

# GEETANJALI GROUP OF COLLEGES

*(Affiliated to Saurashtra University)*

## BCA & BSc IT SEM-4 - UNIX

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## CS – 22 : Operating Systems Concepts With Unix / Linux

### Objectives:

- To provide the basic feature, function and interface with the hardware and application software to run the computer smoothly.

### Prerequisites:

Basic knowledge of operating system and its functionality along with its types.

Sr. No	Topic	Details	Marks in Weight %	Min. Lec.
1	Introduction,  Process and Thread,  Process Scheduling	<ul style="list-style-type: none"><li>Meaning of OS</li><li>Functions of OS</li><li>Features of OS</li><li>OS Types (User Point of View)</li><li>OS Types (Features Point of View)</li></ul> <ul style="list-style-type: none"><li>Process Definition ,</li><li>Process states ,</li><li>Process State transitions ,</li><li>Process Control Block ,</li><li>Context switching ,</li></ul> <ul style="list-style-type: none"><li>Threads,</li><li>Concept of multithreads ,</li><li>Benefits of threads,</li><li>Types of threads.</li></ul> <ul style="list-style-type: none"><li>Types of Schedulers</li><li>CPU scheduling algorithms<ul style="list-style-type: none"><li>FCFS</li><li>SJN</li><li>Round Robin</li><li>Priority Base Non Preemptive</li><li>Priority Base Preemptive</li></ul></li></ul>	20	18

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

2	<b>Deadlocks</b>  <b>Memory management</b>	<ul style="list-style-type: none"><li>• Deadlocks: Definition,</li><li>• Deadlock Prevention</li><li>• Deadlock Avoidance</li><li>• Deadlock Detection</li></ul>	20	12
		<ul style="list-style-type: none"><li>• Physical Memory and Virtual Memory</li><li>• Memory Allocation</li><li>• Internal and External fragmentation</li><li>• Contiguous Memory Allocation</li><li>• Noncontiguous Memory Allocation</li><li>• Virtual Memory Using Paging</li><li>• Virtual Memory Using Segmentation</li></ul>		
3	<b>Getting Started with Unix</b>  <b>Unix Shell Command</b>	<ul style="list-style-type: none"><li>• Unix Architecture</li><li>• Unix Features</li><li>• Types Of Shell ( C, Bourn, Korn )</li><li>• Unix File System</li><li>• Types Of Files<ul style="list-style-type: none"><li>o Ordinary Files</li><li>o Directory Files</li><li>o Device Files</li></ul></li><li>• Unix File &amp; Directory Permissions</li></ul>	20	15
		<ul style="list-style-type: none"><li>• Connecting Unix Shell : Telnet</li><li>• Login Commands<ul style="list-style-type: none"><li>o passwd, logout, who, who am i, clear,uname</li></ul></li><li>• File / Directory Related Command<ul style="list-style-type: none"><li>o ls, cat, cd, pwd, mv, cp, ln, rm, rmdir, mkdir, chmod, chown, chgrp, find,more,less,head,tail,wc,touch, stat, alias,type</li></ul></li><li>• Operators in Redirection &amp; Piping<ul style="list-style-type: none"><li>o &lt;, &gt;, &lt;&lt;, &gt;&gt;,  </li></ul></li><li>• Finding Patterns in Files<ul style="list-style-type: none"><li>o grep,fgrep,egrep</li></ul></li><li>• Working with columns and fields<ul style="list-style-type: none"><li>o cut,paste,join</li></ul></li></ul>		

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

		<ul style="list-style-type: none"> <li>• Tools for sorting :sort,uniq</li> <li>• Comparing files : cmp,comm,diff</li> <li>• Changing Information in Files : tr,sed,</li> <li>• Examining File Contents : od</li> <li>• Tools for mathematical calculations: bc,factor</li> <li>• Monitoring Input and Output :tee,script</li> <li>• Tools For Displaying Date and Time : cal,date</li> <li>• Co</li> <li>• mmunications : telnet,wall,write,mail,finger,mesg, ping</li> <li>• Process Related Commands :</li> <li>• ps, command to run process in background, nice,kill,at,batch,wait,sleep,top,jobs</li> <li>• Concept of Mounting a File System : mount command</li> <li>• Concept of DeMounting a File System : umount command</li> </ul>		
4	Text Editing With vi and nano Editor, Shell Programming	<ul style="list-style-type: none"> <li>• Introduction of vi editor</li> <li>• Modes in vi</li> <li>• Switching mode in vi</li> <li>• Cursor movement</li> <li>• Screen control commands</li> <li>• Entering text, cut, copy, paste in vi editor</li> <li>• Introduction of nano editor</li> </ul>	20	08
		<ul style="list-style-type: none"> <li>• Shell Keywords</li> <li>• Shell Variables</li> <li>• System variables <ul style="list-style-type: none"> <li>○ PS2, PATH, HOME,LOGNAME, MAIL,IFS, SHELL, TERM, MAILCHECK</li> </ul> </li> <li>• User variables <ul style="list-style-type: none"> <li>○ set, unset and echo command with shell variables</li> </ul> </li> <li>• Positional Parameters</li> <li>• Interactive shell script using read and echo</li> <li>• Decision Statements <ul style="list-style-type: none"> <li>○ if then fi</li> <li>○ if then else fi</li> </ul> </li> </ul>		

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

		<ul style="list-style-type: none"> <li>○ if then elif else fi</li> <li>○ case esac</li> <li>• test command</li> <li>• Logical Operators</li> <li>• Looping statements <ul style="list-style-type: none"> <li>o for loop</li> <li>o while loop</li> <li>o until loop</li> <li>o break, continue command</li> </ul> </li> <li>• Array</li> <li>• Function</li> <li>• Various shell script examples</li> </ul>		
5	Getting Started with Linux, Linux Booting, Linux Admin (Ubuntu)	<ul style="list-style-type: none"> <li>• History of Linux</li> <li>• GNU, GPL Concept</li> <li>• Open Source &amp; Freeware</li> <li>• Structure and Features of Linux</li> <li>• Installation and Configuration of Linux <ul style="list-style-type: none"> <li>○ Using with Ubuntu</li> </ul> </li> <li>• Startup, Shutdown and boot loaders of Linux</li> </ul>	20	07
		<ul style="list-style-type: none"> <li>• Linux Booting Process <ul style="list-style-type: none"> <li>○ LILO Configuration</li> <li>○ GRUB Configuration</li> </ul> </li> </ul>		
		<ul style="list-style-type: none"> <li>• Creating Linux User Account and Password</li> <li>• Installing and Managing Samba Server</li> <li>• Installing and Managing Apache Server</li> <li>• Optimizing LDAP Services</li> <li>• Optimizing DNS Services</li> <li>• Optimizing FTP Services</li> <li>• Optimizing Web Services</li> <li>• Configure Ubuntu's Built-In Firewall</li> <li>• Working with WINE</li> </ul>		

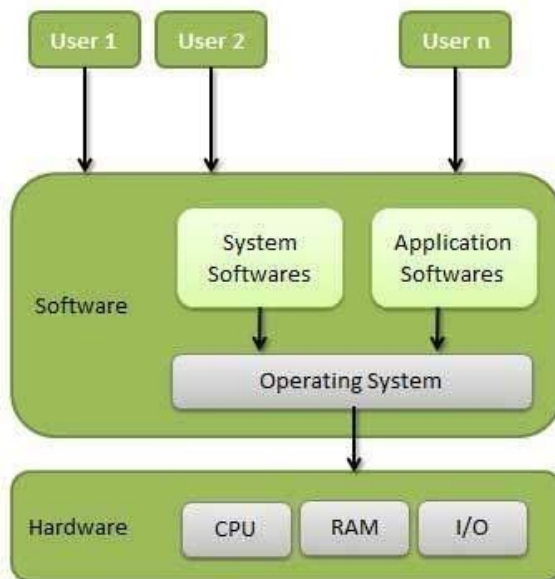
## UNIT – 1 (Part-1)

### Introduction, Process and Thread, Process Scheduling

Topic: 1      What is Operating System?

Mark 1

Ans.



Detailed:

- **Definition 1:**  
An operating system (OS), in its most general sense, is software that allows a user to run other applications on a computing device.
- **Definition 1:**  
Operating system is an interface between user and hardware.  
Operating system is a system application where user can run their applications and manage hardware on it.

#### One word question answer

##### Question

1. OS means?
2. What is Operating System?

##### Answer

Operating System  
Interface between user and hardware

**Topic: 2**      **Explain types of an operating system.**

**Mark 3**

**Ans.**

**Detailed:**

Here some types of an operating systems are:

- Batch Operating System
- Multitasking/Time Sharing OS
- Real Time OS
- Distributed OS
- Network OS

## **1. Batch Operating System**

- Some computer processes are very lengthy and time-consuming. To speed the same process, a job with a similar type of needs is batched together and run as a group.
- The user of a batch operating system never directly interacts with the computer.
- In this type of OS, every user prepares his or her job on an offline device like a punch card and submit it to the computer operator.

## **2. Multi-Tasking/Time-sharing Operating systems**

- Time-sharing operating system enables people located at a different terminal(shell) to use a single computer system at the same time.
- The processor time (CPU) which is shared among multiple users is termed as time sharing.

## **3. 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.

## **4. Distributed Operating System**

- Distributed systems use many processors located in different machines to provide very fast computation to its users.

## **5. Network Operating System**

- Network Operating System runs on a server.
- It provides the capability to serve to manage data, user, groups, security, application, and other networking functions.

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## One word question answer

### Question

1. Which operating system never directly interacts with the computer?
2. The processor time (CPU) which is shared among multiple users is known?
3. Which Operating System runs on a server?

### Answer

Batch operating system  
Time sharing operating system  
Network operating system

**Topic: 3 Explain features of an operating system.**

**Mark 3**

**Ans.**

### Detailed:

Here is a list commonly found important features of an Operating System:

#### 1. Security Management

- The information and confidential data stored in the system is protected by the operating system and block the system by giving strong authorized keys to the user securing the system from malware attack.
- The operating system acts as supervisor mode and gives the system a strongly protected firewall. OS handle and rectify the errors as soon as possible without any difficulty.

#### 2. Process Management

- The program execution is managed effectively by the operating system without any overlapping or time delay.
- The management of the process supports OS to develop and eliminate process and provides mechanism form communication and synchronization within multiple processes.

#### 3. Storage and Memory Management

- OS performs memory management and virtual memory multitasking.
- The need for memory management in OS is to allocate and de-allocate memory space to process in demand of resources or if they are running out of memory that leads to alerts called file system and disk space is high or full.
- The operating system can do resource allocation and prevent the system from overloading.



# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## 4. Disk Management

- OS permits disk access to manage files systems, file system device drivers and related activities of files like retrieval, naming, sharing, storage and protection of files.

## 5. I/O Operations

- OS can effectively handle I/O operations to hide the odd behavior of hardware from the end-user.

## 6. Device Management

- The device and printer controlling are managed by an operating system by permitting a correct flow.
- Disk management is performed to track all hardware connected to the system and also the resource utilized by different jobs and users.

### One word question answer

#### Question

1. The operating system acts as a\_\_\_\_\_.
2. Operating system can do process management.
3. OS permits disk access to manage \_\_\_\_\_
4. The operating system can do resource allocation and prevent the system from \_\_\_\_\_

#### Answer

supervisor mode  
True  
files systems.  
overloading

**Topic: 4 Explain functions of an operating system.**

**Mark 3**

**Ans.**

#### Detailed:

Following are some of important functions of an operating System.

- I/O System Management
- Secondary-Storage Management
- Security
- Command interpretation
- Networking
- Job accounting
- Communication management

#### 1. I/O System Management:

- One of the main objects of any OS is to hide the specialty of that hardware devices from the user.

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## 2. Secondary-Storage Management:

- Systems have several levels of storage which includes primary storage, secondary storage, and cache storage.
- Instructions and data must be stored in primary storage or cache so that a running program can reference it.

