock can be avoided using resource allocation graph. Explain.

s. In addition to request and allocation edge, we introduce a new type of edge called im edge. A claim edge Pi →Rj indicates that Pi may request resource Rj at some time he future. This edge resembles a request edge in direction but it is represented by,

dge, Whem a process Pi request resource Rj the claim edge Pi→Rj in converted to uest edge. 111[ly] when the resources Rj is released by Pi, the assignment edge Rj→Pi onverted to claim edge. Pi →Rj i.e, before a process starts executing, all its claim edge st already appear in the resource allocation graph.

Suppose that process Pi requests resources Rj, the request can be granted only if verting the request edge.

Pi → Rj to an assignment edge Rj →Pi does not result in the formation of a cycle in resource allocation graph. If no cycle exists there the allocation of the resources will e the system in safe state.

Explain how to recover from deadlock.

. There are two options for breaking the deadlock. One is to abort one or more process reak the circular wait. the second option is to preempt some resources from one or e of the deadlocked processes.

ess termination:

iminate deadlock by aborting a process uses one of the two methods.

bort all deadlocked processes.

Abort one process at a time until deadlock cycle eliminated.

urces preemption:

preempt some resources from process and give these resources to other processes a deadlock cycle broken. There are three issues to be considered under this

lecting a victim process

ll back

rvation.

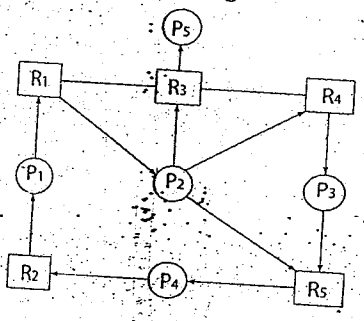
10 Marks Questions and Answers

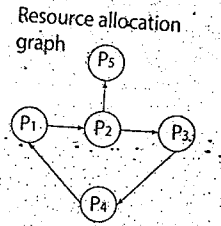
1. How deadlock can be detected. Explain.

Ans. i. Single instance of each resource type.

If all resources have only a single instance then a deadlock detection algorithm uses a variant of resource allocation graph called"wait-for" graph is defined.

This graph is obtained from resource allocation graph by removing the resource nodes and collapsing the appropriate edge.





A deadlock exist in the system if and only if the wait for graph contains a cycle.

To detect dead locks, the system needs to maintain wait for graph and periodically invoke an algorithm, that searches for cycles in graph.

The algorithm to detect a cycle in a graph requires an order of n^2 operation (n-nor of vertices in a graph).

ii. Several instance of resource allocation.

Wait for graph scheme is not applicable to resource allocation system with multiple instances of each resource type.

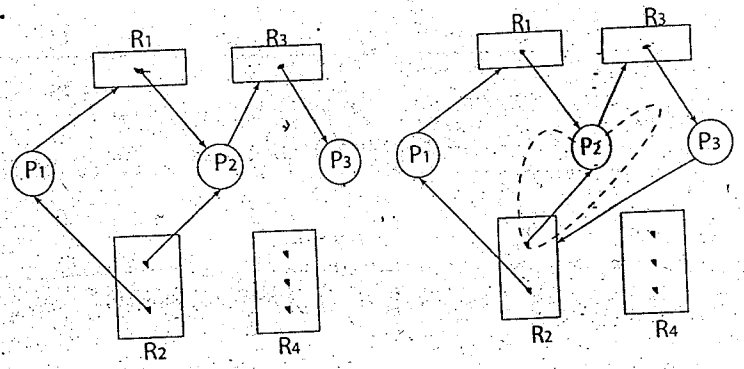
Then a dead lock detection algorithm is used, which is similar to Banker's algorithm.

i. Available - A vector of length 'm' indicates the number of available resources of each type.

ii. Allocation - An nxm matrix defines the number of resources of each type currently allocated to each process.

iii. Request - An nxm matrix indicates the current request of each process if request [i][j] = k, then Pi is requesting 'k' more instance of resource Rj.

3. Explain resource-allocation graph with dead lock and without deadlock with examples.



(a) Resource allocation graph:

Deadlocks can be described more precisely in terms of resource allocation graph. This graph consist of a set of vertices and a set of edges. The set of vertices is divided in to two types.

1. Set of active processes in the system P=P0, P1, P2 upto Pn
2. A set of resources available in the system

R=R0, R1, R2--------Rn.

A set of edges are divided into two types.

1. Request edge→A directed edge from process Pi from resource type Rj is denoted by Pi→Rj, which indicates that process Pi is requesting an object of resource type Rj.

2. Allocation edge:-A directed edge from resource type Rj to process Pi is denoted by Rj to Pi, which indicates that an object of resource type Rj is allocated to process Pi.

In the graph each process Pi is represented as a circle and each resource type Rj is represented as a rectangle. The number of objects in each resource types are represented as dots with in the rectangle.

For example consider the following situation.

1. P=P1, P2, P3
2. R=R1, R2, R3, R4
3. E=R2→P1, R2→P2, P1→R1, R1→P2, P2→R3, R3→P3.

4. There exists one object of resource type R1 Two instance of resource type are R2, one instance of resource type R3, three instance of resource type R4.

2. Explain banker's algorithm.

Ans. The resource allocation graph algorithm is not suitable to a system with multiple objects in each resource type. Hance banker's algorithm is used.

In this algorithm, when a new process enters a system, it must declare the maximum number of objects of each resource type, that it may need. This number should not exceeds the total number of resources in the system. When a user request a set of resource, the system must determine whether the allocation of this resources will leave the system in safe state. If yes the resources are allocated otherwise the process must wait until some other processors release enough resources.

To implement banker's algorithm following data structures are needed.

1. Available-It is a vector of length m, which indicates the number of available resources of each type.
2. max-It is a nxm that define the maximum demand of each process.
3. Allocation- It is a mxn matrix that defines the number of resources of each type are currently allocated for each process.
4. Need- It is a nxm matrix that indicates that remaining resource need of each process.

There are two algorithm in banker's algorithm.

i. Safety algorithm

ii. Resource request algorithm.

i. Safety algorithm:

This algorithm is used to findout whether the system is in safe state or not.

1. Let work and finish be vectors m and n respectively. Initialize

work=Available.

finish [i]= false for i=0, 1------n-1,

2. Find an i such that both.

finish [i]==false.

Need i ≤ work.

If no such i exists go to step 4.

3. Work= Work+Allocation

Finish [i]=true.

go to step 2.

4. If finish [i] = = true for all i, then the system is in safe state.

ii. Resource request algorithm.

1. If request i ≤ need i, go to step 2 other wise raise an error condition. since the process has exceeded its maximum claim.

2. If request i≤ Available, go to step 3 other wise Pi must wait, since the process resource are not available.

3. Hare the system pretend to have allocated the requested resource to process Pi by modifying the state as follows.

Available= Available-Request i

Allocation i = Allocation i + Request i

Need i = Need i-Request i.

This algorithm is used to determine whether the request can be safely granted or not.

UNIT 4

MEMORY MANAGEMENT STRATEGIES

SYLLABUS

1. Background.
2. Swapping.
3. Contiguous Allocation.
4. Paging.
5. Structure of page table.
6. Segmentation.

SYNOPSIS

1. Background

Program must be brought into memory and placed within a process for it to be run. Input queue-collection of processes on the disk that are waiting to be brought into memory to run the program.

1.1. Logical vs. Physical Address Space.

- Logical address - generated by the CPU; also referred to as virtual address.
- Physical address - address seen by the memory unit.

2. Swapping

A process can be swapped temporarily out of memory to a backing store, and then brought back into memory for continued execution. Backing store- fast disk large enough to accommodate copies of all memory images for all users; must provide direct access to these memory images.

3. Contiguous Allocation

Main memory usually into two partitions:

* Resident operating system, usually held in low memory with interrupt vector.
* User processes then held in high memory.

4. Fragmentation

1. External Fragmentation - total memory space exists to satisfy a request, but it is not contiguous.

2. Internal Fragmentation - allocated memory may be slightly larger than requested memory; this size difference is memory internal to a partition, but not being used.

5. Paging

Divide physical memory into fixed-sized blocks called frames Divide logical memory into blocks of same size called pages.

6. Page Table Structure

1. Hierarchical Paging: Break up the logical address space into multiple page tables.
2. Hashed Page Tables: The virtual page number is hashed into a page table. This page table contains a chain of elements hashing to the same location.
3. Inverted Page Tables: One entry for each real page of memory. Entry consists pf the virtual address of the page stored in that real memory location, with information about the process that owns that page.

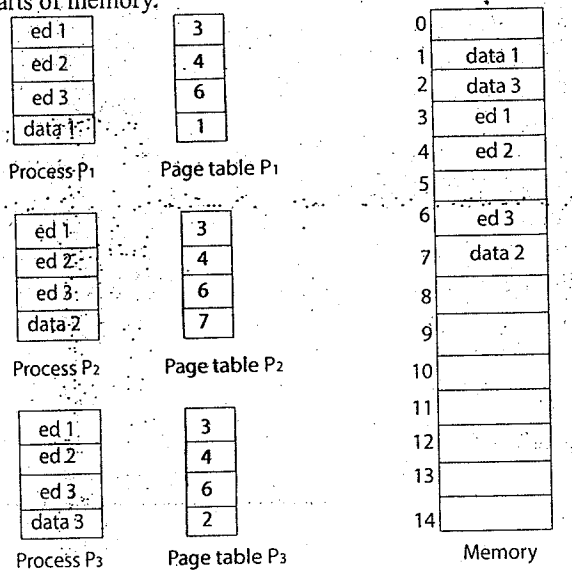
6. Segmentation

Memory -management scheme that supports user view of memory. A program is a collection of segments.

