MODULE-1

1. ***What is operating system?**

   OS is a system software and is an interface between user and computer hardware.

2. ***What are the functions of an operating system?**

   Main functions:

   Processes management, Memory Management, Disk Management, I/O Management, File Management, Security and Protection.

3. **Latest versions of windows, Linux, Apple Mac.**

   Linux: Ubuntu 23.04, Windows: Windows 11, Apple: macOS Ventura 13.4, Android 13, iOS 16.5

4. ***System calls to create a process.**

   System call is an operating system function, it is used by programs to request services from the operating system.

   fork () (used to create a new process)
   -Types of System calls

   - Process control (allocate memory, create process, load and execute, end and abort)
   - File manipulation (read, write, open, close, create, delete files)
   - Device Management (request and release devices)
   - Information maintenance (get time, date, system data, get and set processes)
   - Communications (create and delete communication connections, send and receive messages)

5. ***What are the returning values of fork () system call and why is it used?**

   - In the parent process, \`fork ()\` returns the process ID (PID) of the child process.
   - In the child process, \`fork ()\` returns 0, indicating that it is the child process.

   However, it's important to note that \`fork()\` is a low-level operating system call, and the exact return values may vary depending on the programming language or platform being used.

6. ***Different types of OS?**
   Operating systems can be classified into various types, such as mobile, desktop,server, real-time, embedded, multi-user, distributed, and virtualization operating systems, each catering to specific device or system requirements.

7. **What is the difference between fork () and exec()?**

The `fork ()` system call is used to create a new process by duplicating the existing process, while the `exec()` system call is used to replace the current process with a new executable program, effectively changing the code and memory space of the process.

8. ***What are the different structures of an Kernel/Operating system?**

There are several different structures or architectural models that can be used to design an operating system, including monolithic, layered, microkernel, modular, and hybrid architectures.

9. **Which structure is used by Linux and windows?**

Windows predominantly follows a hybrid architecture, combining elements of both monolithic and microkernel designs, while Linux is primarily based on a monolithic architecture.

10. **Difference between MBR and UEFI?**

MBR (Master Boot Record) is an older legacy partitioning scheme, while UEFI (Unified Extensible Firmware Interface) is a modern firmware interface that supports larger disk sizes and offers more advanced features.

11. **What is bootstrap?**

To start a computer by loading its OPERATING SYSTEM from disk storage into memory.

12. **Briefly explain about the system booting process.**

The system boot process is the sequence of events that occurs when a computer is powered on, including hardware initialization, loading the operating system, and launching system services and applications.

13. ***Shell script**

-> Bash – born again shell.
-> $ implies waiting for command in the command-line
-> # implies shell is running as root

-> # ! /bin/bash is called shebang statement

-> rwx – read write execute permission for a file

-> chmod u+x helloworld.sh - changing the mode and giving user the permission to execute.

10. ***Why is shell scripting used?**

Shell scripts are used for automating tasks, executing commands, system administration, customization, batch processing, task scheduling, rapid prototyping, and cross-platform compatibility.

11. ***What is kernel?**

The kernel is the core component of an operating system that manages system resources, provides essential services, and acts as an interface between software applications and the hardware of a computer system. It handles tasks such as memory management, process scheduling, device drivers, and system call handling, ensuring the proper functioning of the operating system.

14. ***What are the two modes of operating system?**

User mode - user application is allowed to execute.
Kernel mode - used to execute main operating system functions, which are notdirectly accessible for the normal application but can be accessed through system call.

15. **What is the latest Linux kernel version? - 6.3.8**

16. ***Difference between multiprogramming and multitasking?**

Multiprogramming refers to the capability of a system to simultaneously execute multiple programs by dividing the CPU time into fixed-size time slots, while multitasking refers to the ability of an operating system to execute multiple tasks or processes concurrently by rapidly switching the CPU between them.

17. ***What is the degree of multiprogramming?**

The degree of multiprogramming refers to the number of programs or processes that are concurrently present in the main memory of a computer system, ready to execute and compete for CPU time.

18. **What is WSL?**

WSL (Windows Subsystem for Linux) is a compatibility layer that enables running native Linux command-line tools and applications directly on Windows 10 or Windows Server.

19. ***What is shell?**

A shell is a command-line interface or program that acts as a user interface to interact with an operating system, allowing users to enter commands and execute them to perform various tasks such as running programs, managing files, and controlling system configurations.

20. *How do you execute a shell script?

- Open the terminal.
- Go to the directory where you want to create your script.
- Create a file with .sh extension.
- Write the script in the file using an editor.
- Make the script executable with command chmod +x <filename>.
- Run the script using ./<filename>

21. *Differentiate between shell script and c program:

Shell Script:

- Used for automating tasks and interacting with the command line.
- Interpreted language. Easier syntax, primarily focused on command-line operations.
- Slower performance compared to compiled languages.
- Highly portable between Unix-like systems.
- Suited for quick tasks and system administration.

