Operating System Assignment

Assignment Given by:

Mrs.Kiran Waghmare

Submitted by: Team 11

Creative Debuggers.

1.Define the types of Operating System?

-An Operating System (OS) is a software that acts as an interface between computer hardware components and the user.

Following are the popular types of OS (Operating System):

1.Batch Operating System

2.Multitasking/Time Sharing OS

3.Multiprocessing OS

4.Real Time OS

5.Distributed OS

1. Batch Operating System –

-This type of operating system does not interact with the computer directly.

-There is an operator which takes similar jobs having the same requirement and group them into batches. -

It is the responsibility of the operator to sort jobs with similar needs.

Examples of Batch based Operating System: Payroll System, Bank Statements, etc.

2.Multitasking/Time Sharing OS

-Each task is given some time to execute so that all the tasks work smoothly.

-Each user gets the time of CPU as they use a single system. These systems are also known as Multitasking Systems.

-The task can be from a single user or different users also. The time that each task gets to execute is called quantum.

-After this time interval is over OS switches over to the next task.

Examples of Time-Sharing OSs are: Multics, Unix, etc.

3.Multiprocessing OS

-A multiprocessing operating system (OS) is one in which two or more central processing units (CPUs) control the functions of the computer.

Example: UNIX Operating system is one of the most widely used multiprocessing systems.

4.Real time OS

-A real time operating system time interval to process and respond to inputs is very small.

Examples: Military Software Systems, Space Software Systems are the Real time OS example.

5. Distributed Operating System

–These types of the operating system is a recent advancement in the world of computer technology and are being widely accepted all over the world and, that too, with a great pace.

-Various autonomous interconnected computers communicate with each other using a shared communication network.

-Independent systems possess their own memory unit and CPU. These are referred to as distributed systems. -These system's processors differ in size and function.

-The major benefit of working with these types of the operating system is that it is always possible that one user can access the files or software which are not actually present on his system but some other system connected within this network i.e., remote access is enabled within the devices connected in that network.

Examples of Distributed Operating System are- LOCUS, etc

2. Explain DHCP?

-DHCP (Dynamic Host Configuration Protocol) is a network management protocol used to dynamically assign an Internet Protocol (IP) address to any device, or node, on a network so they can communicate using IP.

-DHCP automates and centrally manages these configurations rather than requiring network administrators to manually assign IP addresses to all network devices.

-DHCP can be implemented on small local networks, as well as large enterprise networks.

-DHCP will assign new IP addresses in each location when devices are moved from place to place, which means network administrators do not have to manually configure each device with a valid IP address or reconfigure the device with a new IP address if it moves to a new location on the network.

-This dynamic process eliminates the need to manually assign IP addresses.

-A DHCP server can be set up and the hosts can be configured to automatically obtain an IP address.

3.Explain DNS?

-Domain Name Server (DNS) is a standard protocol that helps Internet users discover websites using human readable addresses. -Like a phonebook which lets you look up the name of a person and discover their number, DNS lets you type the

address of a website and automatically discover the Internet Protocol (IP) address for that website.

-Without DNS, the Internet would collapse - it would be impossible for people and machines to access Internet servers via the friendly URLs they have come to know.

DNS is Used For

-Resolving names of World Wide Web (WWW) sites

-Routing messages to email servers and webmail servicesConnecting app servers, databases and middleware within a web application

-Virtual Private Networks (VPN)

-Peer-to-peer sharing programs

-Multiplayer games

-Instant messaging and online meeting services

-Communication between IoT devices, gateways and servers.

4.Explain paging?

-Paging permits the physical address space of a process to be non-contiguous. It is a fixed-size partitioning scheme. In the Paging technique, the secondary memory and main memory are divided into equal fixed-size partitions.

-Paging solves the problem of fitting memory chunks of varying sizes onto the backing store and this problem is suffered by many memory management schemes.

-Paging helps to avoid external fragmentation and the need for compaction.

-The paging technique divides the physical memory(main memory) into fixed-size blocks that are known as Frames and also divide the logical memory(secondary memory) into blocks of the same size that are known as Pages.

-This technique keeps the track of all the free frames.

-The Frame has the same size as that of a Page. A frame is basically a place where a (logical) page can be (physically) placed.

5.Explain segmentation?

-Segmentation is a way of dividing the addressable memory.

-It is scheme of memory management and it generally supports the user view of memory.

-The Logical address space is basically the collection of segments. Each segment has a name and a length.

-Basically, a process is divided into segments. Like paging, segmentation divides or segments the memory. But there is a difference and that is while the paging divides the memory into a fixed size and on the other hand, segmentation divides the memory into variable segments these are then loaded into logical memory space.

Some characteristics of the segmentation technique are as follows:

1.The Segmentation partitioning scheme is variable-size.

2.Partitions of the secondary memory are commonly known as segments.

3.Partition size mainly depends upon the length of modules.

Thus with the help of this technique, secondary memory and main memory are divided into unequal-sized partitions.

Q6)Explain memory managmment?

-> Memory managment is a functionality of a operating system which handles on manages primary memory and moves processes back and fourth between main memory maanagment keeps tracks of each and every memory location,regardless of either it is allocated to some processes on it is true.

Q7)Explain the function of os?