## 3. Security:

- Security module protects the data and information of a computer system against malware threat and authorized access.

## 4. Command interpretation:

- This module is interpreting commands given by the and acting system resources to process that commands.

## 5. Networking:

- A distributed system is a group of processors which do not share memory, hardware devices.
- The processors communicate with one another through the network.

## 6. Job accounting:

- Keeping track of time & resource used by various job and users.

## 7. Communication management:

- Coordination and assignment of compilers, interpreters, and another software resource of the various users of the computer systems.

### One word question answer

#### Question

1. Instructions and data must be stored in\_\_\_\_\_.
2. The\_\_\_\_communicate with one another through the network.

#### Answer

primary storage or cache  
processors

**Topic: 5** Explain operating system with user view and system view.

**Mark 3**

**Ans.**

#### Detailed:

Operating System is designed both by taking user view and system view into consideration.

# GEETANJALI GROUP OF COLLEGES

*(Affiliated to Saurashtra University)*

## User View

- The goal of the Operating System is to maximize the work and minimize the effort of the user.
- Most of the systems are designed to be operated by single user, however in some systems multiple users can share resources, memory. In these cases, Operating System is designed to handle available resources among multiple users and CPU efficiently.
- Operating System must be designed by taking both usability and efficient resource utilization into view.
- In embedded systems (Automated systems) user view is not present.
- Operating System gives an effect to the user as if the processor is dealing only with the current task, but in background processor is dealing with several processes.

## System View

- From the system point of view Operating System is a program involved with the hardware.
- Operating System is allocator, which allocate memory, resources among various processes. It controls the sharing of resources among programs.
- It prevents improper usage, error, and handle deadlock conditions.
- It is a program that runs all the time in the system in the form of Kernel.
- It controls application programs that are not part of Kernel.

## One word question answer

### Question

1. The goal of the operating system is.
2. In which systems user view is not present.

### Answer

maximize the work and minimize the effort of the user.  
Embedded system

## UNIT – 1 (Part-2)

**Topic: 1**      **What is process in operating system?**

**Mark 1**

**Ans.**

**Detailed:**

- 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 put it in simple terms, we write our computer programs in a text file and when we execute this program, it becomes a process which performs all the tasks mentioned in the program.

### One word question answer

#### Question

1. The process is \_\_\_\_\_
2. When we execute computer program, it becomes.

#### Answer

a program in execution.  
A process

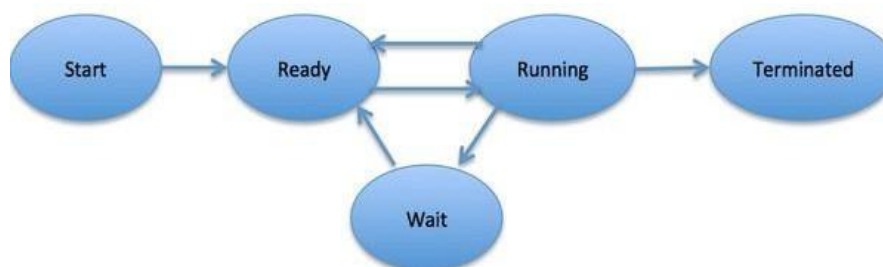
**Topic: 2**      **Explain process life cycle and process state diagram.**

**Mark 3**

**Ans.**

**Detailed:**

### Process State Diagram



### Process Life Cycle

- When a process executes, it passes through different states. These stages may differ in different operating systems, and the names of these states are also not standardized.
- In general, a process can have one of the following five states at a time.

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

1	<b>START</b> THIS IS THE INITIAL STATE WHEN A PROCESS IS FIRST STARTED/CREATED.
2	<b>Ready</b> The process is waiting to be assigned to a processor. Ready processes are waiting to have the processor allocated to them by the operating system so that they can run. Process may come into this state after <b>Start</b> state or while running it by but interrupted by the scheduler to assign CPU to some other process.
3	<b>Running</b> Once the process has been assigned to a processor by the OS scheduler, the process state is set to running and the processor executes its instructions.
4	<b>Waiting</b> Process moves into the waiting state if it needs to wait for a resource, such as waiting for user input, or waiting for a file to become available.
5	<b>Terminated or Exit.</b> Once the process finishes its execution, or it is terminated by the operating system, it is moved to the terminated state where it waits to be removed from main memory.

## One word question answer

### Question

1. Which is the initial state when a process is first started.

### Answer

start

2. Once the process finishes its execution its move on \_\_\_\_\_

exit state

## Topic: 3 Process Control Block.

Mark - 3

Ans.

The **Process Control Block in OS (PCB)** is a data structure used by an operating system to store information about a process. The structure of a PCB may vary depending on the operating system, but it typically contains the following components:

**Process ID (PID):** A unique identifier assigned to each process by the operating system.

**Program counter (PC):** The address of the next instruction to be executed by the process.

**Process state:** The current state of the process, such as running, waiting, or blocked.

**CPU registers:** The values of the processor registers that are being used by the process.

**Memory management information:** Information about the process's memory allocation, such as the base and limit registers.

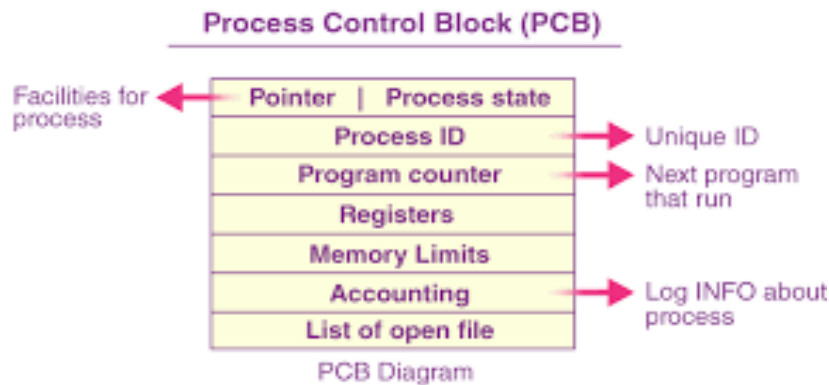
**Input/output (I/O) status information:** Information about the process's I/O operations, such as open files and network connections.

**Accounting information:** Information about the resources used by the process, such as CPU time and memory usage.

**CPU scheduling information:** Information about the process's priority and scheduling requirements.

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)



## Topic: 4 Context Switching.

**Ans.** The Context switching is a technique or method used by the operating system to switch a process from one state to another to execute its function using CPUs in the system. When switching perform in the system, it stores the old running process's status in the form of registers and assigns the **CPU** to a new process to execute its tasks. While a new process is running in the system, the previous process must wait in a ready queue. The execution of the old process starts at that point where another process stopped it. It defines the characteristics of a multitasking operating system in which multiple processes shared the same **CPU** to perform multiple tasks without the need for additional processors in the system.

## Topic: 5 What is Threads ?

Mark 1

### Ans. Detailed:

- A thread refers to a single sequential flow of activities being executed in a process.
- A thread is also called a lightweight process.

## Topic: 6 Concept of Multithread.

Mark 3

### Ans. Detailed:

- A thread is also known as a lightweight process. The idea is to achieve parallelism by dividing a process into multiple threads.
- For example, in a browser, multiple tabs can be different threads. MS Word uses multiple threads: one thread to format the text, another thread to process inputs, etc. More advantages of multithreading are discussed below.
- Multithreading is a technique used in operating systems to improve the performance and responsiveness of computer systems. Multithreading allows multiple threads (i.e., lightweight processes) to share the same resources of a single process, such as the CPU, memory, and I/O devices.

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

**Topic: 7      Benefits of Threads.**

**Mark - 3**

**Ans. Detailed:**

- **Responsiveness:** If the process is divided into multiple threads, if one thread completes its execution, then its output can be immediately returned.
- **Faster context switch:** Context switch time between threads is lower compared to the process context switch. Process context switching requires more overhead from the CPU.
- **Effective utilization of multiprocessor system:** If we have multiple threads in a single process, then we can schedule multiple threads on multiple processors. This will make process execution faster.
- **Resource sharing:** Resources like code, data, and files can be shared among all threads within a process. Note: Stacks and registers can't be shared among the threads. Each thread has its own stack and registers.
- **Communication:** Communication between multiple threads is easier, as the threads share a common address space. while in the process we have to follow some specific communication techniques for communication between the two processes.
- **Enhanced throughput of the system:** If a process is divided into multiple threads, and each thread function is considered as one job, then the number of jobs completed per unit of time is increased, thus increasing the throughput of the system.

**Topic: 8      Types of Threads.**

**Mark - 5**

**Ans. Detailed:**

- **User Level thread (ULT)** – Is implemented in the user level library, they are not created using the system calls. Thread switching does not need to call OS and to cause interrupt to Kernel. Kernel doesn't know about the user level thread and manages them as if they were single-threaded processes.
- **Kernel Level Thread (KLT)** – Kernel knows and manages the threads. Instead of thread table in each process, the kernel itself has thread table (a master one) that keeps track of all the threads in the system. In addition kernel also maintains the traditional process table to keep track of the processes. OS kernel provides system call to create and manage threads.

**Topic: 9      Explain types of process scheduling?**

**Mark 2**

**Ans.**

**Detailed:**

An operating system uses two types of scheduling processes execution, **preemptive** and **non-preemptive**.

**1. Preemptive process:**

- In preemptive scheduling policy, a low priority process must be suspended its execution if high priority process is waiting in the same queue for its execution.

**2. Non - Preemptive process:**

- In non - preemptive scheduling policy, processes are executed in first come first serve basis, which means the next process is executed only when currently running process finishes its execution.

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

Operating system performs the task of scheduling processes based on priorities using these following algorithms:

## 1) First come first serve (FCFS)

In this scheduling algorithm the first process entered in queue is processed first.

## 2) Shortest job first (SJF)

In this scheduling algorithm the process which requires shortest CPU time to execute is processed first.

## 3) Priority scheduling

In this scheduling algorithm the priority is assigned to all the processes and the process with highest priority executed first. Priority assignment of processes is done based on internal factor such as CPU and memory requirements or external factor such as user's choice. The priority scheduling algorithm supports preemptive and non - preemptive scheduling policy.

## 4) Round Robin (RR) scheduling

In this algorithm the process is allocated the CPU for the specific time called time slice, which is normally of 10 to 100 milliseconds. If the process completes its execution within this time slice, then it is removed from the queue otherwise it must wait for another time slice.

### One word question answer

#### Question

1. In which process scheduling type processes are executed in first come first serve basis.
2. In which process scheduling the process with highest priority executed first.

#### Answer

Non - Preemptive process  
Priority scheduling

## UNIT – 1 (Part-3)

**Topic: 1 Explain First Come First Serve (FCFS) process scheduling algorithm.**

**Mark 5**

**Ans.**Detailed:

- In the "First come first serve" scheduling algorithm, as the name suggests, the process which arrives first, gets executed first, or we can say that the process which requests the CPU first, gets the CPU allocated first.