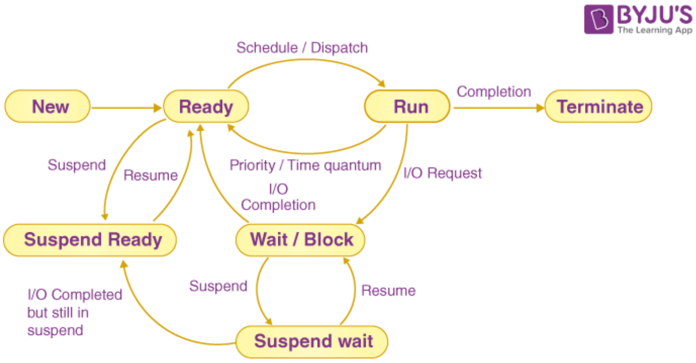
C Program:

- Used for developing applications, system software, and low-level tasks.
- Compiled language.
- More complex syntax, provides low-level control.
- Better performance due to compilation into machine code.
- Might require adjustments for different platforms.
- Suited for performance-critical tasks and system programming

MODULE-2

1. ***What are the different process states?**

The different process states in an operating system are typically represented as "running," where the process is currently executing instructions; "waiting," where the process is waiting for an event or resource to become available; "blocked," where the process is unable to proceed until a specific condition is met; and "terminated," where the process has finished execution and is no longer active. **(Fig important)**



Process State Diagram

2. ***What is starvation and convoy effect?**

Starvation refers to a situation where a process is unable to access a resource indefinitely due to being continuously overshadowed by other processes.
Starvation occurs in SJF and priority scheduling.
The convoy effect describes the inefficiency caused by a long process queue forming behind a resource, even if other resources are available.
Convoy effect in FCFS.

3. ***Thread based system calls (thread creation)**

pthread.h: This is the header file that provides the necessary functions and data types for creating and managing threads.

pthread_create(): This function creates a new thread and takes several parameters, including a pointer to a thread identifier, attributes for the new thread (usually set to NULL for default attributes), a function to be executed by the thread, and an argument to pass to the thread function.

pthread_join(): This function suspends the execution of the calling thread until the target thread terminates, allowing proper synchronization and ensuring that the main thread waits for all threads to complete before exiting.

pthread_exit(): This function terminates the calling thread and returns the exit status. It is typically used to exit from a thread instead of returning from the thread function.

4. ***Inter process communication.**

Inter-process communication (IPC) refers to the methods and mechanisms through which processes in an operating system can exchange data, synchronize their actions, and communicate with each other. IPC allows processes to cooperate, share resources, and coordinate their activities to accomplish tasks collectively. There are several types of IPC techniques, including:

1. Shared Memory: Processes can share a common memory region, allowing them to read and write data directly into that shared space, enabling fast communication.

shmget - is used to create the shared memory segment.
shmat - is used to attach the shared segment with the address space of the process.

2. Message Passing: Processes exchange messages via various methods such as pipes, named pipes (FIFOs), message queues, or sockets, where data is sent and received in a serialized manner.

pipe() - connect two process (inter-process communication)

5. Header file <sys/stat.h>

struct stat or stat () - it contains all the details of a file. (read write permissions, last modified, which type of file)

6. *What is Process Control Block (PCB)?

The Process Control Block (PCB) is a data structure used by the operating system to store and manage information about a specific process, including its process ID, program counter, register values, scheduling information, and resource allocation details, allowing the operating system to effectively control and switch between processes.

7. *Different scheduling algorithm briefly explain. (Basic working)

1. First-Come, First-Served (FCFS): Processes are executed in the order they arrive, with the CPU allocated to the first process in the ready queue until completion.
2. Shortest Job Next (SJN): The process with the smallest total execution time is selected next, minimizing the waiting time and providing optimal scheduling for processes.
3. Round Robin (RR): Each process is given a fixed time slice (quantum) to execute, and if it doesn't complete within the time slice, it is moved to the end of the queue to provide fair execution for all processes.
4. Priority Scheduling: Processes are assigned priorities, and the CPU is allocated to the highest priority process, ensuring that high-priority tasks are completed first.
5. Shortest Remaining Time (SRT): Similar to SJN, but the CPU can be preempted by a new process with a shorter burst time, resulting in faster execution for shorter processes.

8. What are the criterions of the scheduling algorithm

Scheduling algorithms can be evaluated based on criteria such as CPU utilization, throughput, turnaround time, waiting time, response time, and fairness in order to assess their efficiency and effectiveness in managing process scheduling.

9. ***What is aging?**

Aging, in the context of scheduling algorithms, is a technique that gradually increases the priority or preference of a process over time as it waits in a queue. By increasing

the priority of waiting processes, aging helps prevent long-waiting processes from being starved and ensures fairness in resource allocation.