-> An operating System is a program that acts as auser-computer GUI(Graphical user interface)it contains the execution of all types of applications following are thw functions of the os:

1)Instruction:In operating system establishes a mutual understanding in between the various instruction given by the user

2)Input/output managment

3)memory managment

4)file managment

5)process managment

6)job priority

7)security

Q8) Explain kernel,Its architecture and working?

-> The kernel is core of operating system. It is software and providing secure access to the machines hardware since there are many programs and resources are limited,the kernel also decides when and how long a program should run.This is called Scheduling

9. Explain a shell script?

Ans : A shell script is a list of commands in a computer program that is run by the Unix shell which is a command line interpreter. The different operations performed by shell scripts are program execution, file manipulation and text printing.

Types of Shells

Bourne Shell

This is default shell for version 7 Unix. The character $ is the default prompt for the bourne shell. The different subcategories in this shell are Korn shell, Bourne Again shell, POSIX shell etc.

C Shell

This is a Unix shell and a command processor that is run in a text window. The character % is the default prompt for the C shell. File commands can also be read easily by the C shell, which is known as a script.

The capabilities of shell script are:-

Batch jobs

Several commands entered in command line can be executed automatically using shell scripting.

Programming

The various methods, arrays, variables, comments

Generalisation , shortcuts

10. Explain a page fault?

Ans : Page fault dominates more like an error. It mainly occurs when any program tries to access the data or the code that is in the address space of the program, but that data is not currently located in the RAM of the system.

- So basically when the page referenced by the CPU is not found in the main memory then the situation is termed as Page Fault.
- Whenever any page fault occurs, then the required page has to be fetched from the secondary memory into the main memory.

In case if the required page is not loaded into the memory, then a page fault trap arises

The page fault mainly generates an exception, which is used to notify the operating system that it must have to retrieve the "pages" from the virtual memory in order to continue the execution. Once all the data is moved into the physical memory the program continues its execution normally. The Page fault process takes place in the background and thus goes unnoticed by the user.

- The hardware of the computer tracks to the kernel and the program counter (PC) is generally saved on the stack.CPU registers store the information of the current state of instruction.
- An assembly program is started that usually saves the general registers and also saves the other volatile information to prevent the OS from destroying it.

Handling the Page Fault

Given below is the simple procedure to handle the page fault:

Figure: Steps to Handle the Page fault

If you will access a page that is marked as invalid then it also causes a Page Fault. Then the Paging hardware during translating the address through the page table will notice that the invalid bit is set that will cause a trap to the Operating system.

This trap is mainly the result of the failure of the Operating system in order to bring the desired page into memory.

Let us understand the procedure to handle the page fault as shown with the help of the above diagram:

1. First of all, internal table(that is usually the process control block) for this process in order to determine whether the reference was valid or invalid memory access.
2. If the reference is invalid, then we will terminate the process. If the reference is valid, but we have not bought in that page so now we just page it in.
3. Then we locate the free frame list in order to find the free frame.
4. Now a disk operation is scheduled in order to read the desired page into the newly allocated frame.
5. When the disk is completely read, then the internal table is modified that is kept with the process, and the page table that mainly indicates the page is now in memory.
6. Now we will restart the instruction that was interrupted due to the trap. Now the process can access the page as though it had always been in memory.

11. Explain a deadlock?

Ans: Every process needs some resources to complete its execution. However, the resource is granted in a sequential order.

1. The process requests for some resource.
2. OS grant the resource if it is available otherwise let the process waits.
3. The process uses it and release on the completion.

A Deadlock is a situation where each of the computer process waits for a resource which is being assigned to some another process. In this situation, none of the process gets executed since the resource it needs, is held by some other process which is also waiting for some other resource to be released.

Let us assume that there are three processes P1, P2 and P3. There are three different resources R1, R2 and R3. R1 is assigned to P1, R2 is assigned to P2 and R3 is assigned to P3.

After some time, P1 demands for R1 which is being used by P2. P1 halts its execution since it can't complete without R2. P2 also demands for R3 which is being used by P3. P2 also stops its execution because it can't continue without R3. P3 also demands for R1 which is being used by P1 therefore P3 also stops its execution.

In this scenario, a cycle is being formed among the three processes. None of the process is progressing and they are all waiting. The computer becomes unresponsive since all the processes got blocked.

12. Define the necessary conditions for deadlock?

Ans: Necessary conditions for Deadlocks

1. Mutual Exclusion

A resource can only be shared in mutually exclusive manner. It implies, if two process cannot use the same resource at the same time.

2. Hold and Wait

A process waits for some resources while holding another resource at the same time.

3. No pre-emption

The process which once scheduled will be executed till the completion. No other process can be scheduled by the scheduler meanwhile.

4. Circular Wait

All the processes must be waiting for the resources in a cyclic manner so that the last process is waiting for the resource which is being held by the first process.

13) Explain a semaphore?

Ans:- Semaphores are integer variables that are used to solve the critical section problem by using two atomic operations, wait and signal that are used for process synchronization.

The definitions of wait and signal are as follows –

- Wait

The wait operation decrements the value of its argument S, if it is positive. If S is negative or zero, then no operation is performed.

- Signal

The signal operation increments the value of its argument S.