- ☐ This is used in Batch Systems. A perfect real-life example of FCFS scheduling is buying tickets at ticket counter.Example:

**Suppose there are four processes with process ID's P1, P2, P3, and P4 and they enter into the CPU as follows:**

PROCESS ID	ARRIVAL TIME (MILLISECONDS)	BURST TIME (MILLISECONDS)
P1	0	5
P2	2	3
P3	6	2
P4	7	3

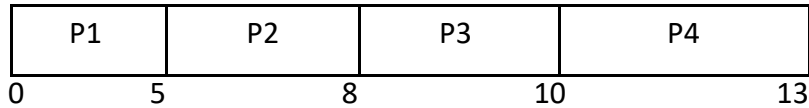


# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

- **Step 1:**

**Gant Chart:**



- **Step 2:**

Turn Around Time = Completion Time – Arrival Time

Waiting Time = Turn Around Time – Burst Time

P ID	Arrival Time	Burst Time	Completion time (milliseconds)	Turn Around Time (milliseconds)	Waiting Time (milliseconds)
P1	0	5	5	5	0
P2	2	3	8	6	3
P3	6	2	10	4	2
P4	7	3	13	6	3

- **Step 3:**

**Avg. Turn Around Time = Total Turn Around Time/No. of Process**

$(5+6+4+6)/4 = 21/4 = 5.25 \text{ ms}$

**Avg. Waiting Time = Total Waiting Time/No. of Process**

$(0+3+2+3)/4 = 8/4 = 2 \text{ ms}$

- **Step 4:**

**Throughput = Total Burst Time/No. of Process**

$(5+3+2+3)/4 = 13/4 = 3.25 \text{ ms}$

- **Step 5:**

**CPU Utilization = Total Estimated Time/Total Estimated Time \* 100**

$13/13*100 = 100\%$

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

**Topic: 2** Explain Shortest Job First (SJF) process scheduling algorithm.

**Mark 5**

**Ans.**

**Detailed:**

- Shortest job first is a scheduling algorithm in which the process with the smallest execution time is selected for execution next.
- Shortest job first can be either preemptive or non-preemptive.

**Example:**

Suppose there are four processes with process ID's **P1**, **P2**, **P3**, and **P4** and they enter the CPU as follows:

PROCESS ID	ARRIVAL TIME (MILLISECONDS)	BURST TIME (MILLISECONDS)
P1	0	5
P2	2	3
P3	3	2
P4	1	3

- **Step 1:**

**Gant Chart:**

P1	P3	P4	P2	
0	5	7	10	13

- **Step 2:**

Turn Around Time = Completion Time – Arrival Time

Waiting Time = Turn Around Time – Burst Time

PID	ARRIVAL TIME	BURST TIME	COMPLETION TIME	TURN AROUND TIME	WAITING TIME
P1	0	5	5	5	0
P2	2	3	7	5	2
P3	3	2	10	7	5
P4	1	3	13	12	9

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

- **Step 3:**

**Avg. Turn Around Time = Total Turn Around Time/No. of Process**

$$(5+5+7+12)/4 = 29/4 = 7.25 \text{ ms}$$

**Avg. Waiting Time = Total Waiting Time/No. of Process**

$$(0+2+5+9)/4 = 16/4 = 4 \text{ ms}$$

- **Step 4:**

**Throughput = Total Burst Time/No. of Process**

$$(5+3+2+3)/4 = 13/4 = 3.25 \text{ ms}$$

- **Step 5:**

**CPU Utilization = Total Estimated Time/Total Estimated Time \* 100**

$$13/13 * 100 = 100\%$$

## One word question answer

### Question

1. In which process scheduling the smallest Execution time is selected for execution next.
2. Formula of avg. turnaround time.
3. Formula of avg. waiting time.

### Answer

SJF  
Total Turn Around Time/No. of Process  
Total Waiting Time/No. of Process

**Topic: 3 Explain Round Robin process scheduling algorithm.**

**Mark 5**

**Ans.**

### Detailed:

- Round Robin is the preemptive process scheduling algorithm.
- Each process is provided a fix time to execute, it is called a **quantum**.
- Once a process is executed for a given time period, it is preempted, and other process executes for a given time period.

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## Example:

Suppose there are four processes with process ID's **P1**, **P2**, **P3**, and **P4** and they enter the CPU as follows:

PROCESS ID	ARRIVAL TIME	BURST TIME
P1	0	5 (2)
P2	2	3
P3	3	2
P4	1	3

**Quantum: 3**

## Step 1:

Gant Chart:

P4 P2 P3 P1					
P1	P4	P2	P3	P1	
0	3	6	9	11	13

## Step 2:

Turn Around Time = Completion Time – Arrival Time

Waiting Time = Turn Around Time – Burst Time

PID	ARRIVAL TIME	BURST TIME	COMPLETION TIME	TURN AROUND TIME	WAITING TIME
P1	0	5	13	13	8
P2	2	3	9	7	4
P3	3	2	11	8	6
P4	1	3	6	5	2

## Step 3:

**Avg. Turn Around Time = Total Turn Around Time/No. of Process**

$$(13+7+8+5)/4 = 33/4 = 8.25$$

**Avg. Waiting Time = Total Waiting Time/No. of Process**

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

$$(8+4+6+2)/4 = 20/4 = 5$$

**Step 4:**

**Throughput = Total Burst Time/No. of Process**

$$(5+3+2+3)/4 = 13/4 = 3.25$$

**Step 5:**

**CPU Utilization = Total Estimated Time/Total Estimated Time \* 100**

$$13/13*100 = 100\%$$

## One word question answer

### Question

1. Round Robin is which type of process scheduling algorithm?
2. Each process is provided a fix time to execute, it is called a\_\_\_\_\_.

### Answer

preemptive  
quantum

**Topic: 4      Explain Priority process scheduling algorithm with non-preemptive method.      Mark 5**

**Ans.**

### Detailed:

- Priority scheduling is one of the most common scheduling algorithms in batch systems.
- Each process is assigned a priority. Process with the highest priority is to be executed first and so on.
- Processes with the same priority are executed on first come first served basis.
- Priority can be decided based on memory requirements, time requirements or any other resource requirement.
- Priority Scheduling can be either preemptive or non-preemptive.

**Example:**

Suppose there are four processes with process ID's **P1, P2, P3, and P4** and they enter the CPU as follows:

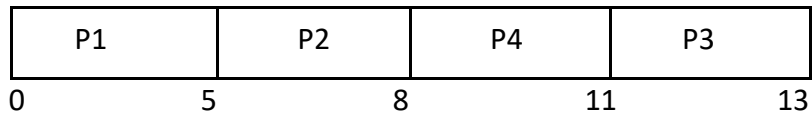
PROCESS ID	ARRIVAL TIME	BURST TIME	PRIORITY
P1	0	5	3
P2	2	3	5
P3	3	2	1
P4	1	3	2

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## Step 1:

Gant Chart:



## Step 2:

Turn Around Time = Completion Time – Arrival Time

Waiting Time = Turn Around Time – Burst Time

PID	ARRIVAL TIME	BURST TIME	COMPLETION TIME	TURN AROUND TIME	WAITING TIME
P1	0	5	5	5	0
P2	2	3	8	6	3
P3	3	2	13	10	8
P4	1	3	11	10	7

## Step 3:

**Avg. Turn Around Time = Total Turn Around Time/No. of Process**

$$(5+6+10+10)/4 = 31/4 = 7.75$$

**Avg. Waiting Time = Total Waiting Time/No. of Process**

$$(0+3+8+7)/4 = 18/4 = 4.5$$

## Step 4:

**Throughput = Total Burst Time/No. of Process**

$$(5+3+2+3)/4 = 13/4 = 3.25$$

## Step 5:

**CPU Utilization = Total Estimated Time/Total Estimated Time \* 100**

$$13/13*100 = 100\%$$

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## One word question answer

### Question

1. Priority Scheduling can be \_\_\_\_\_.
2. Each process is assigned a \_\_\_\_\_.

### Answer

preemptive or non-preemptive  
priority.

**Topic: 5**      **Explain Priority process scheduling algorithm with preemptive method.**

**Mark 5**

**Ans.**

**Detailed:**

- Priority scheduling is one of the most common scheduling algorithms in batch systems.
- Each process is assigned a priority. Process with the highest priority is to be executed first and so on.
- Processes with the same priority are executed on first come first served basis.
- Priority can be decided based on memory requirements, time requirements or any other resource requirement.
- Priority Scheduling can be either preemptive or non-preemptive.

**Example:**

Suppose there are four processes with process ID's **P1, P2, P3**, and **P4** and they enter the CPU as follows:

PROCESS ID	ARRIVAL TIME	BURST TIME	PRIORITY
P1	0	5(3)	3
P2	2	3	5
P3	3	2	1
P4	1	3	2

**Step 1:**

Gant Chart:

P1	P2	P1	P4	P3	
0	2	5	8	11	13

**Step 2:**

Turn Around Time = Completion Time – Arrival Time

Waiting Time = Turn Around Time – Burst Time

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

PID	ARRIVAL TIME	BURST TIME	COMPLETION TIME	TURN AROUND TIME	WAITING TIME
P1	0	5	8	8	3
P2	2	3	5	3	0
P3	3	2	13	10	8
P4	1	3	11	10	7

## Step 3:

**Avg. Turn Around Time = Total Turn Around Time/No. of Process**

$$(8+3+10+10)/4 = 31/4 = 7.75$$

**Avg. Waiting Time = Total Waiting Time/No. of Process**

$$(3+0+8+7)/4 = 18/4 = 4.5$$

## Step 4:

**Throughput = Total Burst Time/No. of Process**

$$(5+3+2+3)/4 = 13/4 = 3.25$$

## Step 5:

**CPU Utilization = Total Estimated Time/Total Estimated Time \* 100**

$$13/13*100 = 100\%$$

## One word question answer

### Question

1. Processes with the same priority are executed on\_\_\_\_\_.
2. Formula to find throughput is\_\_\_\_\_.

### Answer

first come first served basis  
Total Burst Time/No. of Process



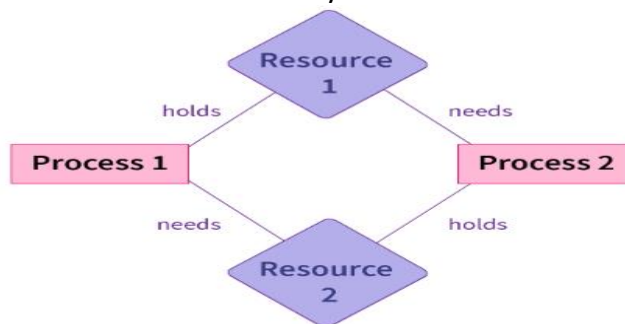
## UNIT – 2

### Deadlocks Memory Management

#### Topic: 1 Deadlocks: Definition

Mark - 1

- All the processes in a system require some resources such as central processing unit(CPU), file storage, input/output devices, etc to execute it. Once the execution is finished, the process releases the resource it was holding. However, when many processes run on a system they also compete for these resources they require for execution. This may arise a deadlock situation.



#### Topic : 2 Deadlock Prevention

Mark - 1

**Deadlock prevention:** The possibility of deadlock is excluded before making requests, by eliminating one of the necessary conditions for deadlock.

**Example:** Only allowing traffic from one direction, will exclude the possibility of blocking the road.

#### Topic : 3 Deadlock Avoidance

Mark - 1

**Deadlock avoidance:** The Operating system runs an algorithm on requests to check for a safe state. Any request that may result in a deadlock is not granted.

**Example:** Checking each car and not allowing any car that can block the road. If there is already traffic on the road, then a car coming from the opposite direction can cause blockage.

#### Topic : 4 Deadlock Detection

Mark - 1

**Deadlock detection & recovery:** OS detects deadlock by regularly checking the system state, and recovers to a safe state using recovery techniques.

**Example:** Unblocking the road by backing cars from one side. Deadlock prevention and deadlock avoidance are carried out before deadlock occurs.

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## UNIT – 2 (Part-2)

**Topic: 1** Explain physical memory and virtual memory.

**Mark 2**

**Ans.**

**Detailed:**

### Physical Memory:

- Physical memory is the actual real memory used in RAM.
- Physical memory is the only memory that is directly accessible to the CPU.
- CPU reads the instructions stored in the physical memory and executes them continuously.
- The data that is operated will also be stored in physical memory in uniform manner.

### Virtual Memory:

- Virtual memory as the name suggests is not real.
- The OS uses virtual memory as a memory management technique in which non-contiguous memory is presented to software as contiguous memory.
- Virtual memory is one classification of memory which was created by using the hard disk for simulating additional RAM, the addressable space available for the user.
- Virtual addresses are mapped into real addresses.