10. ***What is preemptive and non-preemptive CPU scheduling? Explain itwith examples.**

Preemptive CPU scheduling involves the ability of the operating system to interrupt a running process and allocate the CPU to another process with higher priority, such as in Round Robin or Shortest Remaining Time (SRT) scheduling.
non-preemptive CPU scheduling allows a running process to continue until completion or voluntarily yielding the CPU, like in First-Come, First-Served (FCFS) or Shortest Job Next (SJN) scheduling, where processes cannot be interrupted during execution.

11. **Different schedulers: LTS, STS, MTS.**

LTS :-(Long term scheduler) Bring maximum number of resources to ready state. STS:-(Short term scheduler) Bring process from ready state to running state.
MTS:-(Medium term scheduler) Bring processes from waiting /block state to ready /suspend state.
Frequency: STS>MTS>LTS

12. ***Difference between thread and process.**

A process is an instance of a running program that has its own memory space and resources, while a thread is a lightweight unit of execution within a process that shares the same memory space and resources with other threads of the same process.
When u run a c program, normally there will be one process and one thread (unless u are using fork or thread calls)

13. **name of the process with pid 0 and pid 1 in Linux?**

In Linux, the process with PID 0 is known as the "swapper" or "scheduler" process, and it is responsible for managing system resources and creating other processes. The process with PID 1 is called "init" or "systemd" and is the parent process for all other processes in the system, serving as the root of the process hierarchy.

14. ***What are the system calls to get process id and its parent?**

In Linux, the system call `getpid()` can be used to retrieve the process ID of the current process, while the system call `getppid()` can be used to obtain the process IDof the parent process.

15. ***Commands to list kernel version, os version, name etc.**

To display the kernel version:

```
uname -r
```

To display the operating system name:

```
uname -s
```

To display additional system information, such as the kernel name, network node hostname, kernel release, and processor architecture:
uname –a

16. Linux command to list the no of cores and threads in Linux?
lscpu

17. *What are the permissions for file in Linux?

In Linux, file permissions are represented by a sequence of ten characters, where the first character denotes the file type, and the subsequent characters indicate the read, write, and execute permissions for the owner, group, and other users. The symbols "r" (read), "w" (write), "x" (execute), and "-" (absence of permission) are used to represent the different levels of permissions.

18. What does if a file has permission 644 mean?

File Owner = 6 (read and
write)
Group = 4 (read)
World/Public/Others = 4 (read

19. *777 means which all permissions?

In Linux, the file permission mode "777" means that the file has full permissions for the owner, group, and other users. Specifically, it represents read (r), write (w), and execute (x) permissions for all entities, allowing anyone to read from, write to, and execute the file.
Read = 4, write=2, execute=1. So full permission for
user = 7 File Owner=7, Group =7 and outside World=7
(ie. 777)

20. Which cpu scheduling algorithms are used in windows, Linux and Apple macOS?

- Windows: Windows uses a priority-based scheduling algorithm that considers factors like thread and process priority to determine which process runs next.
- Linux: Linux employs the Completely Fair Scheduler (CFS), aiming for fair CPU time distribution among processes based on their priority and weight.
- macOS: macOS historically used a round-robin scheduling algorithm with multilevel feedback queues, but Apple's specific algorithm details might not be as widely documented.

MODULE-3

1. *What is a race condition?

- A race condition occurs when multiple processes or threads access shared resources or variables simultaneously, leading to unpredictable and undesired behavior.

2. *Explain the concept of a critical section.

- A critical section is a part of a program where shared resources are accessed, and only one process or thread should be allowed to execute it at a time to maintain data consistency and avoid race conditions.

3. *What is process synchronization, and why is it necessary?

- Process synchronization refers to the coordination of multiple processes or threads to ensure orderly and predictable execution. It is necessary to prevent race conditions, maintain data integrity, and enforce mutual exclusion when accessing shared resources.

4. *What is a semaphore, and what are its types?

- A semaphore is a synchronization mechanism that controls access to shared resources. Its types include binary semaphore (mutex) and counting semaphore.

A semaphore is variable used for process synchronization. Types of semaphores? binary and counting

Header file <semaphore.h>

sem_init - it is used to initialize the semaphore.
sem_ wait - It is used to decrement thevalue. (wait)
sem_post - It is used to increment the value.
(signal)

5. *How do monitors help with synchronization in concurrent programming?

-Monitors are high-level synchronization constructs that encapsulate shared data and provide synchronization methods to ensure that only one thread can execute a monitor procedure at a time, enabling safe access to shared resources.

6. *Discuss the three classical synchronization problems and their solutions.

- The three classical synchronization problems are the Producer-Consumer problem (solved using bounded buffer and synchronization mechanisms), the Reader-Writer problem (solved using reader/writer locks), and the Dining Philosophers problem (solved using resource allocation algorithms like the "one chopstick at a time" approach).