Types of Semaphores

There are two main types of semaphores i.e. counting semaphores and binary semaphores. Details about these are given as follows –

- Counting Semaphores

These are integer value semaphores and have an unrestricted value domain. These semaphores are used to coordinate the resource access, where the semaphore count is the number of available resources. If the resources are added, semaphore count automatically incremented and if the resources are removed, the count is decremented.

- Binary Semaphores

The binary semaphores are like counting semaphores but their value is restricted to 0 and 1. The wait operation only works when the semaphore is 1 and the signal operation succeeds when semaphore is 0. It is sometimes easier to implement binary semaphores than counting semaphores.

Advantage of Semaphores

Some of the advantages of semaphores are as follows –

- Semaphores allow only one process into the critical section. They follow the mutual exclusion principle strictly and are much more efficient than some other methods of synchronization.
- There is no resource wastage because of busy waiting in semaphores as processor time is not wasted unnecessarily to check if a condition is fulfilled to allow a process to access the critical section.
- Semaphores are implemented in the machine independent code of the microkernel. So they are machine independent.

Disadvantages of Semaphores

Some of the disadvantages of semaphores are as follows –

- Semaphores are complicated so the wait and signal operations must be implemented in the correct order to prevent deadlocks.
- Semaphores are impractical for last scale use as their use leads to loss of modularity. This happens because the wait and signal operations prevent the creation of a structured layout for the system.
- Semaphores may lead to a priority inversion where low priority processes may access the critical section first and high priority processes later.

14) Explain a mutex?

A mutex is a binary variable whose purpose is to provide locking mechanism. It is used to provide mutual exclusion to a section of code, means only one process can work on a particular code section at a time.

There is misconception that binary semaphore is same as mutex variable but both are different in the sense that binary semaphore apart from providing locking mechanism also provides two atomic operation signal and wait, means after releasing resource semaphore will provide signaling mechanism for the processes who are waiting for the resource.

15) Difference among kernel space and user space.

Kernel Space	User Space
Running program by keeping CPU in low privilege mode	Running program by keeping CPU in high privilege mode.
Library calls are present	System calls are present
Library calls are portable	System calls are fast
Indirect way of accessing kernel	Direct way of accessing kernel
Library calls are slow	System calls are fast
Program starts with main()	There are many entry points
Segmentation fault and infinite loop will not give problem	Segmentation fault and infinite loop will corrupt program
Returns is not compulsory	Return is compulsory

16) Write in brief the ping command.

Ans:- Ping is a command-line utility, available on virtually any operating system with network connectivity, that acts as a test to see if a networked device is reachable. The ping command sends a request over the network to a specific device.

17) Explain UNIX

Unix is an operating system which was discovered by Dennis Ritchie and Ken Thompson at AT&T bell labs in 1960. Since then, it has been under constant development and has various flavours.

Some of the UNIX based Operating systems ⟹ Sun Solaris, GNU/Linux, and MacOS X

Characteristics of UNIX

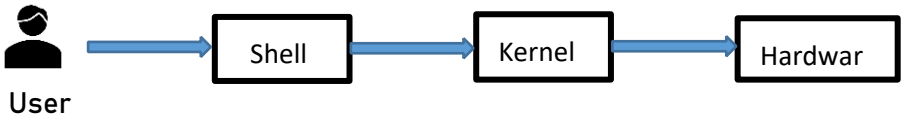
- It is open-source software.

- It is free ware.
- Multiuser Operating system
- Multitasking Operating system ➢ Supports both GUI and CUI.
- More secure as compared with other operating systems.

Components of LINUX operating system

LINUX has two important components

1-Shell 2-Kernel



Shell

It is outer layer of OS.

Shell reads the commands given by Programmer.

Check the validity of that command

If given command is valid then Shell convert that command into Kernel understandable format and handover it to Kernel.

Acts as an interface between user and Kernel.

Kernel

Kernel is the core part of OS.

It acts as a interface between Shell and hardware.

Kernel is responsible for executing user command with the help of hardware.

Memory allocation, Process allocation is handled by Kernel.

Q-18) Explain Grep

Grep is an acronym that stands for Global Regular Expression Print.

Grep is a Linux / Unix command-line tool used to search for a string of characters in a specified file. The text search pattern is called a regular expression. When it finds

a match, it prints the line with the result. The grep command is handy when searching through large log files.

Example1- In the given text file searching a particular keyword and printing the lines in which that word is available using grep command

Syntax → **grep keyword fileName**

```
sanghapal@DESKTOP-NF3DC6U:~$ cat abc.txt
1-Sky is blue
2-Rose is red
3-Leaves are green
4-Blood is red
5-Sunflower is yellow
6-Water is colorless
sanghapal@DESKTOP-NF3DC6U:~$ grep red abc.txt
2-Rose is red
4-Blood is red
```

Example2- In the given text file searching a particular keyword and printing the lines in which that word is unavailable using grep command

Syntax → grep -v keyword fileName

```
sanghapal@DESKTOP-NF3DC6U:~$ cat abc.txt
1-Sky is blue
2-Rose is red
3-Leaves are green
4-Blood is red
5-Sunflower is yellow
6-Water is colorless
sanghapal@DESKTOP-NF3DC6U:~$ grep -v red abc.txt
1-Sky is blue
3-Leaves are green
5-Sunflower is yellow
6-Water is colorless
```

Example3- In the given text file searching a particular keyword and printing the lines in which that word is available with

Case1- Searched keyword + Succeeding line

Case2- Preceding line+ Searched keyword

Case3- Preceding line +Searched keyword + a succeeding line

Note – Numbers after alphabet A/B/C indicates number of lines to be printed. We can change that number according to need.