5 Marks Questions and Answers

1. Write a note on shared pages. **5 Marks**

Ans. Paging allows sharing of common code, consider a 3 page editor which can be shared among three pages. Two or more process execute the same code at the same time each process has it was own data page. As shown in the figure process P1, process P2 and process P3 share the common code editor. But data section for all the three processes occupy different parts of memory.



2. Explain First-fit, Best-fit and Wort-fit strategies for memory allocation.

Ans. Following are the most commonly used strategy to select a free hole from the set of available holes.

1. First fit: Allocate the first hole that is big enough. Searching can start either at the beginning of the set of holes or where the previous first fit search ended. We can stop searching as soon as we find a hole that is large enough.

2. Best fit:-Allocate the smallest hole that is big enough. We must search the entire list, unless the list is kept ordered by size. The strategy produces smallest leftover hole.

3. Worst fit:-Allocate the largest hole. Again we must search the entire list, unless it is sorted by size. This strategy produces the largest leftover hole which may be more useful than the smaller leftover hole from a best fit approach.

3. What is fragmentation. Differentiate between internal and external fragmentation.

Ans. Fragmentation is a phenomenon in which storage space is used inefficiently reducing storage capacity and in most cases performance. The term is also used denote the wasted space itself.

The algorithm such as first fit, best fit and worst fit are usually suffer from external fragmentation. As process loaded and removed from main memory, the free memory space is broken in to little pieces, the main memory is fragmented to a large number of small holes.

External fragmentation exist when the total available memory space satisfies the request but it is not continuous all most between two successive process in memory some space is left which cannot be allocated to any other process.

Another problem is internal fragmentation consider the hole of 1000 bytes. Suppose the next process may require. 997 bytes if this block is allocated to the process which require 997 bytes. Three bytes are left cannot be, used in future. It is very difficult to keep track this hole because maintaining the history of the small hole may introduce an additional cost. Thus in most of the cases internal fragmentation is permitted.

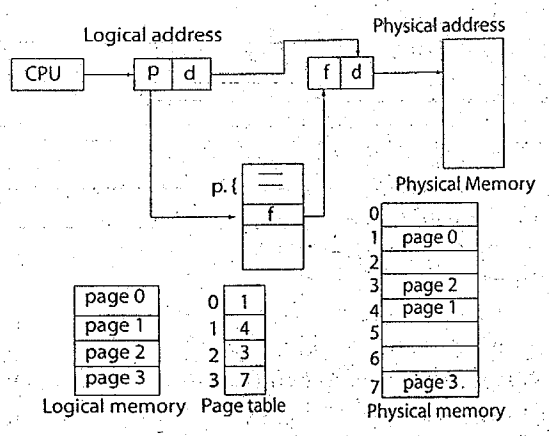
4. Explain basic concept of memory allocation using paging.

Ans. In paging the physical memory is broken into fixed size block called "frames" logical memory is also broken into blocks of same size called "Pages".

The backing store is divided into fixed size blocks that are of the same size as the memory frames. Every address generated by CPU is divided into 2 parts.

i. Pagenumber.
ii. Page offset.

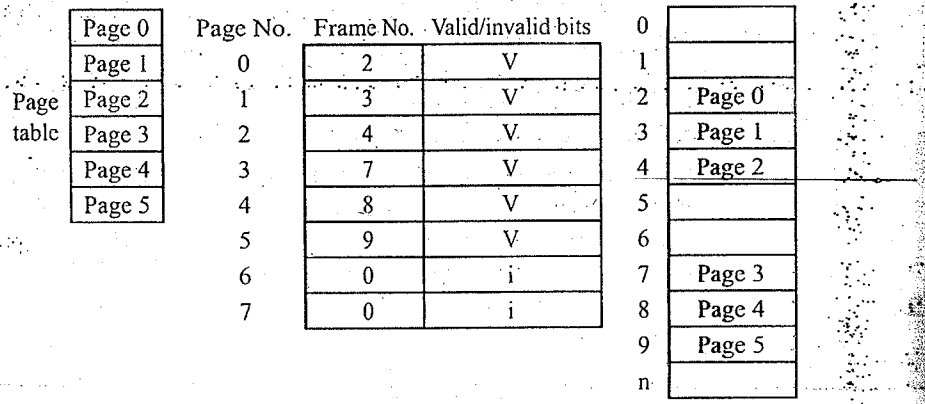
The page number is used as an index to the page table. The table contains the base address of physical memory, this base address is combined with the page offset to define the physical memory. i.e sent to the memory unit. The model is shown in the figure



5. Explain how memory is protected in paging with a neat diagram.

Ans. Memory protection in a paged environments can be done by protection bits, that are associated with each frame. Normally these bits are kept in page table. One bit can define a page to be read and write or read only. Every reference to memory goes through the page table to find the correct frame number. At the same time the physical address can be computed. The protection bits can be checked to verify that no write operation must be made to a read only.

An attempt to write to a read only page causes hardware trap to the OS. one more bit is generally attached to each entry in the page table. This can be done with the help of valid/ invalid bits as shown below.



6. Differentiate between Logical v/s Physical address space.

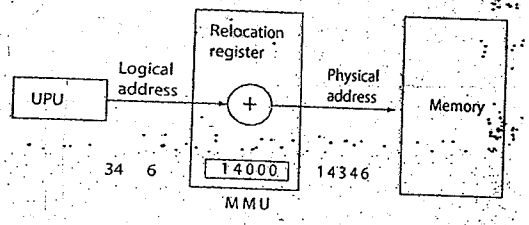
Ans. An address generated by the CPU is commonly know a Logical address.

Example: Code, data in a program (source program).

The set of all Logical address generated by a memory is called Logical address space. and set of all physical address generated by a memory is called physical address space.

The following Fig shows how the Logical address are converted to physical address space. This is done by memory management hardware(MMu).

In the simplest case MMu can be implement with the help of relocation register the value in the relocation register (base register) is added to every address generated by the user process to get the physical address.



7. Explain swapping technique with a neat diagram.

Ans. In case of multi programming which uses round robin scheduling algorithm when quantum (time slice) expires, the memory manager starts to swapout the process that has just finished and swapin another process to memory.

Sometimes swapping is also depending on the priority of the process. So, higher priority process is swapped in and lower priority process is swapped out.

After the completion of time slice of highest priority process low priority process continues the execution.

The other characteristics of swapping are,

a. The process which the CPU wants to execute is not in main memory, then the required. process is swapped in and a process is swapped out to the hard disk.

b. If the amount of memory occupied by a process is more than time required to swap the process is also more.

c. Never swap a process which wants to do any I/O operation. so don't swap a process until it becomes idle.

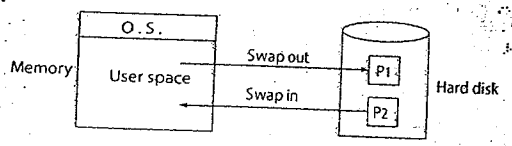


Fig. Swapping of two processes using a disk as a backing store.

8. Explain Address Binding, with a neat diagram. **5 Marks**

Ans.

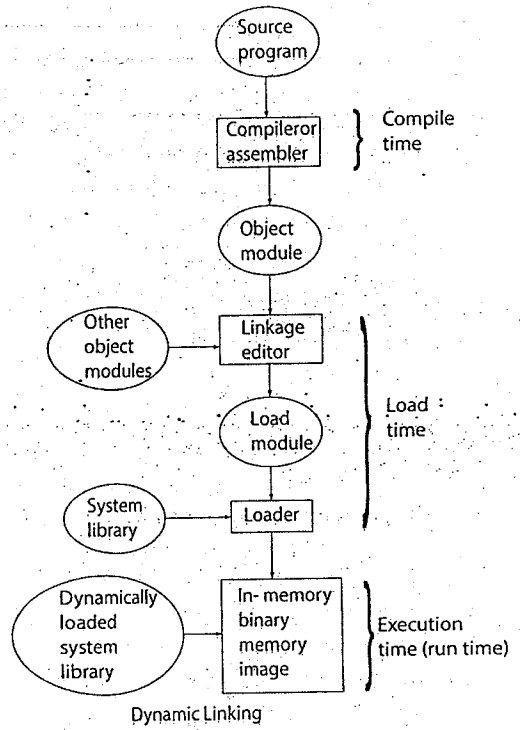


Fig. Multi step processing of a user program

Above figure shows set of actions carried out during the execution of program .

* It is called as life cycle of a program.
* Most of the OS allow user program to reside on any part of physical memory.
* To execute a source program, it must be converted to object program and then binary program.
* During this the program resides on various parts of memory having different address each time.

* Address may be represented in different ways during the life cycle of program.
* Address in the source program are usually represented in the symbolic form.
* A compiler will bind these symbolic address to relocatable address. The linker or loader will in turn bind these relocatable address to absolute address.
* The binding of instruction and data to memory address can be done at any one of the time or all time during its life cycle.
* Compile time-It is the time taken to convert source program to object program.
* Load time or link time- It is the time taken to load binary program or load module on to main memory for execution.
* Execution time- It is the time taken to execute a complete program during this time binary program is moved from one. memory section to another which also delays binding operation.

