OS

Mid-1 Important questions:

Unit-1

1. **Define System Calls? Explain types of System Calls**

System Calls:

- System calls provide an Programming interface to the services provided by the OS.
- It is a programmatic method in which a computer program requests a service from the kernel of the OS.
- These calls are generally available as routines written in a high-level language (C or C++).
- System call offers the services of the operating system to the user programs via API. Mostly accessed by programs via a high-level Application Program Interface (API) rather than direct system call use
- Three most common APIs are Win32 API for Windows, POSIX API for POSIX-based systems (including virtually all versions of UNIX, Linux, and Mac OS X), and Java API for the Java virtual machine (JVM)

Types of System Calls:

There are five types of system calls. These are as follows:

- Process control
- File management
- Device management
- Information maintenance
- Communications

1) Process Control

This system calls perform the task of process creation, process termination, etc.

Functions:

- End and Abort
- Load and Execute
- Create Process and Terminate Process
- Wait and Signal Event
- Allocate and free memory

2) File Management

File management system calls handle file manipulation jobs like creating a file, reading, and writing, etc.

Functions:

- Create a file
- Delete file
- Open and close file
- Read, write, and reposition
- Get and set file attributes

3)Device Management

Device management does the job of device manipulation like reading from device buffers, writing into device buffers, etc.

Functions:

- Request and release device
- Logically attach/ detach devices
- Get and Set device attributes

4) Information Maintenance

It handles information and its transfer between the OS and the user program.

Functions:

- Get or set time and date
- Get process , file and device attributes
- Set process , file and device attributes

5)Communication:

These types of system calls are specially used for interprocess communications.

Functions:

- Create, delete communications connections
- Send, receive message
- Help OS to transfer status information
- Attach or detach remote devices

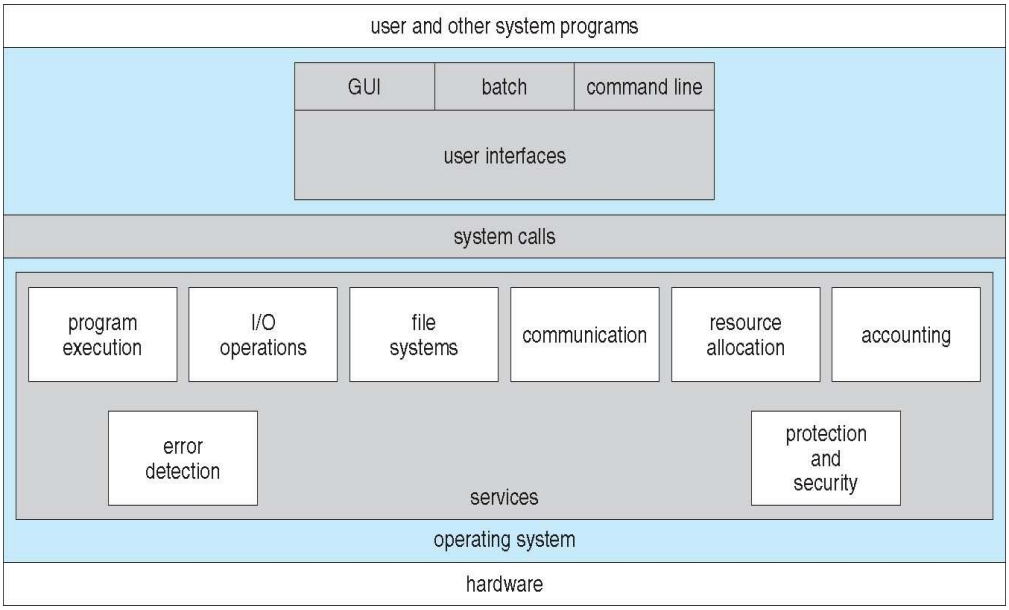
2. What is an Operating System? Explain the services of Operating System?

- An Operating System (OS) is an interface between a computer user and computer hardware.
- An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.
- Many popular Operating systems are Windows , Linux, Mac OS by Apple etc.
- Operating system goals:
    - Execute user programs and make solving user problems easier
    - Make the computer system convenient to use
    - Use the computer hardware in an efficient manner

Operating System Services:

- One set of operating-system services provides functions that are helpful to the user:
    - **User interface** - Almost all operating systems have a user interface (UI)

      Varies between Command-Line (CLI), Graphics User Interface (GUI), Batch

- **Program execution** - The system must be able to load a program into memory and to run that program, either normally or abnormally (indicating error)
- **I/O operations** - A running program may require I/O, which may involve a file or an I/O device
- **Communications** – Processes may exchange information, on the same computer or between computers over a network
- Communications may be via shared memory or through message passing (packets of information moved by the OS)
- **Error detection** – OS needs to be constantly aware of possible errors
- **Resource allocation -** When multiple users or multiple jobs running concurrently, resources must be allocated to each of them
- **Protection and security -** The owners of information stored in a multiuser or networked computer system may want to control use of that information. When separate processes are running concurrently, processes should not interfere with each other
- **Protection** involves ensuring that all access to system resources is controlled
- **Security** of the system from outsiders requires user authentication, extends to defending external I/O devices from invalid access attempts
- 