Syntax

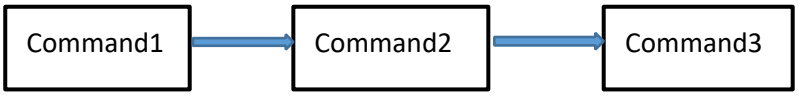
```
grep -A1 keyword filename
grep -B1 keyword filename
grep -C1 keyword filename
```

```
sanghapal@DESKTOP-NF3DC6U:~$ cat abc.txt
1-Sky is blue
2-Rose is red
3-Leaves are green
4-Blood is red
5-Sunflower is yellow
6-Water is colorless
sanghapal@DESKTOP-NF3DC6U:~$ grep -A1 yellow abc.txt
5-Sunflower is yellow
6-Water is colorless
sanghapal@DESKTOP-NF3DC6U:~$ grep -B1 yellow abc.txt
4-Blood is red
5-Sunflower is yellow
sanghapal@DESKTOP-NF3DC6U:~$ grep -C1 yellow abc.txt
4-Blood is red
5-Sunflower is yellow
6-Water is colorless
```

Q-19) Explain Pipe

Output of one command is given as an input to another command.

Two or more commands can be combined with the help of Pipe.



Syntax-

Two commands	Command1 | Command2
Multiple commands	Command1 | Command2 | Command3 | Command4

Example1- Sort the list using Pipe

Here we are displaying the content of a text file and sorting at the same time with the help of piping.

```
sanghapal@DESKTOP-NF3DC6U:~$ cat > new.txt
321
854
763
321
092
854
092
954
193
763
092
sanghapal@DESKTOP-NF3DC6U:~$ cat new.txt | sort
092
092
092
193
321
321
763
763
854
854
954
```

Example2- Sort the list and remove duplicate tags using Pipe

Here we are displaying the content of a text file, sorting and removing duplicate tags at the same time with the help of piping.

```
sanghapal@DESKTOP-NF3DC6U:~$ cat > new.txt
321
854
763
321
092
854
092
954
193
763
092
sanghapal@DESKTOP-NF3DC6U:~$ cat new.txt | sort
092
092
092
193
321
321
763
763
854
854
954
sanghapal@DESKTOP-NF3DC6U:~$ cat new.txt | sort | uniq
092
193
321
763
854
954
```

Q-20) Explain difference between Process and Thread

Comparison Basis	Process	Thread
Definition	A process is a program under execution i.e an active program.	A thread is a lightweight process that can be managed independently by a scheduler.
Context switching time	Processes require more time for context switching as they are more heavy.	Threads require less time for context switching as they are lighter than processes.
Memory Sharing	Processes are totally independent and don't share memory.	A thread may share some memory with its peer threads.
Communication	Communication between processes requires more time than between threads.	Communication between threads requires less time than between processes .
Blocked	If a process gets blocked, remaining processes can continue execution.	If a user level thread gets blocked, all of its peer threads also get blocked.
Resource Consumption	Processes require more resources than threads.	Threads generally need less resources than processes.
Dependency	Individual processes are independent of each other.	Threads are parts of a process and so are dependent.
Data and Code sharing	Processes have independent data and code segments.	A thread shares the data segment, code segment, files etc. with its peer threads.
Treatment by OS	All the different processes are treated separately by the operating system.	All user level peer threads are treated as a single task by the operating system.
Time for creation	Processes require more time for creation.	Threads require less time for creation.
Time for termination	Processes require more time for termination.	Threads require less time for termination.

21) Explain a Scheduling Algorithm?

Ans:-

CPU scheduling deals with the problem of deciding which of the processes in the ready queue is to be allocated the CPU.

Types of Scheduling Algorithms:

There are eight different types of scheduling algorithms whose name are as follows:

To understand these scheduling, first, we discuss some terms that we need in each scheduling process.

- Arrival time is the time at which the process arrives in the ready queue for execution,

  and it is given in our table when we need to calculate the average waiting time.

- Completion time is the time at which the process finishes its execution.
- Turnaround time is the difference between completion time and arrival time,

  i.e. turnaround time = Completion time- arrival time
- Burst time is the time required by the process for execution of the process by CPU.
- Waiting time (W.T) is the difference between turnaround time and burst time,

  i.e. waiting time= Turnaround time – Burst time

==========

22) Explain pre-emptive and non-preemptive scheduling?

Ans:-Preemptive Scheduling is a CPU scheduling technique that works by dividing time slots of CPU to a given process.

The time slot given might be able to complete the whole process or might not be able to it.

When the burst time of the process is greater than CPU cycle, it is placed back into the ready queue and will execute in the next chance.

This scheduling is used when the process switch to ready state.

Algorithms that are backed by preemptive Scheduling are round-robin (RR), priority, SRTF (shortest remaining time first).

Non-preemptive Scheduling is a CPU scheduling technique the process takes the resource (CPU time) and

holds it till the process gets terminated or is pushed to the waiting state.
No process is interrupted until it is completed, and after that processor switches
to another process.