### One word question answer

#### Question

1. What is physical memory?
2. Data stored in physical memory in \_\_\_\_\_.
3. Virtual addresses are mapped into \_\_\_\_\_.

#### Answer

actual real memory  
uniform manner  
real addresses.

**Topic: 2** What is memory allocation?

**Mark 2**

**Ans.**

**Detailed:**

- Memory allocation is a process by which computer programs and services are assigned with physical or virtual memory space.

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

- Memory allocation is the process of reserving a partial or complete portion of computer memory for the execution of programs and processes.
- Memory allocation is achieved through a process known as memory management.
- Memory allocation is primarily a computer hardware operation but is managed through operating system and software applications.
- Memory allocation process is quite similar in physical and virtual memory management.

**Memory allocation has two core types.**

- **Static Memory Allocation:** The program is allocated memory at compile time.
- **Dynamic Memory Allocation:** The programs are allocated with memory at run time.

## One word question answer

### Question

1. Memory allocation is the process of\_\_\_\_\_.
2. The program is allocated memory at compile time in\_\_\_\_\_.
3. The programs are allocated with memory at run time in\_\_\_\_\_.

### Answer

assigning space to memory  
Static Memory Allocation  
Dynamic Memory Allocation

**Topic: 3      What is fragmentation?**

**Mark 3**

**Ans.**

### Detailed:

- In a computer storage system, as processes are loaded and removed from memory, the free memory space is broken into small pieces.
- In this way memory space used inefficiently, so the capacity or performance of the system may degrade.
- The conditions of the fragmentation depend on the system of memory allocation. In most of the cases, memory space is wasted.
- Sometimes it happens that memory blocks cannot be allocated to processes due to their small size and memory blocks remain unused. This problem is known as fragmentation.

### Types of fragmentation

1. External fragmentation
2. Internal fragmentation

#### 1) External fragmentation

External fragmentation exists when total memory space exists to satisfy a request, but it is not contiguous; storage is fragmented into many holes.

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## 2) Internal fragmentation

An approach is to allocate small holes as part of the larger request. Thus, the allocated memory may be larger than the requested memory.

**Topic: 4 Explain contiguous memory allocation?**

**Mark 3**

**Ans.**

**Detailed:**

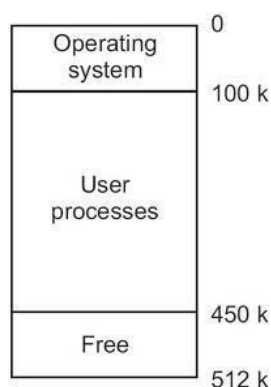
- Contiguous memory allocation is a classical memory allocation model that assigns a process consecutive memory blocks (that is, memory blocks having consecutive addresses).
- Contiguous memory allocation is one of the oldest memory allocation schemes.
- When a process needs to execute, memory is requested by the process. The size of the process is compared with the amount of contiguous main memory available to execute the process. If sufficient contiguous memory is found, the process is allocated memory to start its execution.
- Otherwise, it is added to a queue of waiting processes until sufficient free contiguous memory is available.

There are different types of contiguous memory allocation in operating system.

1. **Single partition allocation**
2. **Multiple Partition allocation**
3. **Dynamic partition allocation**

### 1. Single Partition Allocation

In this scheme Operating system is residing in low memory and user processes are executing in higher memory.



### Advantages

- It is simple.
- It is easy to understand and use.

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## Disadvantages

- Poor utilization of processor and memory.
- Users job is limited to the size of available memory.

## 2. Multiple Partition Allocation

User memory is divided into more than one partitions. There are two types fixed partition or static partition and variable partition or dynamic partition.

- **Fixed partition** has equal or unequal size of partitions and a partition is allocated to an active process in the multiprogramming system, however it may suffer from internal and external fragmentations, it supports only fixed number of active processes.
- **Variable partition** has equal or unequal size of partitions and these partitions are created dynamically. A block or hole is available memory or free partitions. Variable partition has no internal fragmentation and its better utilization of memory than fixed partition. However, variable partition suffers from external fragmentation and it has overhead of compaction.

## 3. Dynamic Allocation Algorithms

If more than one empty space or blocks are available for an active process, then the operating system decides how to select a best space or block using dynamic allocation algorithms. These algorithms are as following:

- **First fit:** It allocated the first hole that is big enough to process, this started from the beginning to search such empty hole, so first fit algorithm is simple to implement and faster to use.
- **Best fit:** This allocates smallest hole that is big enough for process. This is fast when all empty blocks or holes sorted in increasing order of size. Best fit minimizes internal fragmentation.
- **Worst fit:** This allocates largest hole that is empty for process. This creates maximum internal fragmentation. Worst fit is fast when all empty blocks or holes sorted in decreasing order of size.

### One word question answer

#### Question

1. Which is the oldest memory allocation schemes?
2. Which allocation assigns a process consecutive memory block?
3. Where user memory is divided into more than one partitions.

#### Answer

Contiguous memory allocation  
Contiguous memory allocation  
Multiple Partition Allocation

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

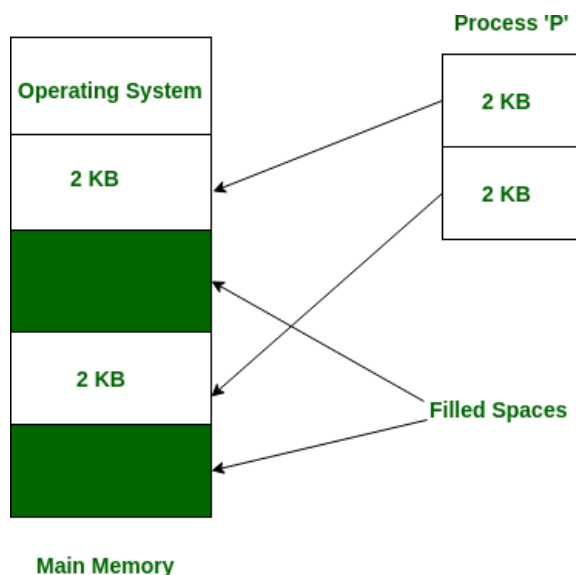
Topic: 5 Explain noncontiguous memory allocation?

Mark 2

Ans.

**Detailed:**

- In the non-contiguous memory allocation, a process will acquire the memory space, but it is not at one place it is at the different locations according to the process requirement.
- This technique of non-contiguous memory allocation reduces the wastage of memory which leads to internal and external fragmentation.
- This utilizes all the free memory space which is created by a different process.
- In this example, process P can be divided into two parts of equal size – 2KB.
- Hence one part of process P can be allocated to first 2KB space of main memory and other part of process P can be allocated to second 2KB space of main memory.
- Below diagram will explain in better way:



There are two types of noncontiguous memory allocation.

1. **Paging**
2. **Segmentation**

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## One word question answer

### Question

1. Which memory allocation reduces wastage of memory.
2. Which allocation a process will acquire the memory space at the different locations?
3. Which are two types of noncontiguous memory allocation?

### Answer

Noncontiguous memory allocation  
Noncontiguous memory allocation  
Paging & Segmentation

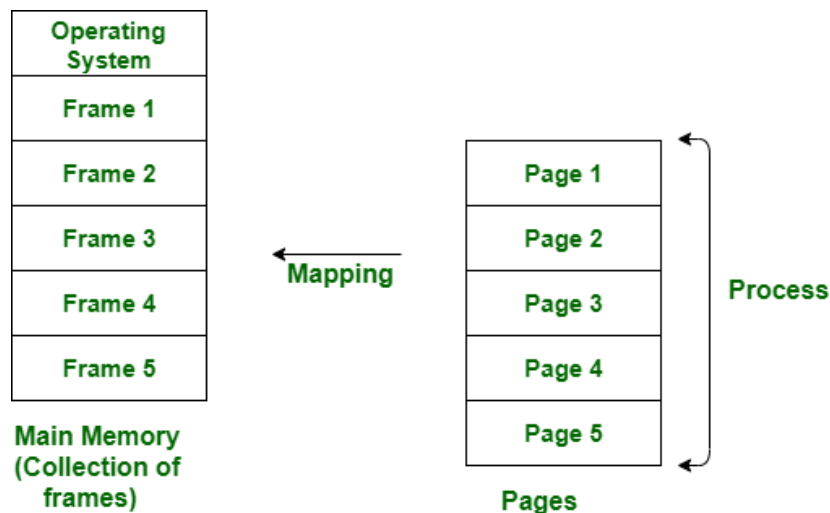
Topic: 6 Explain virtual memory using paging.

Mark 3

Ans.

### Detailed:

- Paging is a method or techniques which is used for non-contiguous memory allocation.
- It is a fixed size partitioning theme (scheme).
- In paging, both main memory and secondary memory are divided into equal fixed size partitions.
- **Page:** A fixed-length contiguous block of virtual memory residing on disk.
- **Frame:** A fixed-length contiguous block located in RAM; whose sizing is identical to pages.



## One word question answer

### Question

1. Which is a fixed size partitioning theme?
2. Paging is a method or techniques which is used for?

### Answer

Paging  
Non-contiguous memory allocation

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

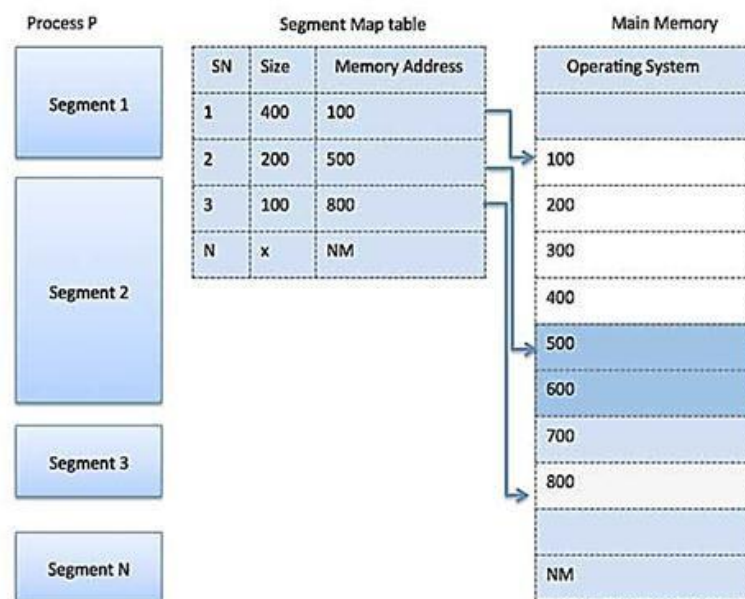
Topic: 7 Explain virtual memory using segmentation.

Mark 3

Ans.

Detailed:

- The process known as segmentation is a virtual process that creates address spaces of various sizes in a computer system, called segments.
- Each segment is a different virtual address space that directly corresponds to process objects.
- When a process executes, segmentation assigns related data into segments for faster processing.
- The segmentation function maintains a segment table that includes physical addresses of the segment, size, and other data.



## One word question answer

### Question

1. The segmentation function maintains a \_\_\_\_\_
2. Segment table contains \_\_\_\_\_

### Answer

segment table  
physical addresses, size, & other data



# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

Topic: 8 Explain difference between paging and segmentation.

Mark 3

Ans.

Detailed:

	Paging	Segmentation
1.	IN PAGING, PROGRAM IS DIVIDED INTO FIXED OR MOUNTED SIZE PAGES.	IN SEGMENTATION, PROGRAM IS DIVIDED INTO VARIABLE SIZE SECTIONS.
2.	For paging operating system is accountable.	For segmentation compiler is accountable.
3.	Page size is determined by hardware.	Here, the section size is given by the user.
4.	It is faster in the comparison of segmentation.	Segmentation is slow.
5.	Paging could result in internal fragmentation.	Segmentation could result in external fragmentation.
6.	In paging, logical address is split into page number and page offset.	Here, logical address is split into section number and section offset.