10 Marks Questions and Answers

1. Explain the contiguous memory allocation.

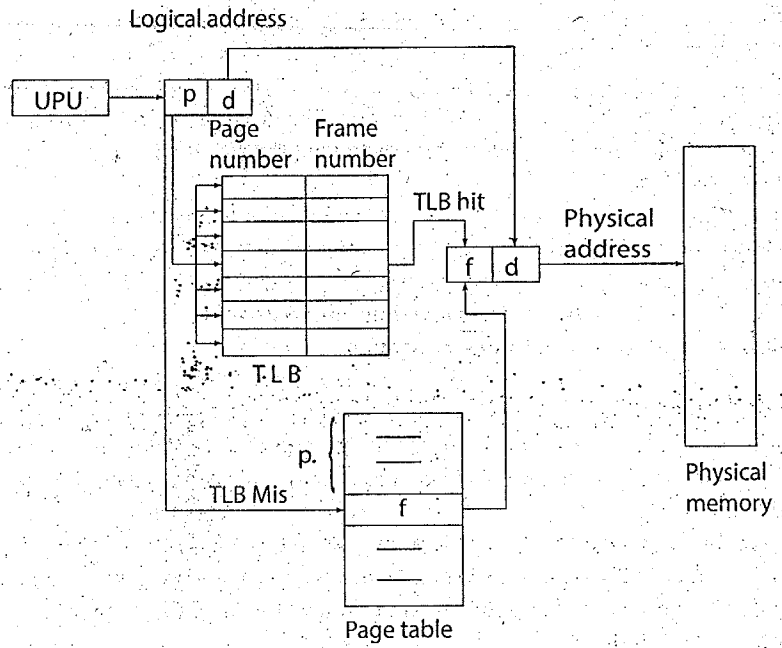
Ans. Contiguous allocation is the simplest schemes of allocating main memory. the OS keeps a table indicating which parts of memory are available and which are occupied. Initially all memory is available for user process and is considered as large block of available memory know as a "hole". when a process arrives and needs memory a hole is searched. If the hole is found then allocate only as much as memory and unused space may be kept for future process. In general at any given time, there is a set of holes of various sizes scattered through out memory, when the process arrives and needs memory, the system(OS) searches for the hole. when the hole is too large then one partis allocated and another part as a hole.

When multiple partitions scheme (for ex: 500k, 300k, 800k partitions) is used, it is important to note, how to handle the processes and their memory allocation. Allocating the best hole for a processes is a challenge to OS.

2. Explain hardware implementation of page table with a neat diagram.

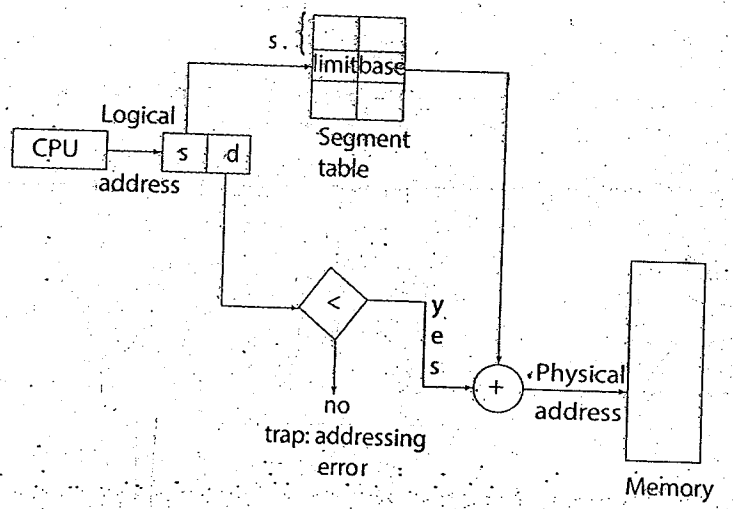
Ans. The figure given below explains the hardware implementation of paging using "Translation look a side buffer (TLB)", the associated registers contain only a few of the page table entries. when a logical address is generated by the CPU its page number is presented to a set of associated registers that compare page numbers and their corresponding frame numbers. if the page number is found in the associative registers, its frame number is immediately available and is used to access memory.

If the page number is not in the associative registers, a reference to the page table (TLB Miss) must be made. when the frame number is obtained, that can be used to access memory. If the TLB is already full of entries it must be flushed.

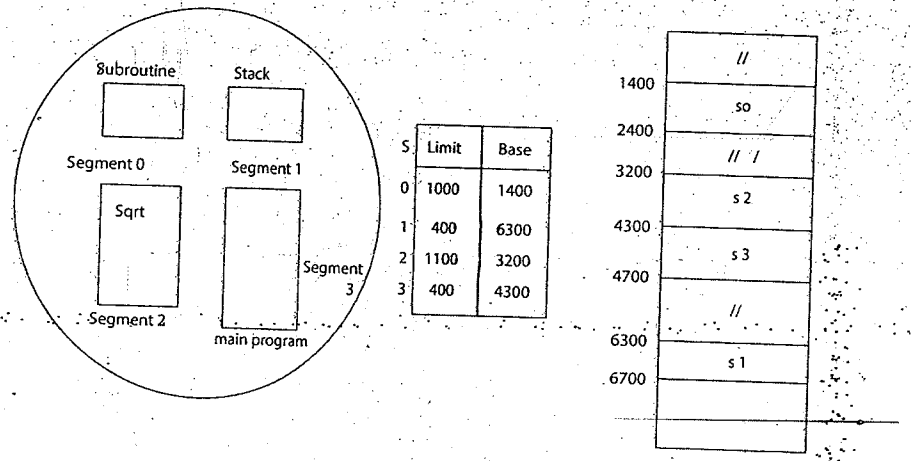


4. What is segmentation. Draw and explain its hardware support.

Ans. Segmentation is a memory management scheme that supports user view of memory. logical address space is a collection of segments. each segment has a name and a Length. Each entry of the segment table contains the physical address where the segment resides in memory, where as the segment limit specifies the Length of the segment. A Logical address consists of two parts, a segment number 's' and an off set in to that segment 'd'. The segment nor 's' is used as an index to the segment table. the offset 'd' of the Logical address must be between 0(zero) and segment limit. Then it is added to the segment base to produce the address in physical memory.



Ex:-



3. List and explain different structure of the page table with a neat diagram.

Ans. Page table structure

1. Hierarchical paging
2. Hashed page tables

Inverted page tables.

Hierarchical page tables.

reak up the logical address space in to multiple page tables. A simple technique is a two-
vel page table.

wo - level paging example.

logical address (on 32 bit machine with 4k page size) is divided into:

A page number consisting of 20 bits.

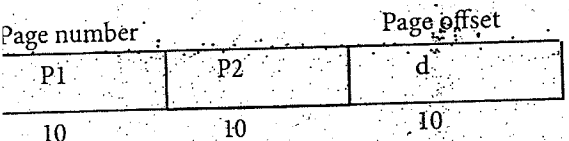
A page offset consisting of 12 bits.

ince the page table is paged, the page number is further divided into:

. 10 bit page number.

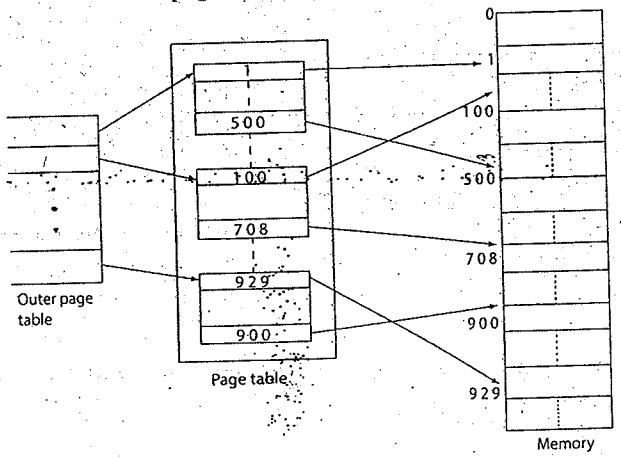
. 10 bit page offset.

ogical address is as follows.



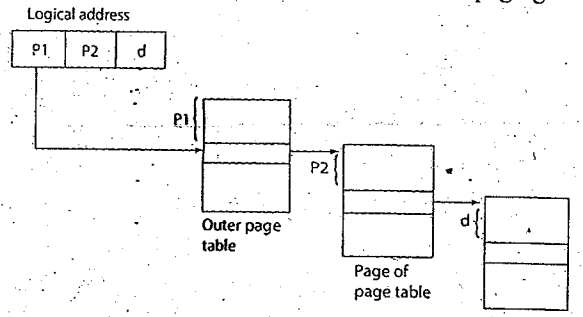
Where Pi is an index into the outer page table, and P2 is the displacement with in the page of the outer page table.

Two level page table scheme.