Algorithms that are based on non-preemptive Scheduling are non-preemptive priority, and shortest Job first.

Preemptive Vs Non-Preemptive Scheduling

Preemptive Scheduling	Non-Preemptive Scheduling
Resources are allocated according to the cycles for a limited time.	Resources are used and then held by the process until it gets terminated.
The process can be interrupted, even before the completion.	The process is not interrupted until its life cycle is complete.
Starvation may be caused, due to the insertion of priority process in the queue.	Starvation can occur when a process with large burst time occupies the system.
Maintaining queue and remaining time needs storage overhead.	No such overheads are required.

==========

23) Define different scheduling algorithm?

Ans:- Different Scheduling Algorithms

First Come First Serve (FCFS): Simplest scheduling algorithm that schedules according to arrival times of processes.

First come first serve scheduling algorithm states that the process that requests the CPU first is allocated the CPU first.

It is implemented by using the FIFO queue. When a process enters the ready queue, its PCB is linked onto the tail of the queue.

When the CPU is free, it is allocated to the process at the head of the queue. The running process is then removed from the queue.

FCFS is a non-preemptive scheduling algorithm.

Note:First come first serve suffers from convoy effect.

Shortest Job First (SJF): Process which have the shortest burst time are scheduled first.

If two processes have the same bust time then FCFS is used to break the tie. It is a non-preemptive scheduling algorithm.

Longest Job First (LJF): It is similar to SJF scheduling algorithm. But, in this scheduling algorithm,

we give priority to the process having the longest burst time. This is non-preemptive in nature

i.e., when any process starts executing, can't be interrupted before complete execution.

Shortest Remaining Time First (SRTF): It is preemptive mode of SJF algorithm in which jobs are schedule according to shortest remaining time.

Longest Remaining Time First (LRTF): It is preemptive mode of LJF algorithm in which we give priority to the process having largest burst time remaining.

Round Robin Scheduling: Each process is assigned a fixed time(Time Quantum/Time Slice) in cyclic way.It is designed especially for the time-sharing system.

The ready queue is treated as a circular queue. The CPU scheduler goes around the ready queue, allocating the CPU to each process for a time interval of up to 1-time quantum.

To implement Round Robin scheduling, we keep the ready queue as a FIFO queue of processes. New processes are added to the tail of the ready queue.

The CPU scheduler picks the first process from the ready queue, sets a timer to interrupt after 1-time quantum, and dispatches the process.

One of two things will then happen. The process may have a CPU burst of less than 1-time quantum. In this case, the process itself will release the CPU voluntarily.

The scheduler will then proceed to the next process in the ready queue. Otherwise,

if the CPU burst of the currently running process is longer than 1time quantum, the timer will go off and will cause an interrupt to the operating system.

A context switch will be executed, and the process will be put at the tail of the ready queue.

The CPU scheduler will then select the next process in the ready queue.

Priority Based scheduling (Non-Preemptive): In this scheduling, processes are scheduled according to their priorities,

i.e., highest priority process is scheduled first. If priorities of two processes match,

then schedule according to arrival time. Here starvation of process is possible.

Highest Response Ratio Next (HRRN): In this scheduling, processes with highest response ratio is scheduled. This algorithm avoids starvation.

Response Ratio = (Waiting Time + Burst time) / Burst time

Multilevel Queue Scheduling: According to the priority of process, processes are placed in the different queues. Generally high priority process are placed in the top level queue.

Only after completion of processes from top level queue, lower level queued processes are scheduled. It can suffer from starvation.

Multi level Feedback Queue Scheduling: It allows the process to move in between queues. The idea is to separate processes according to the characteristics of their CPU bursts.

If a process uses too much CPU time, it is moved to a lower-priority queue.

==========

24) Explain Booting process?

Ans:- An operating system (OS) is the low-level software that manages resources, controls peripherals, and provides basic services to other software. In Linux, there are 6 distinct stages in the typical booting process.

1. BIOS

BIOS stands for Basic Input/Output System. In simple terms, the BIOS loads and executes the Master Boot Record (MBR) boot loader.

When you first turn on your computer, the BIOS first performs some integrity checks of the HDD or SSD.

Then, the BIOS searches for, loads, and executes the boot loader program, which can be found in the Master Boot Record (MBR). The MBR is sometimes on a USB stick or CD-ROM such as with a live installation of Linux.

Once the boot loader program is detected, it's then loaded into memory and the BIOS gives control of the system to it.

2. MBR

MBR stands for Master Boot Record, and is responsible for loading and executing the GRUB boot loader.

The MBR is located in the 1st sector of the bootable disk, which is typically /dev/hda, or /dev/sda, depending on your hardware. The MBR also contains information about GRUB, or LILO in very old systems.

3. GRUB

Sometimes called GNU GRUB, which is short for GNU GRand Unified Bootloader, is the typical boot loader for most modern Linux systems.

The GRUB splash screen is often the first thing you see when you boot your computer. It has a simple menu where you can select some options. If you have multiple kernel images installed, you can use your keyboard to select the one you want your system to boot with. By default, the latest kernel image is selected.

The splash screen will wait a few seconds for you to select and option. If you don't, it will load the default kernel image.