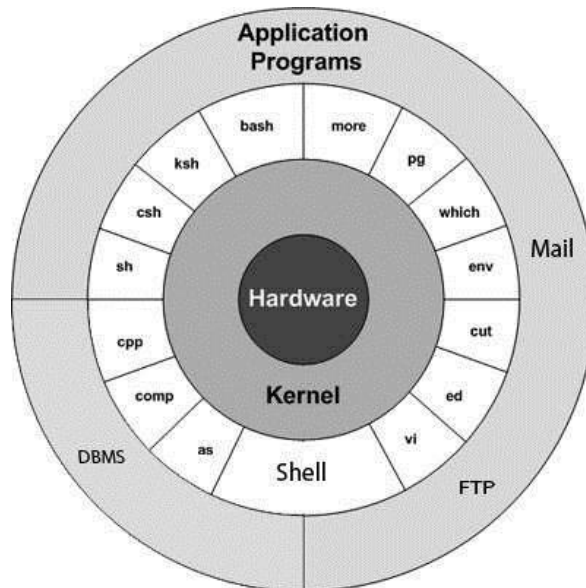
## UNIT – 3 (Part-1)

### Getting Started with Unix, Unix Shell Command

Topic: 1 Explain UNIX Architecture.

Mark 5

Ans.



#### Detailed:

- The UNIX operating system is a set of programs that act as a link between the computer and the user.
- UNIX was originally developed in 1969 by a group of AT&T employees Ken Thompson, Dennis Ritchie at Bell Labs.
- The computer program that allocates the system resources and coordinates all the details of the computer's internal is called the 'operating system' or the 'kernel'.
- Users communicate with the kernel through a program known as the **shell**.
- The shell is a command line interpreter; it translates commands entered by the user and converts them into a language that is understood by the kernel.

The main concept that unites all the versions of Unix is the following four basics –

- **Kernel**

- The kernel is the heart of the operating system.
- It interacts with the hardware and most of the tasks like memory management, task scheduling and file management.

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

- **Shell**

- The shell is the utility that processes your requests.
- When you type in a command at your terminal, the shell interprets the command and calls the program that you want.
- C Shell, Bourne Shell and Korn Shell are the most famous shells which are available with most of the Unix variants.

- **Commands and Utilities**

- There are various commands and utilities which you can make use of in your day-to-day activities.
- **cp, mv, cat** and **grep**, etc. are few examples of commands and utilities.
- There are over 250 standard commands plus numerous others provided through 3<sup>rd</sup> party software.
- All the commands come along with various options.

- **Files and Directories**

- All the data of Unix is organized into files.
- All files are then organized into directories.
- These directories are further organized into a tree-like structure called the **filesystem**.

## One word question answer

### Question

3. UNIX was originally developed in \_\_\_\_.
4. Unix was developed by.
5. How many commands in Unix?

### Answer

- 1969  
Ken Thompson, Dennis Ritchie  
250

**Topic: 2**      **Explain UNIX features.**

**Mark 3**

**Ans.**

**Detailed:**

- The following are the Unix Features.

**Portability:**

- The system is written in high-level language making it easier to read, understand, change and therefore move to other machines.
- The code can be changed and compiled on a new machine.
- Customers can then choose from a wide variety of hardware vendors without being locked in with a particular vendor.

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## Machine-independence:

- The System hides the machine architecture from the user, making it easier to write applications that can run on micros and mainframes.

## Multi-Tasking:

- UNIX is a powerful multi-tasking operating system; it means when a active task in process, there can be a simultaneous background process working too.
- UNIX handles these active and background threads efficiently and manages the system resources in a fair-share manner.

## Multi-User Operations:

- UNIX is a multi-user system designed to support a group of users simultaneously.
- The system allows for the sharing of processing power and peripheral resources, while at the same time providing excellent security features.

## Hierarchical File System:

- UNIX uses a hierarchical file structure to store information.
- This structure has the maximum flexibility in grouping information in a way that reflects its natural state. It allows for easy maintenance and efficient implementation.

## UNIX shell:

- UNIX has a simple user interface called the shell that has the power to provide the services that the user wants.
- It protects the user from having to know the intricate hardware details.

### One word question answer

#### Question

1. Write any 2 features of Unix.

#### Answer

Multi-Tasking, Unix Shell

**Topic: 3**      **Explain types of shell in UNIX.**

**Mark 3**

**Ans.**

#### Detailed:

- A Shell provides you with an interface to the **Unix** system.
- It gathers input from you and executes programs based on that input.
- When a program finishes executing, it displays that program's output.
- There are different shells available in **UNIX**.

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## 1. The Bourne Shell

- The **Bourne shell (sh)**, written by **Steve Bourne** at **AT&T Bell Labs**, is the original UNIX shell.
- It is the preferred shell for shell programming because of its compactness and speed.
- Its command full-path name is **/bin/sh**. User default prompt is **\$**.

## 2. The C Shell

- The **C shell (csh)**, written by **Bill Joy** at the **University of California at Berkeley**.
- Includes convenient programming features, such as built-in arithmetic syntax.
- Its command full-path name is **/bin/csh**. User default prompt is **hostname %**.

## 3. The Korn Shell

- The **Korn shell (ksh)**, written by **David Korn** at **AT&T Bell Labs**
- It supports everything in the Bourne shell. Have interactive features comparable to those in the C shell.
- It is faster than the C shell.
- Its command full-path name is **/bin/ksh**. User default prompt is **\$**.

## 4. The Bourne-Again Shell

- The **Bourne-Again shell (bash)** is compatible to the Bourne shell.
- It use features from the Korn and C shells.
- Have arrow keys that are automatically mapped for command recall and editing.
- It's Command full-path name is **/bin/bash**.

### One word question answer

#### Question

5. The Bourne shell written by.
6. The C shell written by.
7. The Korn shell written by.

#### Answer

Steve Bourne  
Bill Joy  
David Korn

**Topic: 4 Explain UNIX file system.**

**Mark 3**

**Ans.**

#### Detailed:

- The UNIX file system is a methodology for logically organizing and storing large quantities of data such that the system is easy to manage.
- A file can be informally defined as a collection of data. A file is the smallest unit of storage in the UNIX file system.
- The following system directories are present in most UNIX file systems:

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

- / - the root users' home directory.
- /bin - short for binaries, this is the directory where many commonly used executable commands reside
- /dev - contains device specific files
- /etc - contains system configuration files
- /lib - contains all library files
- /mnt - contains device files related to mounted devices
- /proc - contains files related to system processes
- /sbin - system binary files reside here. If there is no sbin directory on your system, these files most likely reside in etc
- /tmp - storage for temporary files which are periodically removed from the filesystem
- /usr - also contains executable commands.

## One word question answer

### Question

3. Which directory contains device specific files?

### Answer

/dev

4. Which directory contains system configuration files processors?

/etc

Topic: 5 Explain types of files in UNIX.

Mark 2

Ans.

### Detailed:

- The UNIX file system contains several different types of files:
- **Ordinary Files**
  - Used to store your information, such as some text you have written.
  - Always located within/under a directory file.
  - Do not contain other files.
- **Directories**
  - Branching points in the hierarchical tree.
  - Used to organize groups of files.
  - May contain ordinary files, special files or other directories.
- **Special Files**
  - Used to represent a real physical device such as a printer, tape drive or terminal, used for Input/output (I/O) operations.

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## One word question answer

### Question

3. How many file types in unix?

### Answer

3

**Topic: 6 Explain UNIX file and directory permissions.**

**Mark 3**

**Ans.**

**Detailed:**

- File ownership is an important component of Unix that provides a secure method for storing files. Every file in Unix has the following attributes –
  - **Owner permissions** – The owner's permissions determine what actions the owner of the file can perform on the file.
  - **Group permissions** – The group's permissions determine what actions a user, who is a member of the group that a file belongs to, can perform on the file.
  - **Other (world) permissions** – The permissions for others indicate what action all other users can perform on the file.

### The Permission Indicators

- While using **ls -l** command, it displays various information related to file permission as follows –

```
$ls -l /home/abc
-rwxr-xr-- 1 abc  users 1024 Nov 2 00:10 myfile
drwxr-xr-- 1 abc  users 1024 Nov 2 00:10 mydir
```

- Here, the first column represents different access modes, i.e., the permission associated with a file or a directory.
- The permissions are broken into groups of threes, and each position in the group denotes a specific permission, in this order: read (r), write (w), execute (x) –
  - The first three characters (2-4) represent the permissions for the file's owner. For example, **-rwxr-xr--** represents that the owner has read (r), write (w) and execute (x) permission.
  - The second group of three characters (5-7) consists of the permissions for the group to which the file belongs. For example, **-rwxr-xr--** represents that the group has read (r) and execute (x) permission, but no write permission.
  - The last group of three characters (8-10) represents the permissions for everyone else. For example, **-rwxr-xr--** represents that there is **read (r)** only permission.

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## File/Directory Access Modes

- The permissions of a file are the first line of defense in the security of a Unix system.
- The basic building blocks of Unix permissions are the **read**, **write**, and **execute** permissions, which have been described below –

### Read

Grants the capability to read, i.e., view the contents of the file.

### Write

Grants the capability to modify, or remove the content of the file.

### Execute

User with execute permissions can run a file as a program.

### One word question answer

#### Question

1. How many types of permissions in unix?
4. Access modes in unix?

#### Answer

3  
Read, write, execute

## UNIT – 3 (Part-2)

**Topic: 1**      **Unix Commands.**

**Ans.**

**Detailed:**

Commands	
Login Commands	Passwd, logout, who, who m i, clear, uname
File & Directory related command	Ls, cat, cd, pwd, mv, cp, ln, rm, rmdir, mkdir, umask, chmod, chown, chgrp, find, pg, more, less, head, tail, wc, touch, stat, alias, type,



# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

Operator & redirecting piping	< , > , << , >> ,
Finding pattern in file	Grep, fgrep, egrep
Working with Column and fields	Sort, uniq
Tools for sorting	
Comparing File	Cmp, comm, diff
Changing information in file	Tr, sed
Examining file content	Od
Tools for mathematical calculation	Bc, factor
Tools for display date and time	Cal, date
Communication	telnet, wall, motd, write, mail, news, finger, mesg, ping
Process related commands	Ps, nice, kill, at, batch, cron, crontab, wait, sleep, top, jobs,
Mounting file system	Mount
Demount file system	unmount

## Unix Shell Commands

### Login Commands

#### Passwd

- Purpose : This command is used to change the password.
- Usage : It first asks for the old password this password is checked with the stored password for this user, if it is valid it asks for the new password and confirm password.
- **General rules for entering a valid password**
- It must contain at least one numeric or special character within first eight characters.
- It must contain two alphabetic characters
- The length of password must be at least 3 characters and not more than 255 characters.
- It must not be same as old password. It must differ by at least 3 positions.
- It must not be same as login id
- These passwords are case sensitive.

#### Options

- -d, ---delete Delete a user's password (make it empty).

# GEETANJALI GROUP OF COLLEGES

*(Affiliated to Saurashtra University)*

- -l, ---lock      Lock the named account. This option disables an account by changing the password
- -u, ---unlock    Unlock the account.

- Syntax : \$passwd
- Example : \$passwd
- Output : old password :

new password :

confirm password :

## logout / exit

- Def : This command terminate the session and quite the current login.
- Syntax : \$logout
- Example : \$logout
- Output : Output may be either login screen or login command depend upon OS.