****SOLUTION for each problem (Imp)**

7. *Explain the necessary conditions for deadlock **to occur.**

- Deadlock occurs when four necessary conditions are satisfied: mutual exclusion, hold and wait, no preemption, and circular wait.

8. *What are some techniques for preventing deadlock?

- Mutual exclusion : Some resources such as read only files shouldn't be mutually exclusive. They should be sharable. But some resources such as printers must be mutually exclusive.
- Hold and wait : To avoid this condition we have to ensure that if a process is requesting for a resource it should not hold any resources.
- No preemption : If a process is holding some resources and requestsanother resource that cannot be immediately allocated to it (that is the process must wait), then all the resources currently being held are preempted(released autonomously).
- Circular wait : the way to ensure that this condition never holds is to impose a total ordering of all the resource types, and to require that each process requests resources in an increasing order of enumeration.

9. *Describe the recovery phase in dealing with deadlocks.

-In the recovery phase, the operating system can take actions like killing processes, preemption, or rolling back to a previous state to break the deadlock and resume normal operation.

10. *How can deadlocks be avoided in a system?

- A dead lock avoidance algorithm dynamically examines the resource-allocation state to ensure that a circular wait condition can never exist. The resource allocation state is defined by the number of available and allocated resources, and the maximum demand of the process.There are two algorithms:
- 1. Resource allocation graph algorithm
- 2. Banker's algorithm

  a. Safety algorithm

  b. Resource request algorithm

11. *Explain the Banker's algorithm for resource allocation.

- The Banker's algorithm is a resource allocation algorithm that ensures safe allocation by considering the current resource allocation, maximum resource requirements, and future resource requests to determine if granting a request will result in a safe state.

Bankers Algorithm (Safety Algol + Resource Request Algol)

Safe Sequence is sequence of resources available to allocate, which after being allocated will assure that system will not go into deadlock.

12. Discuss the request and safety algorithms used in deadlock avoidance.

- The request algorithm checks if granting a resource request will lead to a safe state, while the safety algorithm checks if a particular resource allocation state is safe, meaning no deadlock will occur.

13. *What is the purpose of the P (wait) and V (signal) operations in semaphore handling?

- The P (wait) operation **decreases the semaphore value** and waits if it becomes zero, while the V (signal) operation **increases the semaphore value** and wakes up any waiting process if necessary, allowing synchronization and coordination between processes accessing shared resources.

14. *How is deadlock detected?

Deadlock is detected by analyzing the resource **allocation graph** or state to identify the presence of a **cycle**, indicating a potential deadlock situation.

15. What does bounder buffer means?

A bounded buffer refers to a fixed size buffer that can hold a limited number of data items before additional items must wait until space becomes available.

16. *Write some real-world examples for deadlock?

1. Traffic Deadlock: Imagine a four-way intersection with traffic lights where each road has equal priority. If all the roads have vehicles waiting to proceed but are unable to move due to the intersection being blocked, a deadlock can occur.

2. Resource Deadlock in Supply Chains: In complex supply chains involving multiple suppliers, manufacturers, and distributors, deadlocks can occur if each entity is waiting for resources or materials from another entity, leading to delays and a standstill in the overall process.

3. Banking Deadlock: In banking systems, if two or more customers are involved in a circular dependency where each customer holds an account that the other customer requires to complete a transaction, a deadlock can arise if neither customeris willing to release their account.

17. What is resource allocation graph? What does cycle in resource allocation graph means?

A resource allocation graph is a graphical representation showing the relationships between processes and resources in a system. A cycle in the graph indicates a potential deadlock, where processes are waiting for resources held by other processes,

leading to a system-wide standstill. Resolving the cycle is necessary to avoid or resolve the deadlock situation.

18. *Data Structures used to implement Banker’s Algorithm

- Available: It is a 1-D array that tells the number of each resource type (instance of resource type) currently available.
- Max: It is a 2-D array that tells the maximum number of each resource type required by a process for successful execution.
- Allocation: It is a 2-D array that tells the number of types of each resource type that has been allocated to the process.
- Need: It is a 2-D array that tells the number of remaining instances of each resource type required for execution.

19. *Critical section code structure:

```
do {
    entry section
            critical section
    exit section
            remainder section
  } while (TRUE);
```

20.*Conditions for good solution for critical section:

- *Mutual Exclusion - implies that only one process can be inside the critical section at any time. If any other processes require the critical section, they must wait until it is free.
- **Progress** - means that if a process is not using the critical section, then it should not stop any other process from accessing it. In other words, any process can enter a critical section if it is free.
- **Bounded Waiting** - means that each process must have a limited waiting time. It should not wait endlessly to access the critical section.

21. *Solutions for critical section:

Some of the software-based solutions for critical section problems are Peterson's solution, semaphores, monitors. Some of the hardware-based solutions for the critical section problem involve atomic instructions such as TestAndSet, compare and swap, Unlock and Lock.