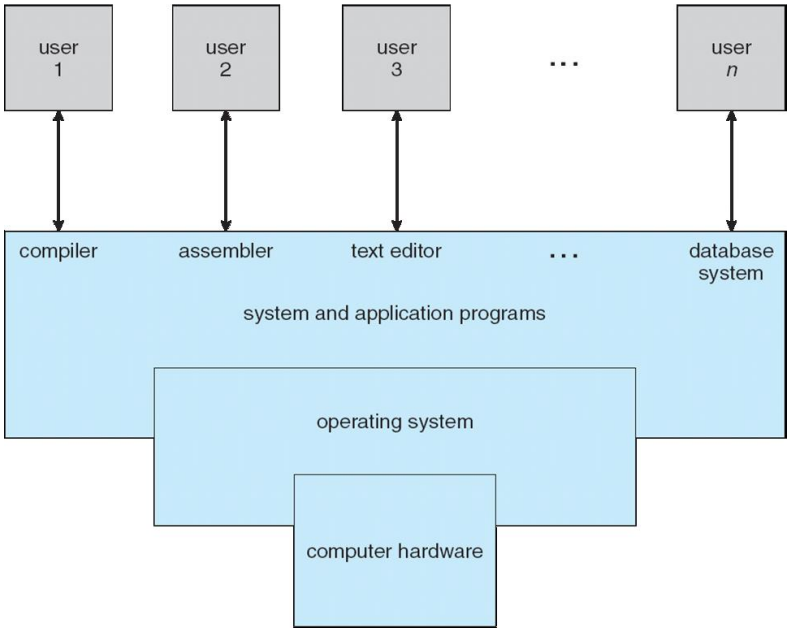
3. **Define Operating system? Explain abstract view of the components of a computer system.**

- An Operating System (OS) is an interface between a computer user and computer hardware.
- An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.
- Many popular Operating systems are Windows , Linux, Mac OS by Apple etc.
- Operating system goals:
    - Execute user programs and make solving user problems easier
    - Make the computer system convenient to use
    - Use the computer hardware in an efficient manner
- OS is a **resource allocator**
    - Manages all resources
    - Decides between conflicting requests for efficient and fair resource use
- OS is a **control program**
    - Controls execution of programs to prevent errors and improper use of the computer

Computer System Structure:

- Computer system can be divided into four components:
    - Hardware – provides basic computing resources
        - CPU, memory, I/O devices
    - Operating system
        - Controls and coordinates use of hardware among various applications and users
    - Application programs – define the ways in which the system resources are used to solve the computing problems of the users
        - Word processors, compilers, web browsers, database systems, video games
    - Users
        - People, machines, other computers

Four Components of a Computer System:



4. **Explain Operating System structure. (Simple and Layered structure)**

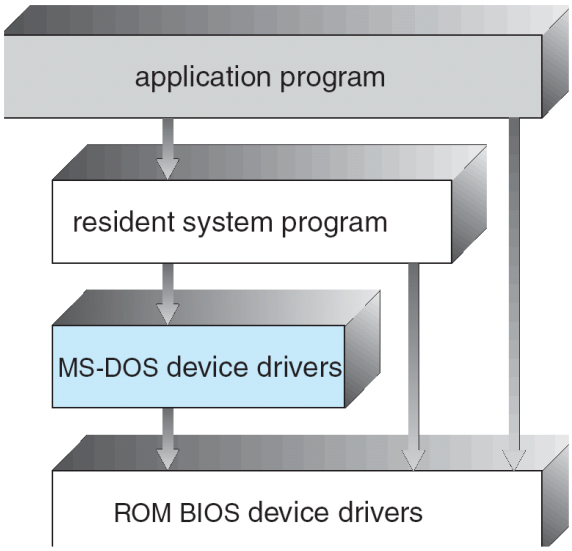
Operating System Structure:

Operating system can be implemented with the help of various structures. The structure of the OS depends mainly on how the various common components of the operating system are interconnected and melded into the kernel. Depending on this we have following structures of the operating system:

Simple Structure

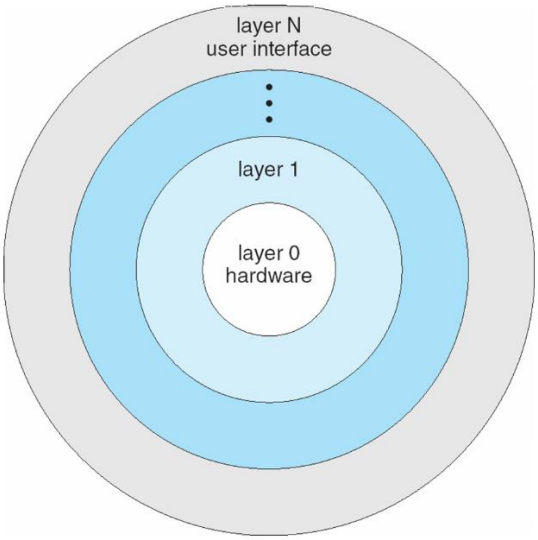
- More commercial operating systems do not have well defined structure.
- They are small, simple and limited systems. The interfaces and levels of functionality are not well separated. MS-DOS is an example of such operating system. In MS-DOS application programs are able to access the basic I/O routines. These types of operating system cause the entire system to crash if one of the user programs fails.
- MS-DOS – written to provide the most functionality in the least space
  - Not divided into modules
  - Although MS-DOS has some structure, its interfaces and levels of functionality are not well separated