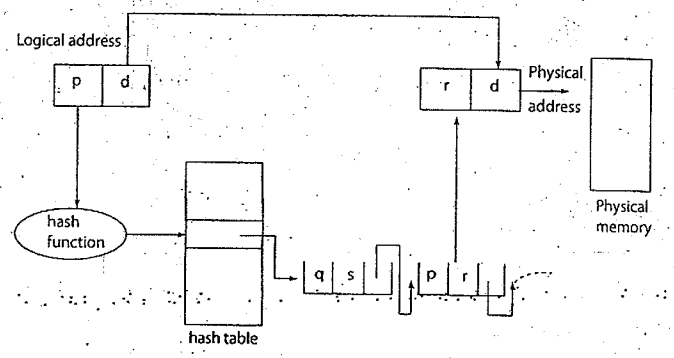
Address-Translation scheme.

Address- Translation scheme for a two-level 32-bit paging architecture.



2. Hashed page table.

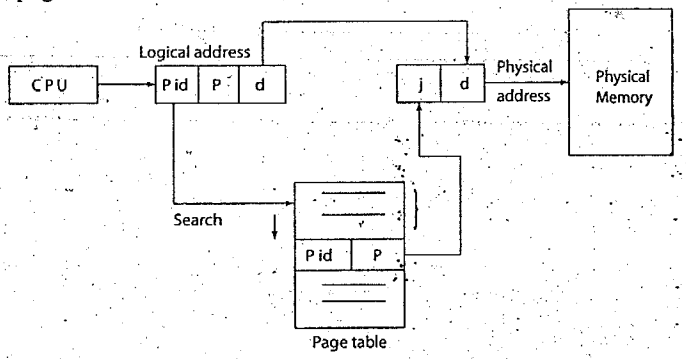
Common in address spaces > 32 bits, the virtual page number is hashed in to a page table. This page table contains a chain of elements hashing to the same Location. virtual page numbers are compared in this chain searching for a match if a match is found, the corresponding physical frame is extracted.



3. Inverted page table.

One entry for each read page of memory. Entry consists of the virtual address of the page stored in that real memory location, with information about the process that own that page. Decreases memory needed to store each page table, but increases time needed to

search the table when a page reference occurs, use hash table to limit the search to one or at most a few page table entries.



UNIT 5

VIRTUAL MEMORY MANAGEMENT

SYLLABUS

1. Background.
2. Demand Paging.
3. Copy-on-Write.
4. Page Replacement.
5. Allocation of Frames.

SYNOPSIS

1. Background.

Code needs to be in memory to execute, but entire program rarely used, Error code, unusual routines, large data structures. Entire program code not needed at same time, Consider ability to execute partially-loaded program. Program no longer constrained by limits of physical memory. Program and programs could be larger than physical memory.

Virtual memory- Separation of user logical memory from physical memory. Only part of the program needs to be in memory for execution. Logical address space can therefore be much larger than physical address space. Allows address spaces to be shared by several processes. Allows for more efficient process creation. More programs running concurrently. Less I/O needed to load or swap processes. Virtual memory can be implemented via Demand paging, Demand segmentation.

2. Demand Paging.

Could bring entire process into memory at load time. Or bring a page into memory only when it is needed. Less I/O needed, no unnecessary I/O, Less memory needed, Faster response, More users.

Page Fault.

If there is a reference to a page, first reference to that page will trap to operating system is page fault.

Demand Paging.

Extreme case - start process with no pages in memory, OS sets instruction pointer to first instruction of process, non-memory-resident - page fault, And for every other process pages on first access is Pure demand paging.

3. Copy-on-Write. (COW) allows both parent and child processes to initially share the same pages in memory. If either process modifies a shared page, only then is the page copied. COW allows more efficient process creation as only modified pages are copied.

4. Page Replacement.

Prevent over-allocation of memory by modifying page-fault service routine to include page replacement, Use **modify (dirty) bit** to reduce overhead of page transfers - only modified pages are written to disk.

5. Allocation of Frames.

Each process needs minimum number of frames. Two major allocation schemes.

a. fixed allocation.

b. priority allocation.

5 Marks Questions and Answers

1. Explain the virtual memory concept with a neat diagram.

Ans. The ability to load only the portions of processes that were actually needed (and only when they were needed) has several benifits.

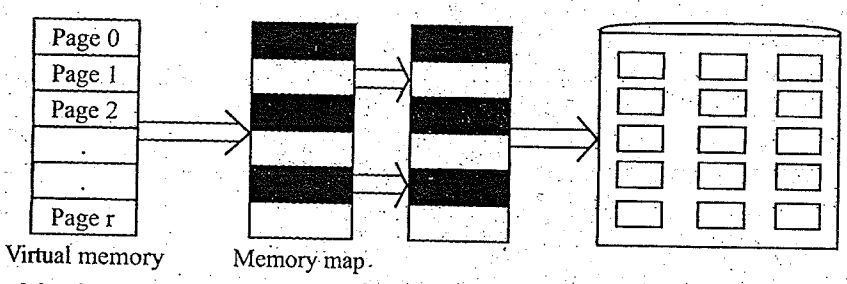
* Programs could be written for a much larger address space (virtual memory space) than physically exists on the computer.

* Because each process is only using a fraction of their total address space, there is more memory left for other programs, improving CPU utilization and system throughput.

* Less I/O needed for swapping processes.

* In and out of RAM, speeding things up.

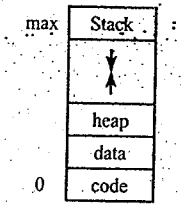
figure shows the general layout of virtual memory: which be much larger than physical memory.



2. Explain virtual address space with a neat diagram. **(5Marks)**

Ans. Figure shows virtual address space, which is the programmers logical view of process memory storage. The actual physical layout is controlled by the process page table.

The address show in figure is a sparse. A great hole in the middle of the address space is never used, unless the stack and/or the heap grow to fill the hole.

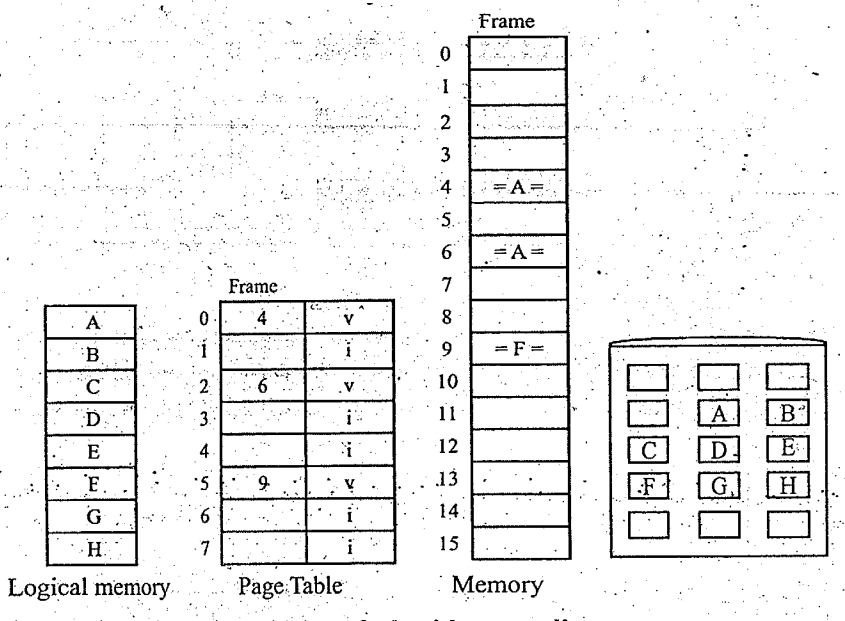


Pages can be shared during for k( ), speeding process creation. Shared memory by mapping pages read - write into virtual address space.

3. Explain demand paging with a neat diagram.

Ans. A demand paging system is similar to "paging system with swapping". A swapper manipulate entire processes, where as a pager is concerned with the individual pages of a process. The appropriate term in demand paging is the pager rather than swapper. Instead of swapping in a whole process, the pager bring only those necessary pages into memory. Thus it avoids reading of pages from the hard disk that will not be used.

With this scheme some sort of hardware support should be needed to differentiate between those pages that are in memory and those pages that are on the disk. The valid/ invalid bits can be used for this purpose. When the bit is set to valid, the value indicates that associated page is both legal and in memory. If the bit is invalid then this value indicate that the page either is not valid or is valid is currently on the disk.



4. Explain the steps for handling page fault with a neat diagram.

The procedure for handling a page fault is as follows.

1. Check the internal table (which is usually kept with process control block). for a process to determine whether the reference is a valid or invalid memory access.

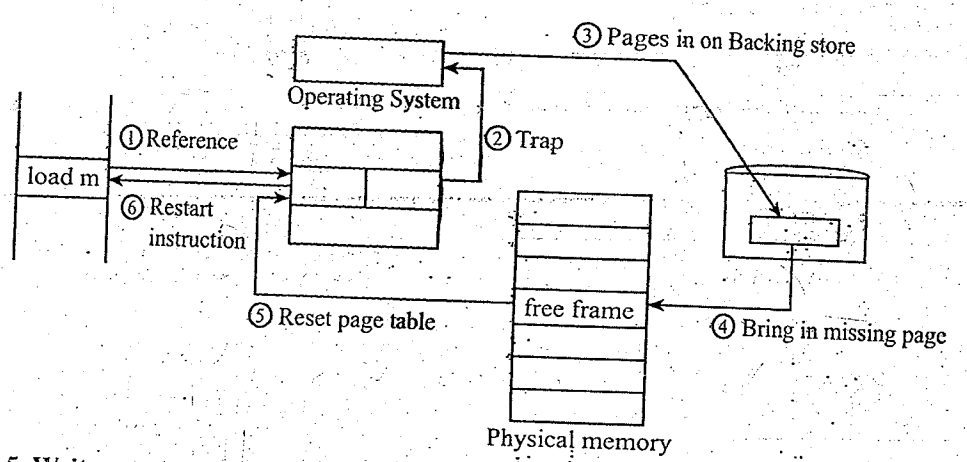
2. If the reference is invalid, we terminate the process. If it is valid, but we have not yet brought in that page, bring the page into memory.