## who

- Def. : This command displays all the users who are currently log into the system. It returns the username, the terminal (Device id) and login time of the user.
- Syntax : \$who
- Example : \$who

## who am I

- Def : Instead of getting information about all user who are currently login, if you want to get information about yourself then who am I is used.
- Who command with the arguments “am I” gives the details of the user who executes this command.
- Syntax : \$who am i
- Example : \$who am i

## Clear

- Def : This command clears the terminal screen. It takes no arguments or ignores any command line arguments. It works similar to that of cls command in Dos.
- Syntax : \$clear
- Example : \$clear

## uname

- uname (short for unix name) is a computer program in Unix and Unix-like computer operating systems that prints the name, version and other details about the current machine and the operating system running on it.
- Syntax : \$uname [options]
- Example : \$ uname -a

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

Options	Description
<b>-a Or --all</b>	Displays all available information.
<b>-s Or --kernel-name</b>	Shows the kernel name.
<b>-n Or --nodename</b>	Displays the network (domain) name of the machine.
<b>-r Or --kernel-release</b>	Shows the kernel release.
<b>-v Or --kernel-version</b>	Displays the kernel version.
<b>-m Or --machine</b>	Shows the machine hardware name.
<b>-p Or --processor</b>	Displays the processor type or “unknown.”
<b>-i Or --hardware-platform</b>	Shows the hardware platform or “unknown.”
<b>-o Or --operating-system</b>	Displays the operating system.

## Files & Directory Commands

Command Name	Definition	Syntax & Ex	Options
<b>ls</b>	ls (List Directories) : It lists the directories and files in the current directory.	Syntax : \$ls Example : \$ls -l Output : detailed list of files and directories.	ls -l : Long listing of file showing with seven attributes (File Type, Permission Links, Owner, Group, File Size, Last Modified, File Name) ls -n : It displays the user group id with long list format. ls -a : It is used to display hidden files. ls -R : It displays Folder and Subfolder.

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

<b>cat</b>	This command is generally used to create a file, see the content of the file and append data to a file.	Syntax 1 : Creating a file \$cat > filename Syntax 2 : Append data to a file \$cat >> filename Syntax 3 : See content of a file \$cat filename Example : \$cat > abc.txt hi hello ^z	
<b>cd</b>	It is stands for change directory. Using this command you can move from one directory to another by changing the directory in inner structure.	Syntax : \$cd abc Example : \$cd abc Output : /abc	
<b>cd..</b>	It is stands for change directory. This command is used to move from one directory to another directory in uplevel in directory.	Syntax : \$cd .. Example : /abc cd..	
<b>rm</b>	It is stands for remove. It is used to delete the link to the file. It doesn't mean that the file is physically deleted. You can remove multiple files using a single command.	Syntax : \$rm file1 file2  Example : \$rm abc.txt xyz.txt 9	
<b>mkdir</b>	It is stands for make directory. This command is used to create directories and sub directories. We can create multiple directories using a single command.	Syntax : \$mkdir d1 d2  Example : \$mkdir d1 d2	

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

<b>stat</b>	The <b>stat</b> is a command which gives information about the file and filesystem. Stat command gives information such as the size of the file, access permissions and the user ID and group ID, birth time access time of the file	Syntax: stat --options filenames  Example stat -f filename  stat /etc/resolv.conf	
<b>rmdir</b>	It is stands for remove directory. It is used to remove the one or more directories. If the directory contains the files, it can not be remove and error message "directory is not empty" is return to the user.	Syntax : \$rmdir <d1> <d2> Example : \$rmdir d1 d2	
<b>ln</b>	It is stands for link. This command create a link file from source file.	Syntax : \$ln <file1> <file2> Example : \$cat > abc.txt hi to all ^z \$ln abc.txt xyz.txt \$cat xyz.txt hi to all	
<b>wc</b>	It is stands for word count. It is used to count the number of lines, words and characters in the given file name.	Syntax:\$wc [option] <file name>  Example 1 (count no. of lines): \$cat abc.txt hi to all \$wc -l abc.txt Output : 3	<ol style="list-style-type: none"> <li>1. wc -l Display no. of lines.</li> <li>2. wc -c: Display no. of characters.</li> <li>3. wc -w :Display no. of words.</li> </ol>

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

		<p>Example 2 (count no. of words): \$wc -w abc.txt Output : 3</p> <p>Example 3 (count no. of characters): \$wc -c abc.txt Output : 7</p> <p>Example 4 (count no. of lines, words, characters): \$wc abc.txt Output : 3 3 7</p>	
<b>head</b>	It is used to display the line from the top of the file. If no argument is passed to this command, by default it displays first 10 lines of the files.	<p>Syntax : \$head [-n] &lt;file name&gt; (n is any positive no.) Example :     1) Without option \$head abc.txt 1 2 3 4 5 6 7 8 9 10 2) With option \$head -2 abc.txt 1 2</p>	
<b>tail</b>	It is used to display the bottom lines of a file. If no argument are specify it also display 10 lines from the bottom of the files.	<p>Syntax : \$tail [-arg] &lt;file name&gt; (arg is number of lines to be display) Example :     1) With argument \$tail -3 abc.txt 9 10 11 2) Without argument</p>	

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

		<pre>\$tail abc.txt 2 3 4 5</pre>	
<b>touch</b>	This command changes the last modification time of the file	Syntax : <pre>\$ touch &lt;file name&gt;</pre> Example : <pre>\$ touch abc.txt \$ls -l</pre>	
<b>more</b>	If the file size is larger than the screen then the content of the file with scroll up until it reach at the end of file. User is only able to view the last page.	Syntax : <pre>\$ more &lt;file name&gt;</pre>	
<b>less</b>	This command is similar to that more command but it allows background movement in the file.	Syntax : <pre>\$ less &lt;file name&gt;</pre> Example : <pre>\$ less abc.txt</pre>	
<b>cp</b>	This command is used for copy. This command is used to copy a single file.	Syntax : <pre>\$ cp &lt;source file&gt; &lt;destination file&gt;</pre> Example : <pre>\$cp abc.txt xyz.txt</pre>	
<b>mv</b>	It is stands for move. It is used to rename a file as well as move a file to a different directory.	Syntax : for rename <pre>\$mv &lt;source file&gt; &lt;destination file&gt;</pre> for move a file or directory <pre>\$mv &lt;filename&gt; &lt;path with new name&gt;</pre> Example 1 : For rename a file <pre>\$mv abc.txt xyz.txt</pre> Output : xyz.txt is renamed file of abc.txt	

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

		Example 2 : For move a file \$mv xyz.txt xyz/ Output : xyz.txt movedto xyz directory	
<b>find</b>	This command examines the directory tree to for files matching the criteria and then takes action on the selected files	Syntax :for find file \$find name <filename>  for find files within directory \$find dir type f Example 1 : Find file \$find name file1 Output : /dirb/file1	
<b>alias</b>	n Linux, an alias is a user-defined shorthand for a longer command or sequence of commands. These aliases can be created and customized according to user preferences, making the command-line interface more user-friendly. <b>alias</b> command instructs the shell to replace one string with another string while executing the commands.	<b>alias</b> shortname='longer command' <b>-p option</b> This option prints all the defined aliases is reusable format.  alias -p  alias --help  alias CD="cd Desktop"	
<b>type</b>	The <b>type</b> command is used to describe how its argument would be translated if used as commands. It is also used to find out whether it is built-in or external binary file.	type [Options] commandnamesalias – if command is a shell alias  keyword – if command is a shell reserved word  builtin – if command is a shell builtin function – if command is a shell function file – if command is a disk file	type -a pwd  <b>-a</b> : This option is used to find out whether it is an alias, keyword or a function and it also displays the path of an executable, if available.  type -t pwd



# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## Operator & redirecting piping

- Redirection is a feature in Linux such that when executing a command, you can change the standard input/output devices. The basic workflow of any Linux command is that it takes an input and give an output.
- The standard input (stdin) device is the keyboard.
- The standard output (stdout) device is the screen.
- With redirection, the above standard input/output can be changed.

Command Name	Definition	Syntax & Ex	Options
<b>Output Redirection</b>	The '>' symbol is used for output (STDOUT) redirection.	ls -al > listings	ls -al is re-directed to file "listings" instead of your screen.  ">>" appends output to an existing file.
<b>Input redirection</b>	The '<' symbol is used for input(STDIN) redirection You can type the contents of the email using the standard device keyboard. But if you want to attach a File to email you can use the input re-direction operator in the following format.	Mail -s "Subject" to-address < Filename	
<b>Piping in Unix</b>	Pipe is used to combine two or more commands, and in this, the output of one command acts as input to another command, and this command's output may act as input to the next command and so on. It can also be visualized as a temporary connection between two or more commands/ programs/ processes.	command_1   command_2   command_3  .....   command_N  <b>Listing all files and directories and give it as input to more command.</b>	

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## Finding pattern in file

Command Name	Definition	Syntax & Ex	Options
<b>grep</b>	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).	<b>grep [options] pattern [files]</b>  <b>Ex: \$grep -i "UNix" geekfile.txt</b>	<b>-c</b> : This prints only a count of the lines that match a pattern <b>-h</b> : Display the matched lines, but do not display the filenames. <b>-i</b> : Ignores, case for matching <b>-l</b> : Displays list of a filenames only. <b>-n</b> : Display the matched lines and their line numbers. <b>-v</b> : This prints out all the lines that do not matches the pattern
<b>fgrep</b>	The <i>fgrep</i> filter is used to search for the fixed-character strings in a file. There can be multiple files also to be searched. This command is useful	<b>fgrep [options] [ -e pattern_list] [pattern] [file]</b>  <b>\$fgrep -c "usin.g" para</b>	<b>-c</b> : It is used to print only a count of the lines which contain the pattern. <b>-h</b> : Used to display the matched lines. <b>-i</b> : During
	when you need to search for strings which contain lots of regular expression metacharacters, such as "^", "\$", etc.		comparisons, it will ignore upper/lower case distinction. <b>-l</b> : Used to print the names of files with matching lines once, separated by new-lines. It will not repeat the names of files when the pattern is found more than once. <b>-n</b> : It is used precede each line by its line number in the file (first line is 1).

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

<b>egrep</b>	<b>egrep</b> is a pattern searching command which belongs to the family of <b>grep</b> functions. It works the same way as <b>grep -E</b> does. It treats the pattern as an extended regular expression and prints out the lines that match the pattern. If there are several files with the matching pattern, it also displays the file names for each line.	<b>egrep [ options ] 'PATTERN' files</b>	<b>-c:</b> Used to counts and prints the number of lines that matched the pattern and not the lines. <b>-v:</b> It prints the lines that does not match with the pattern. <b>-i:</b> Ignore the case of the pattern while matching. <b>-l:</b> Prints only the names of the files that matched. It does not mention the matching line numbers or any other information. <b>-L:</b> Prints only the names of the files that did not have the pattern.
--------------	---	--	--

## Working with columns and fields

### cut

- Purpose : cut command is basically used for extracting the vertical columns or set of characters from either standard input from one or more files. The cut command looks for the character positions as well as fixed aligned columns. For specifying the character position we use the **option -c**.

Example :

Roll No.	Student Name	Email id
101	ABC	abc@d.com
102	DEF	def@b.com

\$ cut -c 1-4 file4

- It will extract only the roll numbers from the data file file4.
- Now if the character length of the column is not fixed, it may be more than the character range specified in **-c option** as in the case of student email id or student name.
- We can use -f option

cut -f [column number] filename

- Here, if we need only the student name from file4 then we need the second column So we can use **cut -f 2 file4**. Here 2 specifies the second column.
- Example :
- \$ cut -f 2 file4
- ABC
- DEF

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## Paste

- Paste command combines the lines together. These lines can be from multiple files. If the input is from standard input then (-) hyphen sign is used. It combines the files vertically.

### Example :

**\$ cat file1.txt**

```
1    ABC
2    DEF
3    EFG
```

**\$ cat file2.txt**

```
1    101
2    102
3    103
```

**\$ paste file1 file3**

```
1      ABC      1      101
2      DEF      2      102
3      EFG      3      103
```

- It combines the first line of the first file with the first line of the second file and displays the combination on the standard output. The two lines are separated by a tab. Paste command treats each line as a column.
- **options :**

- **-d delimiter\_list :** It consecutively use the characters in delimiter\_list instead of tab to separate merged lines.