In many systems you can find the GRUB configuration file at /boot/grub/grub.conf or /etc/grub.conf. Here's an example of a simple grub.conf file:

```
#boot=/dev/sda  default=0 timeout=5
splashimage=(hd0,0)/boot/grub/splash.xpm.gz  hiddenmenu
title CentOS (2.6.18-194.el5PAE)       root (hd0,0)       kernel
/boot/vmlinuz-2.6.18-194.el5PAE ro root=LABEL=/       initrd
/boot/initrd-2.6.18-194.el5PAE.img
```

4. Kernel

The kernel is often referred to as the core of any operating system, Linux included. It has complete control over everything in your system.

In this stage of the boot process, the kernel that was selected by GRUB first mounts the root file system that's specified in the grub.conf file. Then it executes the /sbin/init program, which is always the first program to be executed. You can confirm this with its process id (PID), which should always be 1.

The kernel then establishes a temporary root file system using Initial RAM Disk (initrd) until the real file system is mounted.

5. Init

At this point, your system executes runlevel programs. At one point it would look for an init file, usually found at /etc/inittab to decide the Linux run level.

Modern Linux systems use systemd to choose a run level instead. According to TecMint, these are the available run levels:

Run level 0 is matched by poweroff.target (and runlevel0.target is a symbolic link to poweroff.target).

Run level 1 is matched by rescue.target (and runlevel1.target is a symbolic link to rescue.target).

Run level 3 is emulated by multi-user.target (and runlevel3.target is a symbolic link to multi-user.target).

Run level 5 is emulated by graphical.target (and runlevel5.target is a symbolic link to graphical.target).

Run level 6 is emulated by reboot.target (and runlevel6.target is a symbolic link to reboot.target).

Emergency is matched by emergency.target.

systemd will then begin executing runlevel programs.

6. Runlevel programs

Depending on which Linux distribution you have installed, you may be able to see different services getting started. For example, you might catch starting sendmail …. OK.

These are known as runlevel programs, and are executed from different directories depending on your run level. Each of the 6 runlevels described above has its own directory:

Run level 0 – /etc/rc0.d/ Run

level 1 – /etc/rc1.d/

Run level 2 – /etc/rc2.d/

Run level 3 – /etc/rc3.d/ Run

level 4 – /etc/rc4.d/

Run level 5 – /etc/rc5.d/

Run level 6 – /etc/rc6.d/

25. Explain bios?

Ans. It is Basic Input Output Device which checks the peripheral devices like fans, HDD, etc. are connected or not to the motherboard. The booting of a Windows XP PC begins when the hardware powers on and the BIOS begins executing from ROM. The BIOS identifies the system device to be booted and loads and executes the bootstrap loader from the front of the disk.

26. Explain the difference among static memory allocation and dynamic memory allocation?

Ans. 1). In the static memory allocation, variables get allocated permanently, till program executes or function call finishes. Where in, Dynamic memory allocation, variables get allocated only if your program unit gets active 2). Static Memory Allocation is done before program execution. In the static memory allocation, variables get allocated permanently, till program executes or function call finishes. Where in, Dynamic Memory Allocation is done during program execution. In the Dynamic memory allocation, variables Get allocated only if your program unit gets active. 3). It uses stack for managing the static allocation of memory. Where in dynamic, uses heap for managing the static allocation of memory. 4). Static memory is less efficient. Since, dynamic memory is more efficient. 5). In Static Memory allocation, there is memory re-usability. Where Dynamic Memory allocation, there is memory re-usability and memory can be freed when not required. 6). In static memory allocation, once the memory is allocated, the memory size can not change. Since, in dynamic memory allocation, when memory is allocated the memory size can be changed. 7). In static memory allocation scheme, we cannot reuse the unused memory.This allows reusing the memory. The user can allocate more memory when required. Also, the user can release the memory when the user needs it. 8). In static memory allocation scheme, execution is faster than dynamic memory allocation. In dynamic memory allocation

scheme, execution is slower than static memory allocation. 9). Static memory is allocated at compile time. Where as in, dynamic memory is allocated at compile time. 10). In static allocated memory remains from start to end of the program. In this memory is allocated at compile time. Where dynamically allocated memory can be released at any time during the program.

27. UNIX commands like touch, sed, grep.

Ans 1).touch :- The touch command is a standard command used in UNIX/Linux operating system which is used to create, change and modify timestamps of a file. Syntax :- touch file_name 2).sed :- SED command in UNIX is stands for stream editor and it can perform lot's of function on file like, searching, find and replace, insertion or deletion. Though most common use of SED command in UNIX is for substitution or for find and replace. By using SED you can edit files even without opening it, which is much quicker way to find and replace something in file, than first opening that file in VI Editor and then changing it. Syntax :- sed OPTIONS... [SCRIPT] [INPUTFILE...] 3).grep :- The grep filter searches a file for a particular pattern of characters, and displays all lines that contain that pattern. The pattern that is searched in the file is referred to as the regular expression (grep stands for globally search for regular expression and print out). Syntax :- grep [options] pattern [files]