MS-DOS Layer Structure:



Layered Approach:

- The operating system is divided into a number of layers (levels), each built on top of lower layers. The bottom layer (layer 0), is the hardware; the highest (layer N) is the user interface.
- With modularity, layers are selected such that each uses functions (operations) and services of only lower-level layers
- Each upper layer is built on the bottom layer
- Every layer performs different functions like Hardware , Cpu scheduling, process management, memory management etc,
- These layers are so designed that each layer uses the functions of the lower-level layers only. It simplifies the debugging process as if lower-level layers are debugged, and an error occurs during debugging. The error must be on that layer only as the lower-level layers have already been debugged.
- Advantage is It is very easy to perform debugging and system verification.



Unit-2

5. **Explain the states of process clearly with process state diagram.**

6. **What is multithreading? Explain different types of Multithreading models.**
7. **What is process scheduling? Explain the working of Priority Scheduling Algorithm for process given below and Find their average turnaround time and average waiting time.**

Process	Burst Time	Priority
P1	5	2
P2	4	1
P3	3	3
P4	6	4

8. **What are the different CPU process scheduling algorithms. Explain with examples. FCFS, SJF, PRIORITY , Round Robin Algorithms**
9. **Explain the significance of Process Control Block and describe its typical elements.**
10. **What is the need of Inter Process Communication mechanism? Explain the methods of IPC in detail.**

Unit-3

11. **What is Semaphore and Explain types of Semaphore . Discuss operations of semaphore?**

Semaphores in Process Synchronization

Difficulty Level : Easy • Last Updated : 22 Nov, 2021

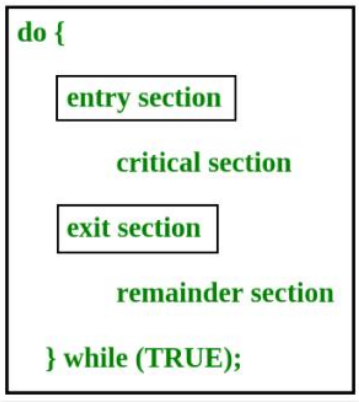
Semaphore was proposed by Dijkstra in 1965 which is a very significant technique to manage concurrent processes by using a simple integer value, which is known as a semaphore. Semaphore is simply an integer variable that is shared between threads. This variable is used to solve the critical section problem and to achieve process synchronization in the multiprocessing environment.
Semaphores are of two types:

1. **Binary Semaphore –**
   This is also known as mutex lock. It can have only two values – 0 and 1. Its value is initialized to 1. It is used to implement the solution of critical section problems with multiple processes.
2. **Counting Semaphore –**
   Its value can range over an unrestricted domain. It is used to control access to a resource that has multiple instances.

12. **Explain Critical section problem discuss the solutions for critical section with an example.**

Critical Section Problem

Critical section is a code segment that can be accessed by only one process at a time. Critical section contains shared variables which need to be synchronized to maintain consistency of data variables.



In the entry section, the process requests for entry in the **Critical Section.**

Any solution to the critical section problem must satisfy three requirements:

- **Mutual Exclusion** : If a process is executing in its critical section, then no other process is allowed to execute in the critical section.
- **Progress** : If no process is executing in the critical section and other processes are waiting outside the critical section, then only those processes that are not executing in their remainder section can participate in deciding which will enter in the critical section next, and the selection can not be postponed indefinitely.
- **Bounded Waiting** : A bound must exist on the 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 and before that request is granted.

13. What is race condition. Discuss how to solve race condition problems with an example.

Race Condition

When more than one processes are executing the same code or accessing the same memory or any shared variable in that condition there is a possibility that the output or the value of the shared variable is wrong so for that all the processes doing the race to say that my output is correct this condition known as a race condition. Several processes access and process the manipulations over the same data concurrently, then the outcome depends on the particular order in which the access takes place.

A race condition is a situation that may occur inside a critical section. This happens when the result of multiple thread execution in the critical section differs according to the order in which the threads execute.

Race conditions in critical sections can be avoided if the critical section is treated as an atomic instruction. Also, proper thread synchronization using locks or atomic variables can prevent race conditions.