### Example :

**\$ paste -d '%\_' file1 file3**

```
1    ABC%1    101
2    DEF%2    102
3    EFG%3    103
```

## join

- Join command joins two files, here files are joined on common column. In join command the common column (key field) should exist in both files and the data must be sorted on the common column in same order. The default join field is first, delimited by whitespace.
- The join command reads the first line of the first file, then reads lines from the second file until the contents in the key field of the second file match or exceed the contents in the key field of the first file.

# GEETANJALI GROUP OF COLLEGES

*(Affiliated to Saurashtra University)*

## Syntax :

- **join [option] file1 file2**

## Example :

- Suppose we have two files of the student data, one file contains roll number and marks and another contain roll number and name then output will be :

## \$ join file1 file2

```
1 ABC 101
2 DEF 102
3 EFG 103
```

## If we have a file, myfile1.txt, whose contents are :

```
1 India
2 US
3 Ireland
4 UK
5 Canada
```

## And another file, myfile2.txt, whose contents are :

```
1 New Delhi
2 Washington
3 Dublin
4 London
5 Toronto
```

- The common fields are the fields which begin with the same number. We can join the contents using the following command :

## \$ join myfile1.txt myfile2.txt

## Output :

```
1 India New Delhi
2 US Washington
3 Ireland Dublin
4 UK London
5 Canada Toronto
```

If we wanted to create a new file with the joined contents, we could use the following command :

## \$ join myfile1.txt myfile2.txt > newfile.txt

It directs the output into the new file named newfile.txt.

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## Commands for Sorting

Command Name	Definition	Syntax & Ex	Options
<b>sort</b>	The sort command is used for arranging the data in ascending or descending order.	<b>\$ sort file1.txt</b>  <b>\$ sort -r file1</b> 103 102	- <b>r</b> : sorts the file in reverse order - <b>c</b> : Checks if the file is sorted - <b>n</b> : Numeric sort
<b>uniq</b>	Uniq command selects one non repeated line and removes other duplicate lines from the output. It needs a sorted file as an input.		- <b>u</b> : Displays the non-duplicate lines and displays only the uniq lines from the file. - <b>d</b> : display only the duplicate line.  - <b>c</b> : count the number of duplicate lines in the file. It needs sorted file as an input.

## Comparing files

Command Name	Definition	Syntax & Ex	Options
<b>cmp</b>	This command compares two files byte by byte, that means the first byte of first file with the first byte of the second file, if both are same it checks the next byte, wherever the bytes are not same it displays the message that first file and second file differ : byte number, line number as shown below.	<b>\$ cmp file1 file2</b> <b>file1 file2 differ : byte 13, line1</b>	

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

<b>comm</b>	<p>This command is useful to compare two sorted files line by line.</p> <p>The output of comm command consists of three columns, the first column is the lines which are unique to the first file. The second column displays the lines which are unique to the second file and the third column displays the lines which are common between both the files.</p>	<b>\$ comm [option] file1 file2</b>	<p>-1 lines unique to file1</p> <p>-2 lines unique to file2</p> <p>-3 lines that appear in both files</p>
<b>diff</b>	<p>This command finds difference between two files.</p> <p>diff command compares file1 and file2. If both file1 and file2 are directories, diff command compares corresponding files in both directories in alphabetical order.</p>	<b>diff file1 file2</b>	<p>-a Treat all files as text and compare them line by line.</p> <p>-b Ignore changes in amount of white space.</p> <p>-i Ignore changes in case. Consider upper and lower case letters same.</p>

## Changing information in files

Command Name	Definition	Syntax & Ex	Options
<b>tr</b>	<p>This command is useful for changing the case from upper to lower. Translate and delete characters from standard input, writing to standard output.</p>	<b>tr [option] 'set1' 'set2'</b> <b>\$cat file   tr "[a-z]" "[A-Z]"</b>	<p>-d : delete characters in the first set from the output.</p> <p>-s : replaces repeated characters listed in the set1 with single occurrence</p>

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

<b>sed</b>	<p>sed is a Stream Editor used for modifying the files in unix (or linux). Whenever you want to make changes to the file automatically, Consider the below text file as an input.</p> <p>sed command is mostly used to replace the text in a file. The below simple sed command replaces the word "unix" with "linux" in the file.</p>	<p><b>\$ sed 's/unix/linux/' file.txt</b></p> <p>Here the "s" specifies the substitution operation. The "/" are delimiters. The "unix" is the search pattern and the "linux" is the replacement string.</p> <p>By default, the sed command replaces the first occurrence of the pattern in each line and it won't replace the second, third...occurrence in the line.</p>	
------------	--	---	--

## Examining File Contents

Command Name	Definition	Syntax & Ex	Options
<b>od</b>	od command in Linux is used to output the contents of a file in different formats with the octal format being the default.	od [OPTION]... [FILE]... \$ od -b input	<b>Display contents of file in octal format using -b option</b>

## Tools for mathematical calculations

Command Name	Definition	Syntax & Ex	Options
<b>bc</b>	bc is a unix command which stands for bench calculator. bc command is used to do some simple arithmetic.	bc [options] [file...] <b>Example :</b> \$ x=20 \$ y=10 \$ echo \$x+\$y   bc 30	



# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

<b>factor</b>	The <b>factor</b> command in Linux is used to print the prime factors of the given numbers, either given from command line or read from standard input.	factor [NUMBER] <b>Example :</b> \$ factor 100	
---------------	---	--	--

## Monitoring input and output

Command Name	Definition	Syntax & Ex	Options
<b>tee</b>	Reads from standard input, and writes to standard output and to files. tee copies data from standard input to each FILE, and also to standard output.	tee [OPTION]... [FILE]... ls -l *.txt   wc -l   tee count.txt	<b>-a Option :</b> It basically do not overwrite the file but append to the given file. Suppose we have <b>file1.txt</b> <b>-help Option :</b> It gives the help message and exit. <b>-version Option :</b> It gives the version information and exit.
<b>script</b>	script makes a typescript of everything printed on your terminal. It is useful for users who need a hardcopy record of an interactive session as proof of work done.	script [-a] [file]	<b>-a, -append:</b> This option is used when we want to append the output, retaining the prior content of the file. The multiple contents get separated by adding a line that states the date and time of the script started. <b>-c, -command:</b> This option is used when we want to run a particular command rather than interactive shell and get terminal information in the file given as argument or typescript by default.

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

			The script will automatically exit after successful execution. <b>-q, –quiet:</b> This option does not display the notification stating that the script has started and quietly execute and exit the <i>script</i> command.
--	--	--	--

## Tools for displaying date and time

Command Name	Definition	Syntax & Ex	Options
<b>Cal</b>	<b>To display a calendar</b> A single parameter specifies the 4 digit year (1 - 9999) to be displayed. Two parameters denote the Month (1 - 12) and Year (1 - 9999). If arguments are not specified, the current month is displayed. A year starts on 01 Jan.	cal [-m]y] [[month] year]	-m Display monday as the first day of the week. -j Display julian dates (days one-based, numbered from January 1). -y Display a calendar for the current year.
<b>date</b>	The date command is used to print out, or change the value of, the system's time and date information.	date [OPTION]... [+FORMAT] \$ date "+DATE: %m/%d/%y%nTIME: %H:%M:%S"	-d, --date=STRING display time described by string STRING, as opposed to the default, which is 'now'

## Communications

Command Name	Definition	Syntax & Ex	Options
<b>telnet</b>	The telnet command is used to communicate with another host using the TELNET protocol. If telnet is invoked without the host argument, it enters	<b>Syntax :</b> telnet [host [port ]] <b>Example :</b> \$ telnet	

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

	command mode, indicated by its prompt (telnet>).		
<b>wall</b>	wall displays the contents of file or, by default, its <u>standard input</u> , on the <u>terminals</u> of all currently <u>logged in</u> users.	wall [-n] [file] \$ wall message.txt	-n, --nobanner Suppress banner
<b>motd</b>	To display message of the day at login time. The contents of /etc/motd are displayed after a successful login but just before it executes the login shell. The motd stands for "message of the day".		
<b>write</b>	The write utility allows you to communicate with other users, by copying lines from your terminal to theirs. When you run the write command, the user you are writing to gets a message of the format: Message from yourname@yourhost on yourtty at hh:mm ... Any further lines you enter will be copied to the specified user's terminal. If the other user wants to reply, they must run write as well.	<b>Example :</b> \$write hope	user The user to write to.
<b>mail</b>	Being able to send	<b>Syntax :</b> \$mail [option]	

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

	emails from command-line from a server is quite useful when you need to generate emails programmatically from shell scripts or web applications.	[message] [mailid] <b>Example :</b> \$ mail -s "Hello World" someone@example.com	
<b>finger</b>	finger looks up and displays information about system users. It is known as user information look up program.	<b>Syntax :</b> finger [option][user ...]  <b>Example :</b> finger -s myuser	
<b>mesg</b>	useful for controlling write access to your terminal. mesg is useful when you want to control whether other users can write to your terminal. This is especially useful in multi-user environments where privacy or interruption concerns may warrant the blocking of unsolicited messages.	\$mesg n \$mesg y	-n: This option denies write access to your terminal.  -y: This option allows write access to your terminal.
<b>ping</b>	PING (Packet Internet Groper) command is used to check the network connectivity between host and server/host. This command takes as input the IP address or the URL and sends a data packet to the specified address with the message "PING" and get a response from the server/host this time is recorded which is called latency.	ping [options] host_or_IP_address  \$ping www.google.com	

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## Process related commands

Command Name	Definition	Syntax & Ex	Options
<b>ps</b>	On every <u>UNIX</u> -like <u>operating system</u> , the process status command (ps) displays information about active processes.	<b>Syntax :</b> ps <b>Example :</b> \$ ps	
<b>nice</b>	nice command affects <u>process</u> scheduling. A process with a lower niceness value is given higher priority. A process with a higher niceness value is given a lower priority.	<b>Syntax :</b> nice <b>Example :</b> \$nice	
<b>kill</b>	On Unix-like operating systems, the <b>kill</b> command sends a signal to a process. If you don't specify which signal to send, by default the <b>TERM</b> signal is sent, which terminates the process.	<b>Syntax :</b> kill pid <b>Example :</b> \$kill -9	
<b>at and batch</b>	The at and batch command schedules a command to be run once at a particular time. This can be any command that you normally have permission to run. You start by running the at command at the command line, passing it the scheduled time as the option. batch does not accept any parameters	<b>Example :</b> \$ at 9:30 PM Tue \$ batch	

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

<b>cron and crontab</b>	Linux crontab command. On Unix-like operating systems, the crontab command opens the cron table for editing. The cron table is the list of tasks scheduled to run at regular time intervals on the system.	<b>syntax</b> crontab [-u user] file <b>Example :</b> crontab -l	-l Display the current crontab. -r Remove the current crontab.
<b>wait</b>	wait is a built-in <u>shell</u> command which waits for a given <u>process</u> to complete, and returns its exit status. wait waits for the process identified by process ID pid (or the job specified by job ID jobid), and reports its termination status.	<b>Syntax :</b> wait [pid] <b>Example :</b> wait 2017	
<b>Sleep</b>	The sleep command pauses for an amount of time defined by NUMBER. SUFFIX may be "s" for seconds (the default), "m" for minutes, "h" for hours, or "d" for days.	<b>syntax</b> sleep NUMBER[ SUFFIX]... <b>example :</b> \$ <b>sleep 10</b>	
<b>top</b>	<b>top</b> command is used to show the Linux processes. It provides a dynamic real-time view of the running system. Usually, this command shows the summary information of the system and the list of processes or threads which are currently managed by the Linux Kernel.	top top -n 10	<b>Display Specific User Process</b> top -u paras
<b>jobs</b>	<b>jobs</b> - list processes Jobs command is used to list the jobs that you are running in the background and in the foreground. If the prompt is returned with no information no jobs are present. All shells are not capable of running this command. This command is only available in the csh, bash, tcsh, and ksh shells.	\$ jobs \$ jobs -p %p	-l Lists process IDs in addition to the normal information. -n List only processes that have changed status since the last notification. -p Lists process IDs only. -r Restrict output to running jobs. -s Restrict output to stopped jobs.