28. Explain a process and process table? Define different states of process?

Ans. Process :- A process is basically a program in execution. The execution of a process must progress in a sequential fashion. A process is defined as an entity which represents the basic unit of work to be implemented in the system. To keep track on all running programs, an OS maintains few data structures referred as kernel data structures: 1. Job queue: it contains list of PCB's (Procees Control Block) of all submitted processes. 2. Ready queue: it contains list of PCB's of processes which are in the main memory and waiting for the CPU time. 3. Waiting queue: it contains list of PCB's of processes which are requesting for that particular device. 1. Job Scheduler/Long Term Scheduler: it is a system program which selects/schedules jobs/processes from job queue to load them onto the ready queue. 2. CPU Scheduler/Short Term Scheduler: it is a system program which selects/schedules job/process from ready queue to load it onto the CPU. Dispatcher: it is a system program which loads a process onto the CPU which is scheduled by the CPU scheduler, and the time required for the dispatcher to stops an execution of one process and to starts an execution of another process is referred as dispatcher latency. States of Process:- - Throughout execution, process goes through different states out of which at a time it can be only in a one state. 1. New state: upon submission or when a PCB for a process gets created into the main memory process is in a new state. 2. Ready state: after submission, if process is in the main memory and waiting for the CPU time, it is in a ready state. 3. Running state: if currently the CPU is executing any process then state of that process is considered as a running state. 4. Waiting state: if a process is requesting for any i/o device then state of that process is considered as a waiting state. 5. Terminated state: upon exit, process goes into terminated state and its PCB gets destroyed from the main memory.

Que 29. Define the benefits of multithreaded programming?

Ans :- Multithreading allows execution of multiple parts of a program at a same time.

These

Parts are known as threads and are lightweigth process available within the process.

Benifits of Multithreading are:-

1. Ressource sharing:- All the threads of a process share its resouces such as memory, data, files etc.

2. Responsivness:-Program responsivness allows a program to run even if part of it is blocked using multithreading. this can also be done if the process is performing a lengthy operations.

3. Utilization of multiporcessor:- In multiprocessor architecture, each thread can run of a different processor in parallel using multithreading. this increases concurrency of the system.

Que 30. What is thrashing?

Ans :- Thrashing is a condition when the system is spending a major portion of its in servicing the the page fault, but the actual processing done is very negligible.

If a number of program are running in comparitively smaller RAM, then a lot of system time will be spend into page swappig (Paging) activity.

due to this performance is reduced and this problem can be solved by increasing RAM size in machine.

Que 31. Explain Belady's Anomaly?

Ans: - In Operating System, process data is loaded in fixed-sized chunks and each chunk is referred to as a page.

The processor loads these pages in the fixed-sized chunks of memory called frames. Typically the size of each page is always equal to the frame size.

A page fault occurs when a page is not found in the memory and needs to be loaded from the disk. If a page

fault occurs and all memory frames have been already allocated, then replacement of a page in memory is

required on the request of a new page. This is referred to as demand-paging. The choice of which page to

replace is specified by page replacement algorithms. The commonly used page replacement algorithms are FIFO,

LRU, optimal page replacement algorithms, etc.

Generally, on increasing the number of frames to a process' virtual memory, its execution becomes faster as

fewer page faults occur. Sometimes the reverse happens, i.e. more page faults occur when more frames are allocated to a process. This most unexpected result is termed Belady's Anomaly.

Que 32. Explain starvation and aging?

Ans :- In CPU scheduling priority algo. each process is associated with a number called as priority of process. lower the number higher the priority.

Process having low priority may not get enough cpu time for execution , this is called starvation.

to resolve this priority of such processes can be incremented priodically this is called aging.

33. Explain a trap and trapdoor?

Answer: Trapdoor is a secret undocumented entry point into a program used to grant access without normal methods of access authentication.

A trap is a software interrupt, usually the result of an error condition, and is also a non maskable inerrupt and has highest priority.

34. Explain a daemon?

Answer: In modern computers, everything is done through processes. Now a process can be either user interactive or a system process running in background.A user-interactive process consists of browsers, text editors etc.

A daemon is typically a system process running in background - most of the times waiting for an event to occur, or can provide certain services.

35. Which application software's executed on OS?

Answer:

36. Define daemon objects and thread objects?

Answer: Daemon threads is a low priority thread that provide supports to user threads. These threads can be user defined and system defined as well. Garbage collection thread is one of the system generated daemon thread that runs in background. These threads run in the background to perform tasks such as garbage collection. Daemon thread does allow JVM from existing until all the threads finish their execution. When a JVM founds daemon threads it terminates the thread and then shutdown itself, it does not care Daemon thread whether it is running or not.

1. void setDaemon(boolean status)

In Java, this method is used to create the current thread as a daemon thread or user thread. If there is a user thread as obj1 then obj1.setDaemon(true) will make it a Daemon thread and if there is a Daemon thread obj2 then calling obj2.setDaemon(false) will make it a user thread.

2. boolean isDaemon()

In Java, this method is used to check whether the current thread is a daemon or not. It returns true if the thread is Daemon otherwise it returns false.

37. Give commands for finding process ID.

PS:

The most common way to find out the Linux PID is to use the ps command: ps aux | grep name_of_process

In addition, the command will also display the PID for grep, because the process was started during the search. To remove it, add the following filter: ps aux | grep name_of_process | grep -v grep

Command PGREP:

If you do not need to see detailed information about the process, but only the PID is enough, you can use the pgrep utility:

PIDOF

The pidof command searches for the PID of a particular process by its name. No occurrences, the process name should only match the desired one:

Using the -s option, you can ask the command to display only one PID:

PSTREE:

The pstree command allows you to view a list of child processes for a specific process, as well as their pid identifiers. For example, look at the Apache process tree:

How to edit, rename and move file in Linux?

cat <file name>

This is the most popular and easy way to display the file content. It simply prints the file content to the terminal. It provides many options to make it more specific. To go in-depth with cat command, visit Linux cat

Rename a file

To rename a file there are other commands also like 'mv'. But 'rename' command is slightly advanced then others. This command will be rarely used and it works differently on different distros of linux. We'll work on Debian/Ubuntu examples.