Mid-2 Important questions:

Unit-3

1. **What is resource allocation graph. Explain about resource allocation graphs for deadlock with examples . How to use resource allocation graphs for deadlock detection.**
2. **Explain briefly deadlock avoidance algorithm with an example using safe and unsafe sequence? (Bankers algorithm)**
   Consider system with five processor P0 to P4 and 3 resources A, B and C, Resources type A has 10 instances, B has 5 instances and C has 7 instances. The snapshot at time T0 is given below and Available resources are of A, B, C are 3,3, 2 respectively.. Find the Safe sequence.

	ALLOTED			MAX		
	A	B	C	A	B	C
P0	0	1	0	7	5	3
P1	2	0	0	3	2	2
P2	3	0	2	9	0	2
P3	2	1	1	2	2	2
P4	0	0	2	4	3	3

3. **What is Deadlock? Explain conditions or characterization needed for deadlock?**

Deadlock is a situation which involves the interaction of more than one resources and processes with each other.

We can visualise the occurrence of deadlock as a situation where there are two people on a staircase. One is ascending the staircase while the other is descending. The staircase is so narrow that it can only fit one person at a time. As a result, one has to retreat while the others moves on and uses the staircase. Once that person is finished, the other one can use that staircase. But here, none of the people is willing to retreat and waits for the each other to retreat. None of them is able to use the staircase. The people here is the process and the staircase is the resource.

1. **Mutual Exclusion:**
   When two people meet in the landings, they can't just walk through because there is space only for one person. This condition to allow only one person (or process) to use the step between them (or the resource) is the first condition necessary for the occurrence of the deadlock.
2. **Hold and Wait:**
   When the 2 people refuses to retreat and hold their grounds, it is called holding. This is the next necessary condition for the the deadlock.
3. **No Preemption:**
   For resolving the deadlock one can simply cancel one of the processes for other to continue. But Operating System doesn't do so. It allocates the resources to the processors for as much time needed until the task is completed. Hence, there is no temporary reallocation of the resources. It is third condition for deadlock.
4. **Circular Wait:**
   When the two people refuses to retreat and wait for each other to retreat, so that they can complete their task, it is called circular wait. It is the last condition for the deadlock to occur.

4. **Explain strategies or methods to handle deadlocks?**

Methods of handling deadlocks :

There are three approaches to deal with deadlocks.

1. Deadlock Prevention
2. Deadlock avoidance
3. Deadlock detection

2. Deadlock Avoidance **:**

This approach allows the three necessary conditions of deadlock but makes judicious choices to assure that deadlock point is never reached. It allows more concurrency than avoidance detection

A decision is made dynamically whether the current resource allocation request will, if granted, potentially lead to deadlock. It requires the knowledge of future process requests. Two techniques to avoid deadlock :

1. Process initiation denial
2. Resource allocation denial

1. Deadlock Prevention **:**

The strategy of deadlock prevention is to design the system in such a way that the possibility of deadlock is excluded. Indirect method prevent the occurrence of one of three necessary condition of deadlock i.e., mutual exclusion, no pre-emption and hold and wait. Direct method prevent the occurrence of circular wait.

Prevention techniques –

Mutual exclusion – is supported by the OS.

Hold and Wait – condition can be prevented by requiring that a process requests all its required resources at one time and blocking the process until all of its requests can be granted at a same time simultaneously. But this prevention does not yield good result because :

- long waiting time required
- in efficient use of allocated resource
- A process may not know all the required resources in advance

No pre-emption – techniques for 'no pre-emption are'

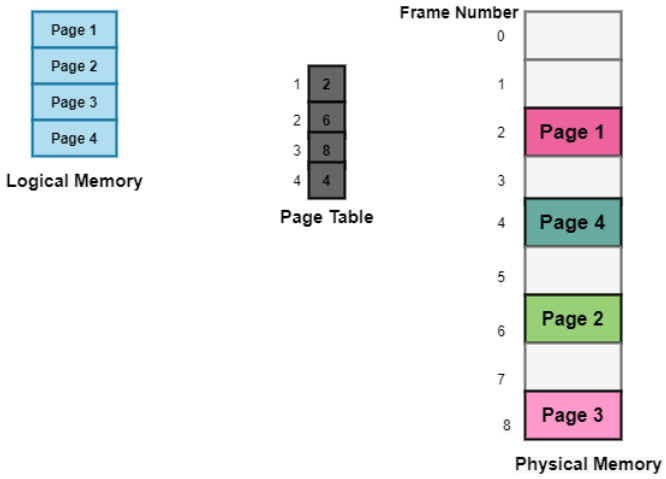
- If a process that is holding some resource, requests another resource that can not be immediately allocated to it, the all resource currently being held are released and if necessary, request them again together with the additional resource.
- If a process requests a resource that is currently held by another process, the OS may pre-empt the second process and require it to release its resources. This works only if both the processes do not have same priority.