MODULE-4

1. *What is a physical address?

A: A physical address refers to the actual location of data in the physical memory (RAM) of a computer.

2. *What is a logical address?

A: A logical address is the address generated by the CPU and seen by the programs running on a computer. It is also known as a virtual address.

3. *What is the difference between contiguous and non-contiguous memory allocation?

A: Contiguous memory allocation refers to the allocation of consecutive blocks of memory to a process, while non-contiguous memory allocation allows a process to be scattered in different areas of memory.

4. *What is paging in operating systems?

A: Paging is a memory management scheme that allows memory to be divided into **fixed-size** blocks called pages. These pages are used to store and manage processes in the system.

5. *What is segmentation in operating systems?

A: Segmentation is a memory management technique where memory is divided **into variable-sized** segments. Each segment represents a logical unit such as code, data, stack, etc.

6. *What is demand paging?

A: Demand paging is a memory management technique where pages are loaded intomemory only when they are demanded by the running program, rather than loading the entireprogram into memory at once.

7. *What is virtual memory?

A: Virtual memory is a technique that allows the execution of processes that are larger than the physical memory. It uses disk space as an extension of physical memory to store data that is not currently in use.

8. *What is a TLB (Translation Lookaside Buffer)?

A: TLB is a cache-like memory structure that stores recently accessed page table entries. It is used to accelerate the translation of virtual addresses to physical addresses.

9. *What is page replacement?

A: Page replacement is a process where an operating system selects a page in memory to be replaced when a new page needs to be brought in. It is used when there is no free memory available for a new page.

Page Replacement Algorithms: FIFO, LRU and Optimal

Page Hit vs Fault

Bélády's anomaly is the name given to the phenomenon where increasing the number of page frames results in an increase in the number of page faults for a given memory access pattern.

10. *What is fragmentation in memory management?

A: Fragmentation refers to the inefficient use of memory space, where memory is divided into small, non-contiguous chunks, making it difficult to allocate larger blocks of memory.

11. *What is compaction in memory management?

A: Compaction is a memory management technique used to reduce external fragmentation. It involves moving the allocated memory blocks to eliminate the gaps betweenthem, thereby creating larger contiguous blocks.

12. *What is the purpose of a page table in virtual memory?

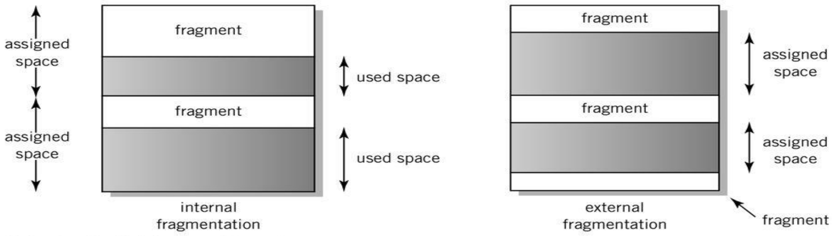
A: The page table is a data structure used by the operating system to map virtual addresses to physical addresses. It helps in translating virtual memory references to actual physical memory locations.

13. *How does a Translation Lookaside Buffer (TLB) improve memory access speed?

A: The TLB stores recently accessed page table entries, which allows for faster translation of virtual addresses to physical addresses. By caching these translations, the TLB reduces the number of memory accesses required for address translation, resulting in improved memory access speed.

15. *What is the difference between internal fragmentation and external fragmentation?

A: Internal fragmentation occurs when memory is allocated in fixed-size blocks, and each block may have some unused space. External fragmentation, on the other hand, refers to the existence of unused memory blocks scattered throughout the memory space, making it challenging to allocate contiguous memory to new processes.



Englander: The Architecture of Computer Hardware and Systems Software, 2nd edition Chapter 16, Figure 16-13

Figure 16.13 Internal and external fragmentation

Compaction or De-Fragmentation (Solution to Fragmentation Problem)

Compaction is a technique to collect all the free memory present in form of fragments into one large chunk of free memory, which can be used to run other processes.

It does that by moving all the processes towards one end of the memory and all the available free space towards the other end of the memory so that it becomes contiguous.

16. *What is the role of a page replacement algorithm in virtual memory management?

A: A page replacement algorithm is responsible for selecting which page in memory should be replaced when a new page needs to be brought in. It aims to minimize the number of page faults and optimize memory utilization by considering factors like page accesspatterns and page usage frequency.

17. **What are the advantages of using segmentation over paging?

A: Segmentation allows for flexible memory allocation as segments can vary in size, whereas paging requires fixed-size blocks. Segmentation also provides better protectionbetween segments, as each segment can have its own access permissions. However, segmentation may result in external fragmentation.

18. How does a TLB (Translation Lookaside Buffer) reduce memory access time?

A: The TLB stores recently accessed page table entries, which eliminates the need to access the page table in memory for every address translation. By caching these translations, the TLB speeds up the memory access process and reduces the overall access time.