3. Find a free frame for the required page which has to be brought into the memory.

4. A disk operation (disk scheduling) is initiated to read the desired page into the newly allocated frame.

5. After the page is brought into the main memory modify the internal table. Also modify the page table to indicate that the page is now in memory.

6. Now page is in memory the instruction will be restarted.

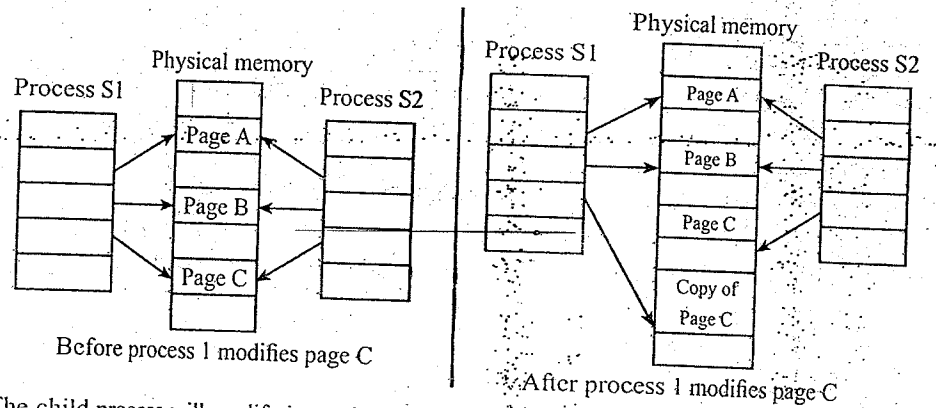


5. Write a not an copy-on-write.

Ans. The fork command system call creates a child process as a duplicate of its parent considering that many child process invoke the exec() system call immediately after the copying of the parents address space may be unnecessary.

Alternative we can used the technique known as copy-on-write, which works by allowing the parent and child process initially to share the same page.

These shared pages are marked as copy on write pages, meaning that if either process writes to a shared page, a copy of the share page is created.

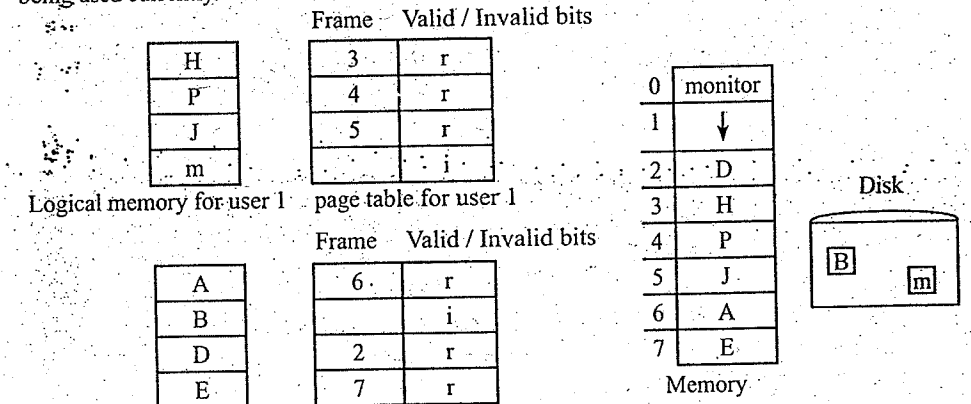


The child process will modify its copies page and not the page belonging to the parent process. Copy on write is a common techniques used by, several OS including windows XP, Linux etc. Many OS provides a pool of free pages on request these free pages are typically allocated when

the stacker heap for a process must expand or when there are copy on write pages to be managed. OS typically allocate these pages using a technique known as zero fill on demand.

6. Explain the need for page replacement.

While a user process is executing a page fault occurs. As a result the hardware traps to the OS. The OS determines where the desired page is placed on the disk. But then it finds no free frames and all memory is in use. Under these conditions the OS has several options. One option is to terminate the process because there is no memory space for the desired page. But this is not the best choice. Another choice is to find a page that is not currently being used. so free it . Thus in order to bring a desired page into the main memory, it is necessary to replace a page which is not being used currently.



7. How frames are allocated. Explain,

Ans. In the paged allocation a process is allocated the number of frames required by the process. In demand paging the frames are allocated to the process only on demand. The techniques are

1. Minimum number of frames allocation

2. Allocation algorithms. (a) Equal allocation (b) Proportional allocation

3. Global v/s Local allocation

1. Minimum number of frames allocation: To execute a process at least a minimum number of frames must be allocated, one of the reason for allocating at least a minimum number of frames involves performance.

As the number of frames allocated to each process decreases, the page fault rate, which slows down the process execution.

2(a). Equal allocation: It is the easiest way of frame allocation. In this "m" frames are equally divided among "n" process to give every process on equal share of frames.

2(b). Proportional allocation: The different process have different memory needs. This algorithm allocate available memory to each process based on its size. Let the size of virtual memory for a process pi is Si then the total number of frames are defined by $S = \sum Si$.

If the total number of available space is m then Ai frames are allocated to process pi where $Ai = Si \times m$.

3. Global v/s Local allocation: Local process use local replacement and global process use global replacement of page replacement algorithm.

Global replacement allows a process to select a replacement frame from the set of all the frames even if that frame is currently allocated to some other process.

Local replacement select the replacement frame only from its own set of allocated frames.

The problem with global replacement algorithm is that a process cannot control its own page fault rate.

8. Explain the step involved in the basic page replacement with diagram.

Ans. This involves the following steps.

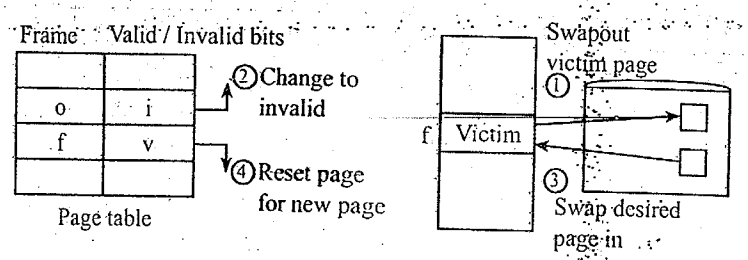
1. Find the location of the desired page on the disk.
2. Find a free frame.

(i). If there is a free frame, use it.

(ii). Otherwise, use a page replacement algorithm to select a "victim" frame.

(iii). Write the victim frame to the disk, change the page table accordingly.

3. Read the desired page into free frame. Since a new page has entered into the memory change the page table.
4. Restart the process.



10 Marks Questions and Answers

1. Consider the following reference string 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6. How many page fault would occur for the following page replacement algorithm assuming 3 page frames 1. Optimal page replacement. 2. FIFO. 3. LRU.

Optimal page replacement

Pages	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
Page frame	1	1	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
		2	2	2	2	2	2	2	2	2	2	2	7	7	7	2	2	2	2	2
			3	4	4	4	5	6	6	6	6	6	6	6	6	6	1	1	1	6
Page fault	✓	✓	✓	✓	No	No	✓	✓	No	No	No	✓	✓	No	No	✓	✓	No	No	✓

Page fault 11

FIFO

Pages	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
Page frame	1	1	1	4	4	4	4	6	6	6	6	3	3	3	3	2	2	2	2	6
		2	2	2	2	1	1	1	2	2	2	2	7	7	7	7	1	1	1	1
			3	3	3	3	5	5	5	1	1	1	1	6	6	6	6	6	3	3
Page fault	✓	✓	✓	✓	No	✓	✓	✓	✓	✓	No	✓	✓	✓	No	✓	✓	No	✓	✓

Page fault 16

LRU

Pages	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3
Page frame	1_0	1_1	1_2	4_0	4_1	4_2	5_0	5_1	5_2	1_0	1_1	1_2	7_0	7_1	7_2	2_0	2_1	2_0	2_1
		2_0	2_1	2_2	2_0	2_1	2_2	6_0	6_1	6_2	6_3	3_0	3_1	3_2	3_0	3_1	3_2	3_3	3_0
			3_0	3_1	3_2	1_0	1_1	1_2	2_0	2_1	2_0	2_1	2_2	6_0	6_1	6_2	1_0	1_1	1_2
Page fault	✓	✓	✓	✓	No	✓	✓	✓	✓	✓	No	✓	✓	✓	No	✓	✓	No	✓

Page fault 15

UNIT 6

FILE SYSTEM

SYLLABUS

1. File Concept.
2. Access Methods.
3. Directory Structure.
4. File-System Mounting.
5. File Sharing.
6. Protection.

SYNOPSIS

File Concept

Contiguous logical address space. Types are

1. Data
   a. numeric.
   b. character.
   c. binary.

File Attributes

1. **Name** - Only information kept in human - readable form.
2. **Identifier** - Unique tag (number) identifies file within file system.
3. **Type** - Needed for systems that support different types.
4. **Location** - Pointer to file location on device
5. **Size** - Current file size.
6. **Protection** - Controls who can do reading, writing, executing.
7. **Time, date, and user identification** - data for protection, security, and usage monitoring.
8. Information about are kept in the directory structure, which is maintained on the disk.