Generally, renaming is not a big task, but when you want to rename a large group of files at once then it will be difficult to rename it with 'mv' command. In these cases, it is adviced to use 'rename' command. It can convert upper case files to lower case files and vice versa and cn overwrite files using perl expressions. This command is a part of perl script.

1. rename 's/old-name/new-name/' files

2. mv /old file name /new file name.

move file in Linux mv file1(file to be moved.) file2(to the file where it needs to be moved)

Q3) Give 5 commands in Linux with explanation?

Sol

1. pwd command

Use the pwd command to find out the path of the current working directory (folder) you're in. The command will return an absolute (full) path, which is basically a path of all the directories that starts with a forward slash (/). An example of an absolute path is /home/username.

2. cd command

To navigate through the Linux files and directories, use the cd command. It requires either the full path or the name of the directory, depending on the current working directory that you're in.
Let's say you're in /home/username/Documents and you want to go to Photos, a subdirectory of Documents. To do so, simply type the following command: cd Photos.
Another scenario is if you want to switch to a completely new directory, for example,/home/username/Movies. In this case, you have to type cd followed by the directory's absolute path: cd /home/username/Movies.
There are some shortcuts to help you navigate quickly:

- cd .. (with two dots) to move one directory up
- cd to go straight to the home folder
- cd- (with a hyphen) to move to your previous directory

On a side note, Linux's shell is case sensitive. So, you have to type the name's directory exactly as it is.

3. ls command

The ls command is used to view the contents of a directory. By default, this command will display the contents of your current working directory. If you want to see the content of other directories, type ls and then the directory's path. For example, enter ls /home/username/Documents to view the content of Documents.
There are variations you can use with the ls command:

- ls -R will list all the files in the sub-directories as well
- ls -a will show the hidden files
- ls -al will list the files and directories with detailed information like the permissions, size, owner, etc.

4. cat command

cat (short for concatenate) is one of the most frequently used commands in Linux. It is used to list the contents of a file on the standard output (sdout). To run this command, type cat followed by the file's name and its extension. For instance: cat file.txt.
Here are other ways to use the cat command:

- cat > filename creates a new file
- cat filename1 filename2>filename3 joins two files (1 and 2) and stores the output of them in a new file (3)
- to convert a file to upper or lower case use, cat filename | tr a-z A-Z >output.txt

5. cp command

Use the cp command to copy files from the current directory to a different directory. For instance, the command cp scenery.jpg /home/username/Pictures would create a copy of scenery.jpg (from your current directory) into the Pictures directory.

40) Sol

Deadlock Detection

Deadlock can be detected by the resource scheduler as it keeps track of all the resources that are allocated to different processes. After a deadlock is detected, it can be handed using the given methods –

- All the processes that are involved in the deadlock are terminated. This approach is not that useful as all the progress made by the processes is destroyed.
- Resources can be preempted from some processes and given to others until the deadlock situation is resolved.

Deadlock Prevention

It is important to prevent a deadlock before it can occur. So, the system checks each transaction before it is executed to make sure it does not lead to deadlock. If there is even a slight possibility that a transaction may lead to deadlock, it is never allowed to execute.

Some deadlock prevention schemes that use timestamps in order to make sure that a deadlock does not occur are given as follows –

- Wait - Die Scheme
- In the wait - die scheme, if a transaction T1 requests for a resource that is held by transaction T2, one of the following two scenarios may occur –
    - TS(T1) < TS(T2) - If T1 is older than T2 i.e T1 came in the system earlier than T2, then it is allowed to wait for the resource which will be free when T2 has completed its execution.
    - TS(T1) > TS(T2) - If T1 is younger than T2 i.e T1 came in the system after T2, then T1 is killed. It is restarted later with the same timestamp.

- Wound - Wait Scheme
- In the wound - wait scheme, if a transaction T1 requests for a resource that is held by transaction T2, one of the following two possibilities may occur – ○ TS(T1) < TS(T2) - If T1 is older than T2 i.e T1 came in the system earlier than T2, then it is allowed to roll back T2 or wound T2. Then T1 takes the resource and completes its execution. T2 is later restarted with the same timestamp.
    - ○ TS(T1) > TS(T2) - If T1 is younger than T2 i.e T1 came in the system after T2, then it is allowed to wait for the resource which will be free when T2 has completed its execution.

Deadlock Avoidance

It is better to avoid a deadlock rather than take measures after the deadlock has occurred. The wait for graph can be used for deadlock avoidance. This is however only useful for smaller databases as it can get quite complex in larger databases.

Wait for graph

The wait for graph shows the relationship between the resources and transactions. If a transaction requests a resource or if it already holds a resource, it is visible as an edge on the wait for graph. If the wait for graph contains a cycle, then there may be a deadlock in the system, otherwise not.