19. What is the role of a page table entry in paging?

A: A page table entry contains information about a specific page in virtual memory, such as the corresponding physical page frame number, page permissions, and status bits (e.g., dirty bit, valid bit). It is used by the operating system to manage and translate virtualaddresses to physical addresses.

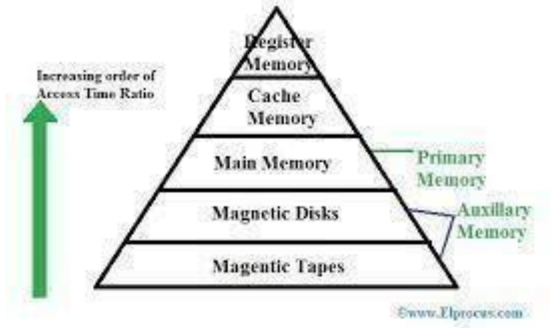
20. How does segmentation differ from paging in terms of memory allocation?

A: Segmentation allows memory to be divided into variable-sized logical units called segments, whereas paging divides memory into fixed-sized blocks called pages. Segmentation provides flexibility in memory allocation, but it may lead to external fragmentation. Paging offers better memory utilization but may result in internal fragmentation.

21. *Describe the difference between fixed partitioning and variable partitioning in contiguous memory allocation.

Fixed partitioning divides memory into fixed-sized partitions, while variable partitioning allocates memory based on the size requirements of each process, resulting in more efficient memory utilization.

22. Memory Hierarchy?



***How is virtual memory implemented in OS?**

Virtual memory usually resides in Hard-disk (C: drive, where Windows OS is installed)

Virtual memory in an operating system is typically implemented using a combination of hardware and software mechanisms, such as page tables and swapping, to manage the mapping of virtual addresses to physical memory.

23. How to change the size of virtual memory in Windows OS?

To change the size of virtual memory in Windows OS, you can adjust the paging file settings by going to the System Properties, selecting the Advanced tab, and clicking on the "Settings"

button under the Performance section to access the Virtual Memory settings.

24. Page size for windows, Linux and Apple macOS?

- in OS X and in earlier versions of iOS, the size of a page is 4 kilobytes. In later versions of iOS, A7- and A8-based systems expose 16-kilobyte pages to the 64-bit user space backed by 4-kilobyte physical pages, while A9 systems expose 16-kilobyte pages backed by 16-kilobyte physical pages.
- On most Windows 10 systems with 8 GB of RAM or more, the OS manages the size of the paging file nicely. The paging file is typically 1.25 GB on 8 GB systems, 2.5 GB on 16 GB systems and 5 GB on 32 GB systems
- The basic unit of memory is the page. Nobody knows how large a page is (that is why the Linux swap space structure, with a signature at the end of a page, is so unfortunate), this is architecture-dependent, but typically PAGE_SIZE = 4096.

MODULE-5

1. ***Name and describe the file systems commonly used in Linux, Windows,macOS, Android, and removable storage devices.**

***File Systems in Operating System**

A file system is a method an operating system uses to store, organize, and manage files and directories on a storage device.

Linux/Ubuntu commonly uses file systems such as ext4, ext3, and XFS.

Windows uses file systems like **NTFS** and FAT32.

Apple Mac OS utilizes the HFS+ (Mac OS Extended) and **APFS** (Apple File System).

Android uses the YAFFS (Yet Another Flash File System) and **F2FS** (Flash-Friendly

File System).

Pen Drives/Memory Cards often use FAT32 or **exFAT.**

2. ***What are the file allocation methods?**

Contiguous Allocation:

Sequential allocation

Files are allocated contiguously on disk, meaning they occupy consecutive blocks of disk space. It provides fast access but suffers from fragmentation issues.

Non-Contiguous Allocation:

Linked Allocation: Files are allocated using linked lists, where each block contains a pointer to the next block. It avoids fragmentation but can have slower access times due to traversing the linked list.

Indexed Allocation: Each file has an index block that contains pointers to all the blocks it occupies on the disk. It provides faster access than linked allocation but requires additional overhead for maintaining the index structure.

3. ***Compare LOOK and CLOOK disk scheduling algorithms in terms of efficiencyand performance.**

Both LOOK and CLOOK are variants of the SCAN disk scheduling algorithm.LOOK scans only in the direction of the requests, while CLOOK scans only within a limited range. CLOOK typically provides better performance by reducing arm movement and seeks time compared to LOOK.