File Operations.

File is an **abstract data type. It's operations are Create, Write, Read, Reposition within file, Delete, Truncate.**

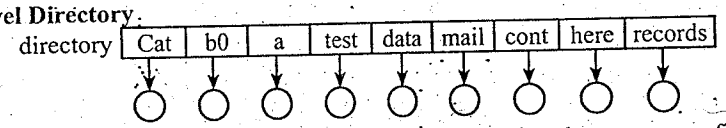
Disk Structure

Disk can be subdivided into partitions. Disks or partitions can be RAID protected against failure. Disk or partition can be used raw - without a file system, or formatted with a file system. Partitions

also known as mini-disks, slices. Entity containing file system known as a volume Operations Performed on Directory.

1. Search for a file.
2. Create a file.
3. Delete a file.
4. List a directory.
5. Rename a file.
6. Traverse the file system.

Single -Level Directory.



Two-Level Directory In two level directory structure each user has there own user file directory (UFD).

Tree-Structured Directories Tree-structured directory structure natural generalization of two level directory structure.

Acyclic-Graph Directories

Have shared subdirectories and files.

File System Mounting

A file system must be **mounted** before it can be accessed. A unmounted file system is mounted at a **mount point.**

File Sharing

Sharing of files on multi-user systems is desirable. Sharing may be done through a **protection** scheme. On distributed systems, files may be shared across a network. Network File System (NFS) is a common distributed file-sharing method.

File Sharing -Remote File Systems

Client -server model allows clients to mount remote file systems from servers. Server can serve multiple clients. Client and user-on-client identification is insecure or complicated. NFS is standard UNIX client-server file sharing protocol. CIFS is standard Windows protocol. Standard operating system file calls are translated into remote calls.

Distributed Information Systems **(distributed naming services)** such as LDAP, DNS, NIS, Active Directory implement unified access to information needed for remote computing.

Protection

File owner/creator should be able to control, what can be done, by whom, Types of access are **Read, Write, Execute, Append, Delete, List.**

5 Marks Questions and Answers

1. Briefly explain the operations that can be performed on files.

Ans. The various operation that can be performed on file are listed below.

1. Creating a file:-This is a two step operation

* In the first step, space in the file system must be found for the file.
* In the second step, an entry for new file must be made in the directory.

2. Writing a file.

To write file, system call is used by specifying the name of the file and the information to be written to the file.

Even the name of the file is searched in the directory by the system to find the file location and key write pointer to the location.

3. Reading a file:-To read information into the file READ system call is used. It specify name of the file. During read operation, when a file name is given to the directory is searched for the associated entry and the file pointer is set to the first character of the file.

4. Repositioning with in a file:-During this operation the directory is searched for the appropriate entry of the file in the directory given file name. The current file position pointer is set to the new position. Repositioning with in a file does not invoke any I/O. This operation is also called as file seek.

5. Deleting a file:-To delete a file DELETE system call is used. It specify the name of the file then the associated directory entry is searched to identify the file and the file is deleted. During deletion operation the contents of the file is erased and the associated space is added to the free space on the disk.

6. Truncating a file:-During this operation some contents of the file is deleted. During this operation except the size attributes all other attributes remain same.

2. What are the different operations performed on a directory.

Ans. Operations performed on directory

1. Search for a file.

Files have a symbolic names and similar names, which may indicate relationship between files.

It enables to search a directory structure to find the entry for a particular file and also to find all files whose names match a particular pattern.

2. Create a file.

New files need to be created and added to the directory.

3. Delete a file.

When a file is no longer needed, a file can be removed from the directory.

4. List a directory.

It is possible to list the files in a directory and contents of directory entry for each file in the list.

5. Rename a file.

The name of a file represents its contents to its users.

It enables the file name to be changed when the contents or use of the file changes.

Renaming allows its position to be changed with in the directory structure.

6. Traverse the file system.

To access every directory and every file with in a directory structure, the contents and structure of the file system should be saved at regular interval.

If a file is no longer in user, the file can be copied to the tape and disk space of that file is released for reused by another file.

3. Write a note on remote file system.

Ans. The advance of network allowed communication between remote computers.

Networking allows the sharing of resources spread with in a campus or even around the world.

With the evolution of network and file technology. the file sharing methods have changed.

In the first implemented method, user manually transfer files between machines via protocols like "FTP"

In the second method a DFS (Distributed file system) is used, in which remote directories are visible from the local machine.

The third method is www (World Wide Web). In which an anonymous access is allow a user to transfer files without having an account on remote system. Thus is uses anonymous file exchange.

4. Discuss briefly client- server model of file sharing.

Ans. *When one computer system remotely mounts a file system that is physically located on another system, the system which physically owns the files acts as a server, and the system which mounts them is the client.

*User ID and group IDs must be consistent across both systems for the system to work properly.

*The same computer can be both a client and a server.

*Servers may restrict remote access to read-only.

*Servers restrict which file systems may be remotely mounted. Generally the information within those subsystems is limited, relatively public and protected by frequent back ups.

* Servers commonly restrict mount permission to certain trusted systems only.

10 Marks Questions and Answers

1. Define file and explain the different file attributes.

Ans. A file is an abstract data type defined and implemented by the operating system. it is a sequence of logical records. A logical record may be a byte, a line or a more complex data item. The file attributes are,

1. Name: Is the file name, which is the only information kept in human readable form.

2. Identifier: Is a number which identifies the file and is a non readable name of the file.

3. Type: Is required for system which support different file types.

Ex:-exe, .obj

4. Location: Points to the device and location of the file on that device.

5. Size: Is the size of the file in bytes, words or blocks. It also includes the maximum allowed size of the file.

6. Protection: Is given to the file by access control information which determines who is allowed to read, write or execute the file.

7. Date, time and user identification: Contains information regarding creation of file.

8. Last modified and last usage: This data is used for protection security and monitoring file usage.

2. List out common file types with their extension and functions.

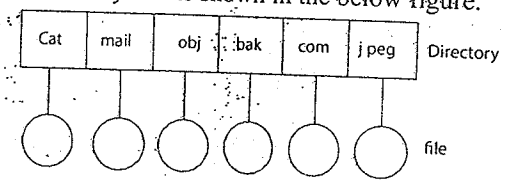
Ans. Common file types are,

Type	Usual extension	Function
Executable	exe, com, bin	Read to reaching Language program
Object	obj, O	Compiled machine language not linked
Source code	c, asm, a, cpp, java	Source code in various language
Batch	bat, sh	Commands to the command interpreted
Text	txt, doc	External data comments
Word processor	wp, tex, ref	Various word processor formats
Library	lib,d	Libraries of routine for programmers
Print or view	p s, dvi, dif	ASCII or binary file in a format for printing or viewing
Archive archiving	arc, zip, tar	Related files grouped into one file, sometimes compressed for archiving or storage.
Multi-media	jpeg, mpeg, mps	Binary file containing audio or video information

3. Explain single-level directory and Two-level directory with a neat diagram.

Ans. Single-level directory

Single Level directory structure is the simplest directory structure in which all the files are stored in the same directory this is shown in the below figure.



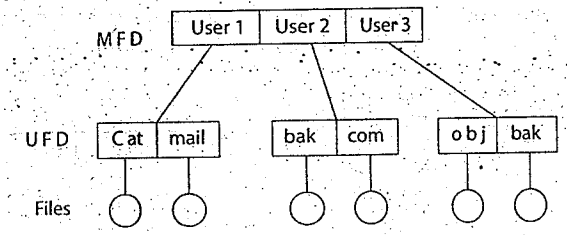
As shown in the Fig. multiple user files are stored in the same directory. as all the files of the directory must have unique file name. which is difficult for the user to find different file names and to remember all the file names.

Hence to overcome above disadvantages and the arrange the files in organised way two level directory structure is used.

Two level directory structure.

The major disadvantage of single level directory structure is confusion of file name between different users. The standard solution to solve the above problem is to create separate directory for each user.

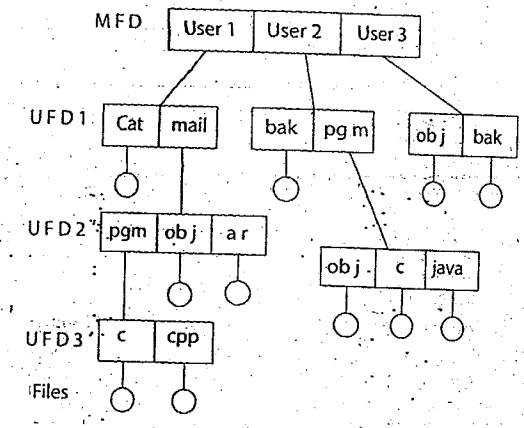
In two level directory structure each user has there own user file directory (U F D). Each user as similar structure and list the files of single user. When ever the user locks into the system the master file directory (M F D) is searched to identify the individual user by user name. Then the user can access his own UFD. This is illustrated in the figure.