Unit-4

5. **What is paging explain it. Explain structure of page table.**

Paging

- Paging is a memory-management scheme that permits the physical address space a process to be noncontiguous. Process is allocated physical memory whenever the latter is available
  - Avoids external fragmentation
  - Avoids problem of varying sized memory chunks
- Divide physical memory into fixed-sized blocks called **frames**
  - Size is power of 2, between 512 bytes and 16 Mbytes
- Divide logical memory into blocks of same size called **pages**
- Keep track of all free frames
- To run a program of size ***N*** pages, need to find ***N*** free frames and load program
- Set up a **page table** to translate logical to physical addresses
- Backing store likewise split into pages
- Still have Internal fragmentation

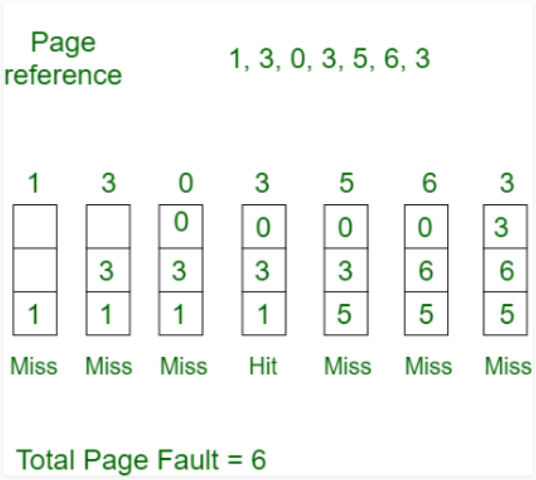


6. **Explain page replacement with examples of**
   1. **FIFO**

1. First In First Out (FIFO) –

This is the simplest page replacement algorithm. In this algorithm, the operating system keeps track of all pages in the memory in a queue, the oldest page is in the front of the queue. When a page needs to be replaced page in the front of the queue is selected for removal.

Example-1 Consider page reference string 1, 3, 0, 3, 5, 6 with 3 page frames. Find number of page faults.



Initially all slots are empty, so when 1, 3, 0 came they are allocated to the empty slots –> **3 Page Faults.**

when 3 comes, it is already in memory so –> **0 Page Faults.**

Then 5 comes, it is not available in memory so it replaces the oldest page slot i.e 1. –>**1 Page Fault.**

6 comes, it is also not available in memory so it replaces the oldest page slot i.e 3 –>**1 Page Fault.**

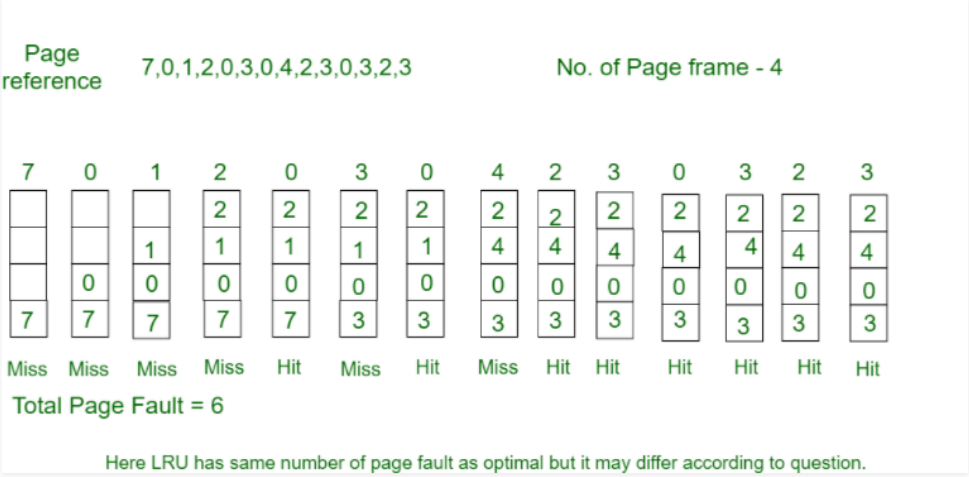
Finally when 3 come it is not available so it replaces 0 **1 page fault**

2.LRU

3. Least Recently Used –

In this algorithm page will be replaced which is least recently used.

Example-3 Consider the page reference string 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2 with 4 page frames.Find number of page faults.



Initially all slots are empty, so when 7 0 1 2 are allocated to the empty slots –> **4 Page faults**

0 is already their so –> **0 Page fault.**

when 3 came it will take the place of 7 because it is least recently used –>**1 Page fault**

0 is already in memory so –> **0 Page fault**.