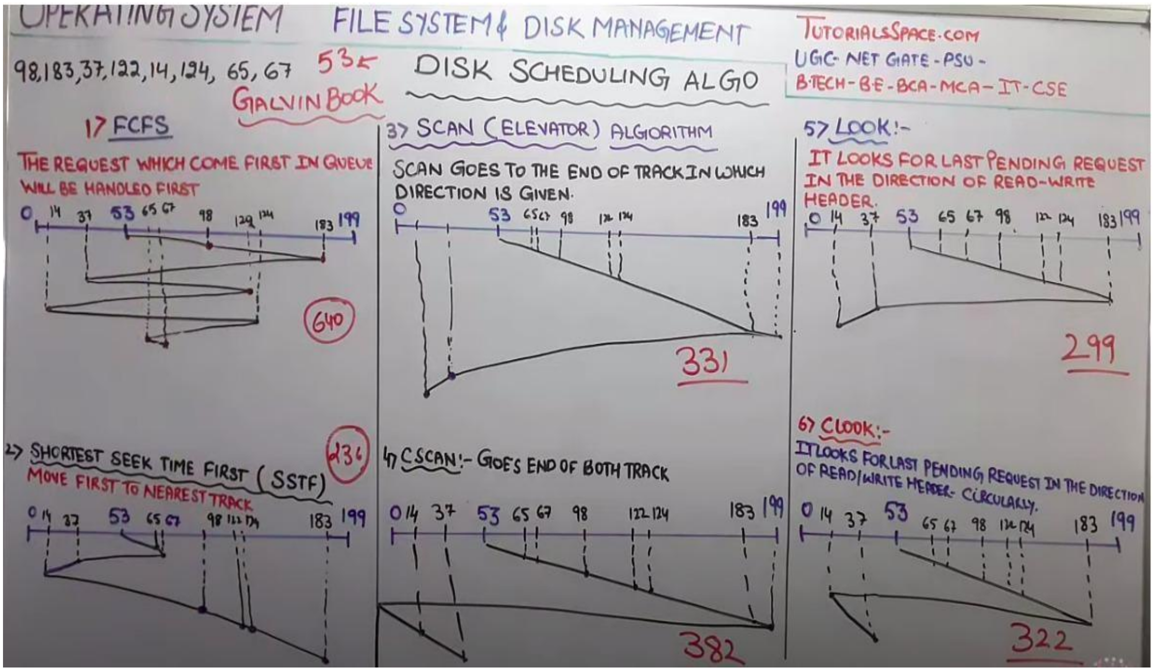
4. ***Discuss the differences between SCAN and CSCAN disk scheduling algorithms.**

The SCAN disk scheduling algorithm scans the disk in a single direction, servicing requests along the way, while the CSCAN algorithm also scans in a single direction but jumps to the other end of the disk once it reaches the end, ensuring no requests are

left unserved

5. ***Compare the characteristics and behavior of LOOK and SCAN disk scheduling algorithms.**

Both LOOK and SCAN aim to optimize disk access and reduce seek time. However, LOOK is more efficient in terms of arm movement as it scans only in the direction of the requests, while SCAN ensures fairness by scanning the entire disk surface.


OPERATING SYSTEM
FILE SYSTEM & DISK MANAGEMENT
TUTORIALSSPACE.COM
UGC- NET GATE - PSU -
B.TECH - B.E - BCA - MCA - IT - CSE
98,183,37,122,14,124, 65, 67
53
GALVIN BOOK
DISK SCHEDULING ALGO
1> FCFS
THE REQUEST WHICH COME FIRST IN QUEUE WILL BE HANDLED FIRST
0 14 37 53 65 67 98 122 124 183 199
640
3> SCAN (ELEVATOR) ALGORITHM
SCAN GOES TO THE END OF TRACK IN WHICH DIRECTION IS GIVEN.
0 53 65 67 98 122 124 183 199
331
5> LOOK:-
IT LOOKS FOR LAST PENDING REQUEST IN THE DIRECTION OF READ-WRITE HEADER.
0 14 37 53 65 67 98 122 124 183 199
299
2> SHORTEST SEEK TIME FIRST (SSTF)
MOVE FIRST TO NEAREST TRACK
0 14 37 53 65 67 98 122 124 183 199
236
4> CSCAN:- GOES END OF BOTH TRACK
0 14 37 53 65 67 98 122 124 183 199
382
6> CLOOK:-
IT LOOKS FOR LAST PENDING REQUEST IN THE DIRECTION OF READ/WRITE HEADER- CIRCULARLY.
0 14 37 53 65 67 98 122 124 183 199
322

6. ***What is an inode in a file system?**
   An inode (index node) is a data structure used by file systems to store metadata about a file, such as file permissions, ownership, size, timestamps, and pointers to the actual data blocks.
7. **Explain the role of inodes in file management and metadata storage.**
   Inodes play a crucial role in file management by providing a reference to the physical location of the file's data blocks and storing essential metadata associated with the file.