The two level directory structure solve the problem of name collision and isolates the user from one another. this isolation creates a problem when many user wants to share a common file.

4. Explain tree-structured directory and a cyclic-graph directory with a neat diagrams.

Ans. The tree-structured directories are the natural generalization of two level directory structure. This generalisation allows the user to create there own sub directories/and to organize their files accordingly. Tree structured directory has a root directory in this organization every file in the system must have unique path name, A path name is path from the root through all sub directories to a specified file.

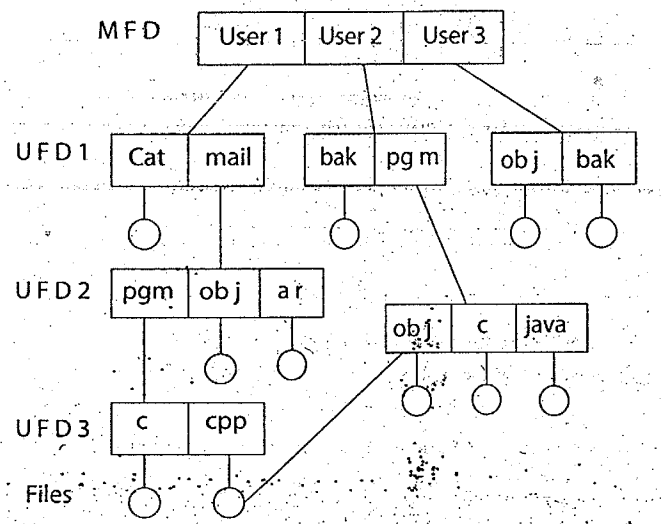


For example, path name for .cpp file of user1 is user1\mail\pgm\cpp. There exists only one path name for the file. The tree structure isolates one user from another but it does not provide file sharing between the users. Hence a cyclic graph directory structure is used.

A cyclic Graph directory Structure: A major disadvantage of tree structure directory is it suffers from isolation of files of different users. It does not provide file sharing.

Consider two programmers, who are working on the same project, In the tree structured directory each programmer must have the own copies of a file sharing is not permitted. This leads to wastage of memory space. It is very difficult to maintain same copy of file during file modification. To avoid such problems a cyclic graph directory is used which supports file sharing.

In a cyclic graph directory only one copy of the shared file can be maintained which can be shared among many user, In this type each user access original copy of the file and any change to the file is immediately reflected and the other user can access the changed file. A cyclic graph directory structure is more flexible than tree structured directory but it is more complex to implement a cyclic graph directory is shown in the figure, in which a file in the system may have more on one path name.



5. Differentiate between sequential access and direct access methods.

Ans.

Sequential access	Direct access
1. Simplest access method is the sequential called access	1. Direct access method is also called access as relative access.
2. In formation the files processed in order, one record after the other	2. It is based on a disk model of file, because disks allow random access to any file block.
3. A read operation read next-reads the next record of the file and automatically advances the file pointer.	3. In this model, the file is viewed as a numbered sequence of blocks or records.
4. The write operation: Write next- appends the record to the end of the file and advances to the file operation to end of the newly written record.	4. Direct access method does not impose any restriction in the order of reading and writing records, Records are read and written randomly.
5. Sequential files can only be reset to the beginning of the file.	5. A block number provided by the user to the OS is a relative block number A relative block number is an index relative to the beginning of the file.
6. Sequential files are typically used in batch processing and pay roll applications.	6. The relative block number allows the OS to decide Where the file should be placed and helps in preventing the user from accessing parts of the file system that may not be a part of the file.

QUESTION PAPERS

DTE MODEL QUESTION PAPER (WITH ANSWER)

Diploma in OPERATING SYSTEM

IV Semester Operating System

Time: 3 Hrs **Marks: 100**

PART-A

Answer any 6 questions. Each carries 5 marks. **5 x 6 = 30 Marks.**

1. Write a note on protection and security.

Ans. Ref. Q3 Introduction to Operating System (unit-1.)

2. What is a process . Draw and explain the state transition diagram of a process.

Ans. Ref. Q1 Process Management (unit-2).

3. Explain the different scheduling criteria.

Ans. Ref. Q4 Process management (unit- 2).

4. Explain the 3 requirements for the solution to critical-section problem.

Ans. Ref. Q2 Synchronization and dead locks (unit-3).

5. Explain how to recover from deadlock.

Ans. Ref. Q7 Synchronization and dead locks (unit-3).

6. Explain address binding with a neat diagram.

Ans. Ref. Q7 Memory management strategies (unit-4).

7. Write a not on shared pages.

Ans. Ref. Q1 Memory management strategies (unit-4).

8. Explain the steps involved in the basic page replacement with diagram.

Ans. Ref. Q8 Virtual memory management (unit-v).

9. Briefly explain the operations that can be performed on files.

Ans. Ref. Q1 File system (unit-6).

Part-B

Answer any SEVEN full questions each carries 10 marks. **10 × 7 = 70 Marks**

1. Compare process management and memory management.

Ans. Ref. Q4 Introduction to operating system (unit-1).

2. Explain clustered system.

Ans. Ref. Q7 Introduction to operating system (unit-1).

3. Explain the different operations on processes.

Ans. Ref. Q1 Process management (unit-2).

4. Consider the following set of process with the length of the CPU burst time given in milliseconds.

Process	Burst time	Priority
P1	6	2
P2	8	1
P3	6	3
P4	3	1

FCFS

P1	P2	P3	P4

0 6 14 20 23

Process	W.T	T.T
P1	0	6
P2	6	14
P3	14	20
P4	20	23

$$\text{average waiting time} = \frac{0+6+14+20}{4} = 10\text{ms}$$

$$\text{average turnaround time} = \frac{6+14+20+23}{4} = 15.75\text{ms}$$

Priority Scheduling

P2	P4	P1	P3

0 8 11 17 23

Process	Waiting time	Turnaround time
P1	11	17
P2	0	8
P3	17	23
P4	8	11

$$\text{average waiting time} = \frac{11+0+17+8}{4} = 9\text{ms}$$

$$\text{average turnaround time} = \frac{17+8+23+11}{4} = 14.75\text{ms}$$

5. How deadlock can be detected. Explain.

Ans. Ref. Q1 Synchronization and deadlocks (unit-3).

6. Explain hardware implementation of page table with a neat diagram.

Ans. Ref. Q2 Memory management strategies (unit-4).

7. List and explain different structure of the page table with a neat diagram.

Ans. Ref. Q3 Memory management strategies (unit-4).

8. Consider the following reference string 1 2 3 4 1 2 5 1 2 8 4 5. How many page faults would occur for the following page replacement algorithm assuming page frames. 1. LRU. 2. FIFO. 3. Optimal page replacement.

Ans.

1. FIFO

Pages	1	2	3	4	1	2	5	1	2	8	4	5
	1	1	1	1			5	5	5	5	4	4
Pages		2	2	2			2	1	1	1	1	5
frames			3	3			3	3	2	2	2	2
				4			4	4	4	8	8	8

Page fault = 10.

2. CRU.

Pages	1	2	3	4	1	2	5	1	2	8	4	5
	0^1	1^1	2^1	3^1	0^1	1^1	2^1	0^1	1^1	2^1	3^1	0^5
Pages		0^2	1^2	2^2	3^2	0^2	1^2	2^2	0^2	1^2	2^2	3^2
frames			0^3	1^3	2^3	3^3	0^5	1^5	2^5	3^5	0^4	1^4
				0^4	1^4	2^4	3^4	4^4	5^4	0^8	1^8	2^8

Page fault = 8

3. Optimate

Pages	1	2	3	4	1	2	5	1	2	8	4	5
	1	1	1	1			1			8		
Pages		2	2	2			2			2		
frames			3	3			5			5		
				4			4			4		

Page fault = 6

9. Differentiate between sequential access and direct access methods.

Ans. Ref. Q3 File system (unit-6).

10. Define file and explain the different file attributes.

Ans. Ref. Q1 File system (unit-6).

SUPER MODEL QUESTION PAPER (WITH ANSWERS)

Diploma in OPERATING SYSTEM

IV Semester Operating System

Time: 3 Hrs **Marks: 100**

PART-A

Answer any 6 questions. Each carries 5 marks. **5 x 6 = 30 Marks**

1. Write a note on client/server computing.
Ans. Ref. Q5 Introduction to operating system. (unit-1).

2. Explain the contents of PCB with neat diagram.
Ans. Ref. Q2 Process Management (unit- 2).

3. Explain the different scheduling criteria.
Ans. Ref. Q4 Process Management (unit- 2).

4. Write a note on critical - section problem and the solution to solve it.
Ans. Ref. Q3 Synchronization and deadlocks (unit-3).

5. How deadlock can be avoided using resource allocation graph. Explain.
Ans. Ref. Q6 Synchronization and deadlocks (unit-3).

6. What is fragmentation. Differentiate between internal and external fragmentation.
Ans. Ref. Q3 Memory Management strategies (unit-4).

7. Explain swapping technique with a neat diagram.
Ans. Ref. Qn 8. Memory Management Strategies (unit-4).

8. Write a note on copy-on-write.
Ans. Ref. Q5 Virtual Memory Management (unit-5).

9. Discuss briefly client-server model of file sharing.
Ans. Ref. Q4 File System (unit-6).

Part-B

Answer any SEVEN full question each carries 10 Marks. **10 × 7 = 70 Marks**

1. What are the different operating system operation.
Ans. Ref. Q2 Introduction to operating system (unit-1).

2. Different activities in connection with process management, memory management and storage management.
Ans. Ref. Q6 Introduction to operating system (unit-1).