4 will takes place of 1 –> **1 Page Fault**

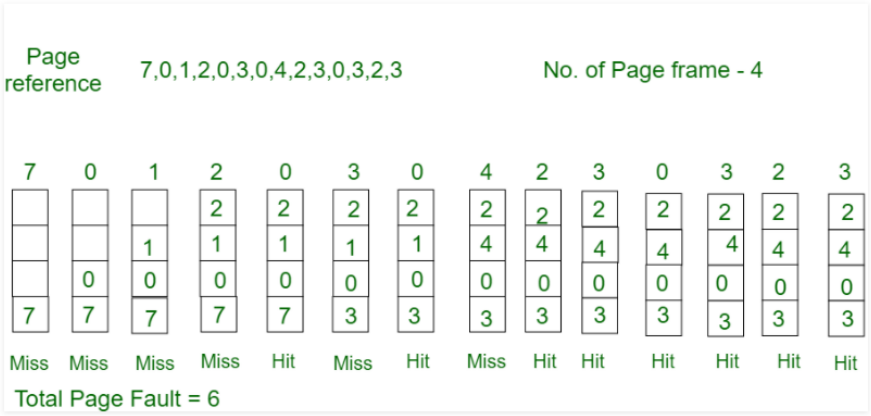
Now for the further page reference string –> **0 Page fault** because they are already available in the memory.

2. Optimal

2. Optimal Page replacement –

In this algorithm, pages are replaced which would not be used for the longest duration of time in the future.

Example-2: Consider the page references 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, with 4 page frame. Find number of page fault.



Initially all slots are empty, so when 7 0 1 2 are allocated to the empty slots –> **4 Page faults**

0 is already there so –> **0 Page fault.**

when 3 came it will take the place of 7 because it is not used for the longest duration of time in the future.–>**1 Page fault.**

0 is already there so –> **0 Page fault.**.

4 will takes place of 1 –> **1 Page Fault.**

Now for the further page reference string –> **0 Page fault** because they are already available in the memory.

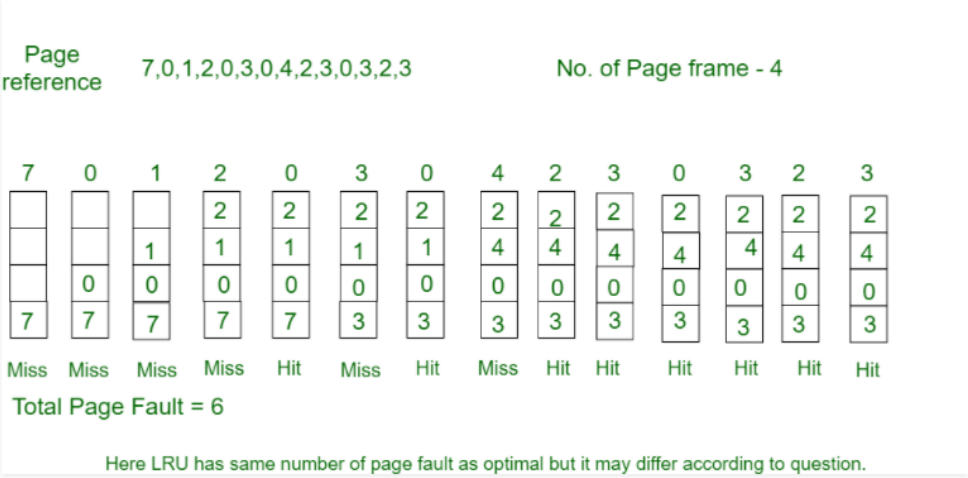
Optimal page replacement is perfect, but not possible in practice as the operating system cannot know future requests. The use of Optimal Page replacement is to set up a benchmark so that other replacement algorithms can be analyzed against it.

7. Explain LRU page replacement algorithm with the reference string 7,0,1,2,0,3,0,4, 2,3,0,3,2 and number of frames is 4.

3. Least Recently Used –

In this algorithm page will be replaced which is least recently used.

Example-3 Consider the page reference string 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2 with 4 page frames.Find number of page faults.



Initially all slots are empty, so when 7 0 1 2 are allocated to the empty slots –> **4 Page faults**

0 is already their so –> **0 Page fault.**

when 3 came it will take the place of 7 because it is least recently used –>**1 Page fault**

0 is already in memory so –> **0 Page fault**.

4 will takes place of 1 –> **1 Page Fault**

Now for the further page reference string –> **0 Page fault** because they are already available in the memory.

8. Explain briefly file or (space) allocation methods in file.

The allocation methods define how the files are stored in the disk blocks. There are three main disk space or file allocation methods.

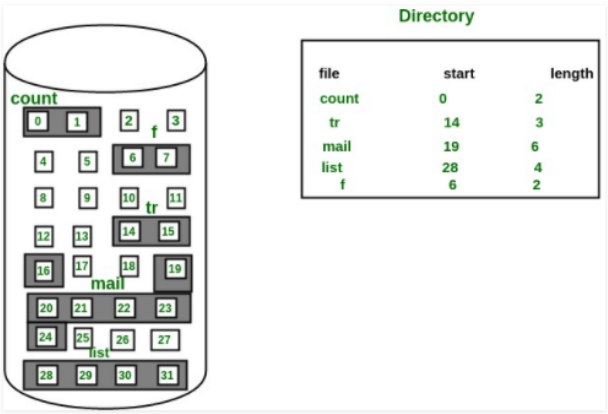
- Contiguous Allocation
- Linked Allocation
- Indexed Allocation

1. Contiguous Allocation

In this scheme, each file occupies a contiguous set of blocks on the disk. For example, if a file requires n blocks and is given a block b as the starting location, then the blocks assigned to the file will be: *b, b+1, b+2,.....b+n-1.* This means that given the starting block address and the length of the file (in terms of blocks required), we can determine the blocks occupied by the file.
The directory entry for a file with contiguous allocation contains

- Address of starting block
- Length of the allocated portion.