8. ***What are file permissions in an operating system?**
   File permissions are access rights assigned to files that determine which users or groups can read, write, or execute the file.
9. ***What are the key protection functions provided by an operating system?**
   The key protection functions provided by an operating system include access control, which ensures authorized resource access, and user authentication, which verifies the identity of users accessing the system. These functions help enforce security policies, maintain data integrity, and protect against unauthorized access or misuse of system resources.
10. ***Different directory structures?**
    There are several different directory structures used in operating systems. Here are some common ones:
    1. **Single-Level Directory**: This is the simplest directory structure where all files are stored in a single directory without any subdirectories.
    2. **Two-Level Directory:** In this structure, each user has their own directory, and files are stored within those individual directories.
    3. **Tree-Structured Directory:** Similar to the hierarchical directory structure, this structure allows for multiple levels of directories organized in a tree-like structure.
11. **Directory structure used in the present OS like windows or Linux or android or Mac?**
    Windows uses a hierarchical directory structure, while Linux, macOS, and Android also follow a similar hierarchical directory structure with a root directory and various

subdirectories.

12. ***Different File access methods with examples?**
There are different file access methods, including:

1. **Sequential Access:** Reading or writing data in a sequential manner from the beginning to the end of a file, such as with tape drives.

2. **Random Access:** Directly accessing specific parts of a file using byte offsets, allowing for efficient access to any part of the file, such as with hard disk drives.

3. **Indexed Access:** Accessing a file through an index or lookup table thatcontains pointers to specific file locations, enabling quick and direct access to desired data, such as with indexed file allocation methods.

13. ***Protection matrix vs Access control list vs Capabilities (OS protection Mechanism)**

Protection Matrix: A protection matrix is a table-based representation that lists subjects (users, processes) and objects (files, resources) with their associated access rights, providing an overview of the permissions and access control in a system.

Access Control Lists (ACL): ACL is a method of defining permissions on objects (files, directories) by associating a list of users or groups with specific access rights, allowing for more granular control over access permissions compared to the traditional owner-group-other permission model.

Capabilities: Capabilities are a security mechanism where processes are granted specific privileges or capabilities to access certain resources or perform certain actions, providing a more fine-grained and dynamic approach to access control in an operating system.

(Protection matrix is also known as the access matrix. **access matrix** is implemented using **ACL and Capability** - implementation of protection matrix column-wise
Capability- implementation of protection **matrix ROW-wise)**

14. **Difference between normal Hard disk vs SSD?**
A normal hard disk (HDD) uses spinning magnetic disks to store data, while a solid-state drive (SSD) uses flash memory chips, providing faster access times and improved durability.

15. **What is UFS (Universal Flash Storage)? Where is it used? Latest version?**
Universal Flash Storage (UFS) is a flash storage specification for digital cameras, mobile phones and consumer electronic devices. It was designed to bring higherdata transfer speed and increased reliability to flash memory storage, while reducing market confusion and removing the need for different adapters for different types of cards.
UFS uses **NAND flash**.
UFS 3.1 has been implemented in Snapdragon 855+/860, Snapdragon 865, Snapdragon 870, Snapdragon 888, Exynos 2100, and Exynos 2200.

UFS 4.0 has been implemented in MediaTek Dimensity 9200 and Snapdragon 8 Gen 2.

16. ***Linux Commands for**
- File creation: touch filename.txt

- Folder/directory creation: mkdir foldername;
- List File properties: ls -l
- To change file permission: chmod u+x filename.txt

17. Basic step in Linux kernel compilation?

1. Obtain the kernel source code from the official Linux kernel website or a repository.
2. Install the necessary build tools and dependencies, including GCC compiler and development libraries.
3. Configure the kernel by running `make menuconfig` or similar commands to select desired features and options.
4. Build the kernel by running `make` command. This will compile the kernel source code into a binary image.
5. Install the compiled kernel by running `make install` or manually copying the resulting kernel image and related files to the appropriate location.
6. Update the bootloader configuration to recognize and boot the new kernel.
7. Reboot the system to start using the newly compiled kernel.

EXTRA'S

***Examples of Pre-emptive CPU Scheduling**: SJF Pre-emptive or SRTF, Round Robin, Priority

***Examples of Non-Pre-emptive CPU Scheduling:** FCFS, SJF(Default)

Arrival Time, AT = The time at which process enters the ready queue or ready state.

Burst Time, BT: Time required (time duration) by a process to get executed on CPU.

*Completion Time, CT= The time at which the process completes its execution.

*Turn Around Time, TAT: CT – AT or WT + BT

Waiting Time, WT = TAT – BT

*Response Time, RT= (Time at which a process gets CPU first time) - AT

***Output?**

a)

```
if(fork() && fork()) // prints 4 times Hello – short-circuit evaluation
    fork();
printf("Hello");
```

b)

```
if (fork() || fork()) // prints 5 times 1
    fork();
printf("1 ");
```

c)

```
fork();            // prints 8(2^3) times Hello
fork();
fork();
printf("hello\n")
;
```

d) *A counting semaphore S is initialized to 10. Then, 6 P operations and 4 V operationsare performed on S. What is the final value of S? // *10 – 6P + 4V = 8*