3. Consider the following set of process with the length of the CPU burst time given in milliseconds.

Process	Burst time	Priority
P1	10	3
P2	13	1
P3	3	3
P4	8	4

The process have arrived in the order P1, P2, P3, P4, all at time 0.
(a) Draw the Gantl chart for FCFS and PRIORITY scheduling algorithms.
(b) What is the waiting time and turnaround time of each process for FCFS and PRIORITY scheduling algorithms.
(c) Calculate the average waiting time and average turnaround time for FCFS and PRIORITY scheduling algorithms.
Ans. Ref. Q3 Process Management (unit-2).

4. Explain the different operations on process.
Ans. Ref. Q1 Process management (unit-2).

5. Explain the banker's algorithm.
Ans. Ref. Q2 Synchronization and deadlocks.

6. List and explain different structure of the page table with a neat diagram.
Ans. Ref. Q3 Memory management strategies (unit-4).

7. Explain the contiguous memory allocation.
Ans. Ref. Q1 Memory Management Strategies (unit-4).

8. Consider the following reference string 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6. How many page would occur the following page replacement algorithm assuming 3 page frames. 1. CRU. 2. FIFO. 3. Optimal page replacement.
Ans. Ref. Q1. Virtual memory management. (unit-5)

9. List out common file types with their extension and functions.
Ans. Ref. Q2. File System (unit-6).

10. Explain tree-structured directory and Acyclic-graph directly with a neat diagram.
Ans. Ref. Q5 File system (unit-6).

SUPER MODEL PRACTICE QUESTION PAPER

Diploma in OPERATING SYSTEM

IV Semester Operating System

Time: 3 Hrs Marks: 100

PART-A

Answer any 6 questions. Each carries 5 marks. 5 x 6 = 30 Marks. I.

Explain OS in user's view and system's view.

2. What is a process. Draw and explain the state transition diagram of a process.
3. Compare long-term, short-term and medium-term scheduler.
4. Write a note on semaphores.
5. What is deadlock. What are the necessary condition for deadlock.
6. Explain first-fit, best-fit and worst-fit strategies for memory allocation.
7. Differentiate between logical and physical address space.
8. Explain demand paging with a neat diagram.
9. What are the different operations performed on a directory.

Part-B

Answer any SEVEN full question each carries 10 marks. 10 × 7 = 70 Marks

1. What is an operating system. Explain the computer system architecture.
2. Explain the special-purpose system.
3. What is IPC. Explain the two models of IPC with neat diagram.
4. Consider the following set of process with the length of the CPU burst time given in milliseconds.

Process	Burst time	Priority
P1	6	4
P2	8	1
P3	7	3
P4	3	2

The processes have arrived in the order P1, P2, P3 and P4, all at time 0.

(a) Draw the Gantt chart for FCFS and PRIORITY scheduling algorithms.

(b) What is waiting time and turnaround time of each process for FCFS and PRIORITY scheduling algorithm.

(c) Calculate the average waiting time and average turnaround time for FCFS and PRIORITY scheduling algorithms.

5. Explain resource - allocation graph with deadlock and without deadlock with examples.
6. What is segmentation. Draw and explain its hardware.
7. Explain hardware support implementation of page table with a neat diagram.
8. Consider the following reference string. 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1. How many page fault would occur for the following page replacement algorithm assuming 3 page frames 1. LRU, 2. FIFO, 3. Optimal page replacement.
9. Define file and explain the different file attributes.
10. Explain single - level directory and two - level directory with a neat diagrams.

DIPLOMA EXAMINATION PAPER Nov./Dec. 2016

IV Semester

OPERATING SYSTEM

Time: 3 Hrs Marks: 100

PART-A

Answer any 6 questions. Each carries 5 marks. 5 x 6 = 30 Marks

1. Explain OS in user's view and system's view.

Ans. Ref. Q1. Unit-1 (5 Marks).

2. What is a process? Draw and explain the state transition diagram of a process.

Ans. Ref. Q1. Unit-2 (5 Marks).

3. Explain the different scheduling criteria.

Ans. Ref. Q4. Unit-2 (5 Marks).

4. Write a note on Semaphores.

Ans. Ref. Q1. Unit-3 (5 Marks).

5. How deadlock can be avoided using resource allocation graph? Explain.

Ans. Ref. Q6. Unit-3 (5 Marks).

6. Explain first-fit and worst-fit strategies for memory allocation.

Ans. Ref. Q2. Unit-4 (5 Marks).

7. Explain swapping technique with a neat diagram.

Ans. Ref. Q7. Unit-4 (5 Marks).

8. Explain how frames are allocated.

Ans. Ref. Q7. Unit-5 (5 Marks).

9. Briefly explain the operations that can be performed on files.

Ans. Ref. Q1. Unit-6 (5 Marks).

Part-B

Answer any SEVEN full question each carries 10 marks. 10 × 7 = 70 Marks

10. Explain the functions of memory management and process management.

Ans. Ref. Q4. Unit-1 (10 Marks).

11. Explain clustered system.

Ans. Ref. Q6. Unit-1 (10 Marks)

12. Consider the following set of process with the length of the CPU burst time given in milliseconds.

Process	Burst time	Priority
P1	10	3
P2	1	1
P3	2	4
P4	1	5
P5	5	2

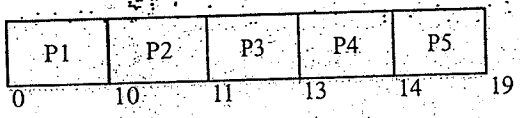
The processes are assumed to have arrived in the order P1, P2, P3 all at time 0.

(a) Draw the Gantt chart for FCFS and PRIORITY scheduling algorithms.

(b) What is the waiting time of each process for FCFS and PRIORITY scheduling algorithm.

(c) Calculate the average waiting time for FCFS and PRIORITY scheduling algorithms.

Ans. FIFO

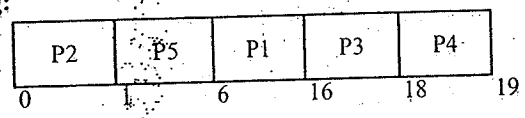


Process	Waiting time	Turnaround time
P1	0	10
P2	10	11
P3	11	13
P4	13	14
P5	14	19

$$\text{Avg. waiting time} = \frac{0+10+11+13+14}{5} = 9.6$$

$$\text{Avg. turnaround time} = \frac{10+11+13+14+19}{5} = 13.4$$

Priority Scheduling:



Process	Waiting time	Turnaround time
P1	6	16
P2	0	01
P3	16	18
P4	18	19
P5	01	06

$$\text{Avg. waiting time} = \frac{6+0+16+18+1}{5} = 8.2$$

$$\text{Avg. turnaround time} = \frac{16+1+18+19+6}{5} = 12$$

13. What is Inter-Process Communication (IPC)? Explain the two models of IPIC.

Ans. Ref. Q2. Unit-2 (10 Marks)

14. What is deadlock can be detected? Explain.

Ans. Ref. Q1. Unit-3 (10 Marks)

15. Explain hardware support implementation of page table with a neat diagram.

Ans. Ref. Q2. Unit-4 (10 Marks)

16. Explain the contiguous memory allocation.

Ans. Ref. Q1. Unit-4 (10 Marks)

17. Consider the following reference string:

5 3 2 7 3 4 3 8 7 4 2 3 5 2 7

How many page faults could occur for the following page replacement algorithm assuming 4 page frames?

1. LRU 2. FIFO 3. Optimal page replacement

Ans. 1. LRU

5	3	2	7	3*	4	3*	8	7*	4*	2	3	5	2*	7
0^5	1^5	2^5	3^5	4^5	0^4	1^4	2^4	3^4	0^4	1^4	2^4	3^4	4^4	0^7
	0^3	1^3	2^3	0^3	1^3	0^3	1^3	2^3	3^3	0^2	1^2	2^2	0^2	1^2
		0^2	1^2	2^2	3^2	4^2	0^8	1^8	2^8	3^8	0^3	1^3	2^3	3^3
			0^7	1^7	2^7	3^7	4^7	0^7	1^7	2^7	3^7	0^5	1^5	2^5

Page fault = 10

2. FIFO

↓	↓	↓	↓		↓		↓					↓	↓	↓	↓
5	3	2	7	3	4	3	8	7	4	2	3	5	2	7
5	5	5	5		4		4				4	4	2	2
	3	3	3		3		8				8	8	8	7
		2	2		2		2				3	3	3	3
			7		7		7				7	5	5	5

Page fault = 10

3. Optimal

↓	↓	↓	↓		↓		↓				↓	↓		
5	3	2	7	3	4	3	8	7	4	2	3	5	2	7
5	5	5	5		4		4				3	3		
	3	3	3		3		8				8	5		
		2	2		2		2				2	2		
			7		7		7				7	7		

Page fault = 8

18. List out common file types with their extension and functions.

Ans. Ref. Q2. Unit-6 (10 Marks)

19. Explain tree structured directory and acyclic graph directory with a neat diagram.

Ans. Ref. Q4. Unit-6 (10 Marks)