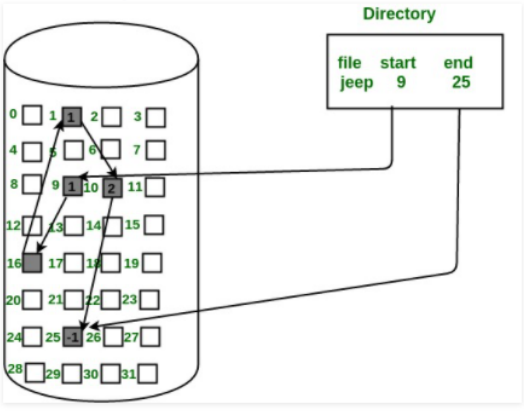
The *file 'mail'* in the following figure starts from the block 19 with length = 6 blocks. Therefore, it occupies *19, 20, 21, 22, 23, 24* blocks.



2. Linked List Allocation

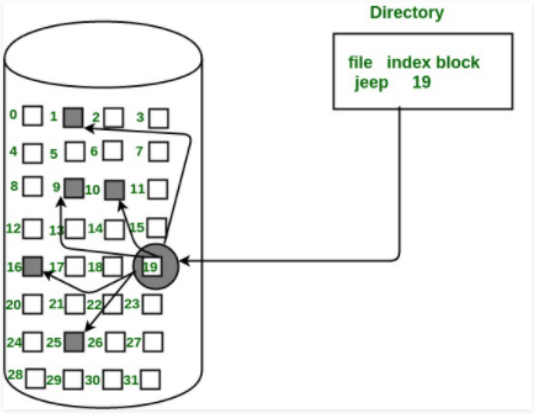
In this scheme, each file is a linked list of disk blocks which **need not be** contiguous. The disk blocks can be scattered anywhere on the disk.
The directory entry contains a pointer to the starting and the ending file block. Each block contains a pointer to the next block occupied by the file.

The file 'jeep' in following image shows how the blocks are randomly distributed. The last block (25) contains -1 indicating a null pointer and does not point to any other block.



3. Indexed Allocation

In this scheme, a special block known as the **Index block** contains the pointers to all the blocks occupied by a file. Each file has its own index block. The ith entry in the index block contains the disk address of the ith file block. The directory entry contains the address of the index block as shown in the image:



9. What is segmentation explain its advantages.

Segmentation in Operating System

Difficulty Level : Medium • Last Updated : 21 Sep, 2021

A process is divided into Segments. The chunks that a program is divided into which are not necessarily all of the same sizes are called segments. Segmentation gives user's view of the process which paging does not give. Here the user's view is mapped to physical memory.
There are types of segmentation:

1. **Virtual memory segmentation –**
   Each process is divided into a number of segments, not all of which are resident at any one point in time.
2. **Simple segmentation –**
   Each process is divided into a number of segments, all of which are loaded into memory at run time, though not necessarily contiguously.

Advantages of Segmentation

1. No internal fragmentation
2. Average Segment Size is larger than the actual page size.
3. Less overhead
4. It is easier to relocate segments than entire address space.
5. The segment table is of lesser size as compared to the page table in paging.

Unit-5

10. What is disc scheduling and disc scheduling algorithms.
FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK
Explain with example(any one can be given)

11. What is RAID? Explain raid structure briefly. What are different RAID levels? Explain them.

RAID (Redundant Arrays of Independent Disks)

Difficulty Level : Easy • Last Updated : 22 Sep, 2021

RAID, or "Redundant Arrays of Independent Disks" is a technique which makes use of a combination of multiple disks instead of using a single disk for increased performance, data redundancy or both. The term was coined by David Patterson, Garth A. Gibson, and Randy Katz at the University of California, Berkeley in 1987.

Why data redundancy?

Data redundancy, although taking up extra space, adds to disk reliability. This means, in case of disk failure, if the same data is also backed up onto another disk, we can retrieve the data and go on with the operation. On the other hand, if the data is spread across just multiple disks without the RAID technique, the loss of a single disk can affect the entire data.

12. What is system security. Discuss program threats , system & network threats in detail.

System security

A system is said to be secure if its resources are used and accessed as intended under all the circumstances, but no system can guarantee absolute security from several of the various malicious threats and unauthorized access.

Program Threats

Operating system's processes and kernel do the designated task as instructed. If a user program made these process do malicious tasks, then it is known as **Program Threats**. One of the common example of program threat is a program installed in a computer which can store and send user credentials via network to some hacker. Following is the list of some well-known program threats.

- **Trojan Horse** − Such program traps user login credentials and stores them to send to malicious user who can later on login to computer and can access system resources.
- **Trap Door** − If a program which is designed to work as required, have a security hole in its code and perform illegal action without knowledge of user then it is called to have a trap door.
- **Logic Bomb** − Logic bomb is a situation when a program misbehaves only when certain conditions met otherwise it works as a genuine program. It is harder to detect.
- **Virus** − Virus as name suggest can replicate themselves on computer system. They are highly dangerous and can modify/delete user files, crash systems. A virus is generatlly a small code embedded in a program. As user accesses the program, the virus starts getting embedded in other files/ programs and can make system unusable for user

System Threats

System threats refers to misuse of system services and network connections to put user in trouble. System threats can be used to launch program threats on a complete network called as program attack. System threats creates such an environment that operating system resources/ user files are misused. Following is the list of some well-known system threats.

- **Worm** − Worm is a process which can choked down a system performance by using system resources to extreme levels. A Worm process generates its multiple copies where each copy uses system resources, prevents all other processes to get required resources. Worms processes can even shut down an entire network.
- **Port Scanning** − Port scanning is a mechanism or means by which a hacker can detects system vulnerabilities to make an attack on the system.
- **Denial of Service** − Denial of service attacks normally prevents user to make legitimate use of the system. For example, a user may not be able to use internet if denial of service attacks browser's content settings.

Network Threats

1. Phishing

This type of online fraud is designed to steal sensitive information, such as credit card numbers and passwords. Phishing attacks impersonate reputable banking institutions, websites, and personal contacts, which come in the form of immediate phishing e-mails or messages designed to look legitimate.

Once you click the URL or reply to the messages, you are prompted to enter your financial details or use your credentials, which then sends your data to the malicious source.

2. Computer Viruses

These are pieces of software designed to spread from one computer device to another. Mostly they are downloaded from particular websites or sent as e-mail attachments with the intent of infecting your computer as well as other computers on your contact list through systems on your network. They can disable your security settings, send spam, steal and corrupt data from your computer, and even delete every single thing on your hard drive.

3. Malware/Ransomware

Malware is a malicious software mostly used by criminals to hold your system, steal your confidential data, or install damaging programs in your device without your knowledge. It spreads spyware, Trojans, and worms through pop-up ads, infected files, bogus websites, or e-mail messages.

On the other hand, ransomware is a type of malware where the cyber-criminals lock your device through a bad app or phishing e-mails then request a ransom to unlock the device. It can hinder you from running applications, encrypting your files, and even from completely using your device.

4. Rogue Security Software

This is malicious software that deceives users by making them believe that their security measures are not up-to-the-minute or their computer has a virus. They then offer to help you install or update the user's security settings by asking you to pay for a tool or download their program to help do away with the alleged viruses. This can lead to the installation of actual malware in your device.

13. What is system protection. Explain goals of system protection, principles and domain of protection.

Protection

- A mechanism that controls the access of programs, processes, or users to the resources defined by a computer system is referred to as protection. You may utilize protection as a tool for multi-programming operating systems, allowing multiple users to safely share a common logical namespace, including a directory or files.
- It needs the protection of computer resources like the software, memory, processor, etc. Users should take protective measures as a helper to multiprogramming OS so that multiple users may safely use a common logical namespace like a directory or data. Protection may be achieved by maintaining confidentiality, honesty and availability in the OS. It is critical to secure the device from unauthorized access, viruses, worms, and other malware.

Goals of Protection

- Operating system consists of a collection of objects, hardware or software. Each object has a unique name and can be accessed through a well-defined set of operations.
  Protection problem - ensure that each object is accessed correctly and only by those processes that are allowed to do so.

1. The policies define how processes access the computer system's resources, such as the CPU, memory, software, and even the operating system. It is the responsibility of both the operating system designer and the app programmer. Although, these policies are modified at any time.
2. Protection is a technique for protecting data and processes from harmful or intentional infiltration. It contains protection policies either established by itself, set by management or imposed individually by programmers to ensure that their programs are protected to the greatest extent possible.
3. It also provides a multiprogramming OS with the security that its users expect when sharing common space such as files or directories.

Principles of Protection

- Guiding principle – principle of least privilege
  - It dictates that programs, users, and even systems should be given just enough privileges to perform their tasks
  - An operating system following the principle of least privilege implements its features, programs, system calls, and data structures so that failure or compromise of a component does the minimum damage and allows the minimum damage to be done.

Domain of Protection

- To facilitate the protection, a process operates within a protection domain, which specifies the resources that the process may access.
- A protection domain specifies the resources that a process may access.
- Each domain defines a set of objects and the types of operations that may be invoked on each object. The ability to execute an operation on an object is an **access right**.
- A domain is a collection of access rights, each of which is an ordered pair <object-name, rights-set>.

For example, if domain D has the access right <file F, {read, write}>, then a process executing in domain D can both read and write file F; it cannot, however, perform any other operation on that object.