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## Mount/Unmount commands

Command Name	Definition	Syntax & Ex	Options
<b>mount and unmount</b>	The mount command <u>mounts</u> a <u>storage device</u> or <u>filesystem</u> , making it accessible and attaching it to an existing <u>directory</u> structure. The umount command "unmounts" a mounted filesystem, informing the system to complete any pending <u>read</u> or <u>write</u> operations, and safely detaching it.	<b>Syntax1</b> mount [OPTION...] DEVICE_NAME DIRECTORY  <b>Syntax2</b> mount [OPTION...] DEVICE_NAME DIRECTORY  <b>Example :</b> \$ mount -t type device dir \$ unmount -t type device dir	Use the -t option to specify the file system type.

## UNIT – 4 (Part-1)

### Text Editing With vi and nano Editor, Shell Programming

Topic: 1 Explain VI editor in detail

Mark 5

Ans.

Detailed:

#### Introduction to VI editor

- No matter what work you do with the UNIX system, you will eventually write some C programs or shell scripts. You may have to edit some of the system files at times.
- For all this you must learn to use an editor, and UNIX provides a very versatile one VI editor, VI editor is a screen editor, where a portion of the file is displayed on the terminal screen, and the cursor can be moved around the screen to indicate where you want to make changes.
- You can select which part of the file you want to have displayed.
- Screen editors are also called display editors, or visual editors. VI is one of the more popular screen editors that run on the UNIX system.

#### Invoking VI

- It will put filename into a buffer and display the file on the screen.
- If the file is larger than the screen can display, the screen will act as a window into the file.
- At the beginning of a session, the screen will display the first part of the file.
- If filename does not exist, VI will create it.

# GEETANJALI GROUP OF COLLEGES

*(Affiliated to Saurashtra University)*

## Modes in VI editor

- One of the most important aspects to remember about **vi** is that most of the commands fall into one of three modes:

### (1) Command mode:

- This is the default mode of VI editor. This mode is used to give some command for navigation, edition and copy or cut.
- This is the base mode of VI editor if user want to switch this mode to **insert or input mode then user has to press i or I or a or A.**
- If user wants to switch execute mode from this mode then user has to **press:**

### (2) Input or Insert Mode:

- This mode is used to enter data in VI editor. If user wants to insert some text to the editor then first user has to select this mode.
- **This mode will be selected from command mode pressing I or i or a or A.**

### (3) Execute Mode:

- This mode is used to save or quit from VI editor. Whatever the changes user has done in file using VI editor if user wants to save, find any particular string then using this mode user can save or find the string and also whenever user wants to quit from the VI editor this mode will supports user to quit from the VI editor.
- **To switch to this mode user has to press: or / on command mode**

## Switching mode in VI

- While you start **VI** editor at that time you will be at vi mode that will ready to accept defined command on that particular key but not any input.
- If you want to input text into the file you will have to go to input mode for that you will have to **press "i"**.
- If you will press **"i" at VI** mode you will be at input mode here you can input any data into the file from this mode if you want to switch to vi mode then you will have to press **"Esc"** key.
- If you will press **"Esc"** key at input mode you will be able to switch to vi mode.
- If you want to switch to command mode from the VI mode then you will have to press **":" or "/"** that will support you to switch you from VI mode to command mode.
- If you want to switch to command mode from the input mode then you will have to first come to the VI mode then you will be able to switch to command mode.
- To switch from input mode to command mode then you first will have to press **"Esc"** key and then after you will have to press **":" or "/"** key that will support you to switch from input mode to command mode via **VI** mode.



# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## Cursor movement in VI editor

- Whatever the command you are giving that will work on VI mode only.
- To move around in the file, use the arrow keys: the up arrow will move the cursor up one line, the down arrow down one line.
- The right arrow will cause the cursor to move one position to the right.
- From the end of a line, it will go to the beginning of the next line.
- For terminals with no functional arrow keys, four keys will move the cursor around:

<b>h</b>	<b>left arrow</b>
<b>j</b>	<b>down arrow</b>
<b>k</b>	<b>up arrow</b>
<b>l</b>	<b>right arrow</b>

## Screen control commands

- Whatever the command you are giving that will work on **vi** mode only.

<b>ctrl + f</b>	This will move the user forward one screen in the file.
<b>ctrl+b</b>	This will move the user backward one screen in the file.
<b>G</b>	This will move the cursor to the end of the file. Note, again that vi, like any other UNIX utility, is always case sensitive.
<b>Num</b>	This will bring the cursor to line defined line number.

## Entering, Editing, Copying data into vi editor

- Whatever the command you are giving that will work on vi mode only.

<b>x</b>	That will delete on character.
<b>dd</b>	That will delete lines beginning at the current line.
<b>yy</b>	That will copy lines beginning at the current line.
<b>p</b>	That will paste copied line.
<b>u</b>	This is very useful command. It cancels the effect of the previously executed command.
<b>Q</b>	When in vi, typing will bring the user into command mode. As the <b>:</b> is typed, it will be displayed on the last line of the screen, and vi will wait for a command to be typed. <b>q</b> is such a command. This will exit vi, if no changes have been made since the last write-to-file command <b>:q!</b> will exit, even if the buffer has not been written to the file.
<b>w</b>	Will cause the contents of the buffer to be written to the file <b>:wq</b> will write the buffer to the file, and exit the calling shell.
<b>I</b>	This will support you to switch from VI mode to input mode.
<b>:</b>	This will support you to switch from VI mode to command mode.
<b>esc</b>	This will support you to switch from input mode to VI mode.
<b>/</b>	This will support you to find any particular pattern from the file.

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## One word question answer

### Question

1. \_\_\_\_\_ is the default mode of VI editor.
2. \_\_\_\_\_ mode is used to enter data in VI editor.
3. \_\_\_\_\_ mode is used to save or quit from VI editor.

### Answer

Command Mode  
Input / Insert Mode  
Execute Mode

## Introduction of nano editor

- nano is a simple text editor
- It was created as part of the GNU project started by Richard Stallman
- Most Linux distributions contain the vi and/or vim editors
- These editors are often used by System Administrators
- And you can always download the emacs editor
- emacs is preferred by programmers
- But both of these editors take some time to learn
- nano does not have the full set of features of these editors
- But it is great for people who are just starting to work with Unix
- Or who want to create a file quickly with a minimum of fuss

### Running nano

- To run nano you type nano followed by the name of the file

### **nano FILENAME**

- If the file does not yet exist it will be created for you
- So if I wanted to create a file named .bash\_profile I would type

### **nano .forward**

- This will bring up a text window with the name of the file on the top line



# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## nano Commands

- All command in nano are run with the help of the Control key
- Control key is labeled Cntrl on all machines except Macs
- In the list of commands the symbol ^ stands for the Control key
- To quit nano hold down the Control key and hit X
- The save command in nano is ^O WriteOut
- You might expect it to be ^S Save
- Unfortunately Control S has special meaning in Unix
- So it cannot be used for this purpose
- The same problem exists for Control Q
- So you have to use Control X to quit

## Editing Text in nano

- To enter text in a file, simply type it in the window
- You can use the arrow keys to move around in the text window
- And the Backspace or Delete keys to delete a character
- You can move to the end of a line with Control E
- And the beginning of the line with Control A
- nano has a very simple cut and paste feature
- But it only works on the whole line of text
- You use the arrow keys to get to a line you want to move
- And cut it with Control K
- You can then move to another position in the file
- And use Control U to paste the line into this new position

## Save and Quit

- To save the file, hold down the Control key while pressing O
- When you do this, a line will appear at the bottom of the screen  
File Name to Write:
- This line is asking you to confirm the name of your file
- Hit Enter or Return to accept this filename
- To quit, hold down the Control key while pressing X

## Searching in nano

- nano has a very basic search feature
- If you hit Control W (for Where) a line will appear at the bottom of the screen Search:
- Type in your search text and hit Enter or Return
- The text entry point will move to the first line that has the text
- The search is not case sensitive

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## UNIT – 4 (Part-2)

**Topic: 1**      **What are Shell keywords?**

**Mark 1**

**Ans.**

**Detailed:**

- Keyword is a word which is having specific meaning for particular language.
- In programming different words are used like, if, for, while, until, case etc. these words are called shell keywords.

### One word question answer

Question	Answer
6. What are Shell keywords?	Word with specific meaning
7. Give the name of some shell keywords.	if, for, while, until, case

**Topic: 2**      **Explain Shell Variables.**

**Mark 2**

**Ans.Detailed:**

- Variable is the location in the memory which is used to store the data.
- A variable is a character string to which we assign a value.
- The value assigned could be a number, text, filename, device, or any other type of data.
- The name of a variable can contain only letters (a to z or A to Z), numbers ( 0 to 9) or the underscore character ( \_).
- The following examples are valid variable names –

**\_RAM**  
**TOKEN\_A**  
**VAR\_1**  
**VAR\_2**

- Following are the examples of invalid variable names –

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

2\_VAR  
-VARIABLE  
VAR1-VAR2  
VAR\_A!

## → Defining Variables

Variables are defined as follows –

**variable\_name=variable\_value**

➤ Shell enables you to store any value you want in a variable. For example –

**VAR1="Rohit Sharma" VAR2=100**

## → Accessing Values

➤ To access the value stored in a variable, prefix its name with the dollar sign (\$)

```
#!/bin/sh  
NAME="Rohit Sharma"  
echo $NAME
```

➤ The above script will produce the following value –

**Rohit Sharma**

### One word question answer

#### Question

1. What is variable?
2. How to define variable?

#### Answer

Location to store data  
variable\_name=variable\_value

**Topic: 3 Explain types of Variables in Linux.**

**Mark 5**

**Ans.**

#### Detailed:

➤ There are three types of variables.

1. **Shell Variables**
2. **System Variables or Environment Variable**
3. **Users define variable**

#### 1. Shell Variables

- These variables are used to configure the shell.
- The shell provides string valued variables.

# GEETANJALI GROUP OF COLLEGES

*(Affiliated to Saurashtra University)*

## Example:

- User=root, which assigns values to the variable user.
- The value of variable is substituted by preceding name with \$.
- Other Shell parameters:
  - \$\* : It is the set of positional parameters as a single string.
  - \$@: It is same as \$\* except when enclosed in double quotes.
  - \$? : The exit status of the last command executed as a decimal string.
  - \$# : The number of positional parameters.
  - \$\$ : The process number of this shell (in decimal).
  - \$\_ : The process number of the last process runs in the background.

## 2. System Variables

- These variables are used to configure the environment during the boot sequence or after user login, so they are also known as environment variables.
- The shell has two prompts stored in PS1 and PS2.

**PS1:** It is the primary prompt string like \$ or %. The shell uses the primary prompt when the command is expected by the shell. For changing the value of this prompt, we can use **PS1= '@'**.

As soon as it is set the new prompt is displayed. It can be restored back to \$ prompt as PS1="\$".

**PS2:** It is the secondary prompt; it is used when the command entered on the first line was not able to be completed. The default prompt is >.

**PATH:** Searches the path for commands. If \$PATH is not set then the current directory, /bin and /usr/bin are searched by default. Otherwise \$PATH consists of directory names separated by:

For Example: PATH=:/usr/Bhavik/bin:/bin:/usr/bin

**HOME:** It is path name for the home directory. When the user logs in the system, unix normally places the user in home directory or login directory and is the path stored in variable HOME.

The change the default argument for the cd command.

For Example: **cd /usr/Bhavik/bin**

**LOGNAME:** It contains login name which is stored in the passwd file. This variable shows your username only while Who am I also displays the user name with the terminal type and login name.

**MAIL:** It displays the absolute path name of user's mailbox and determines where all incoming mail addressed to the user is to be stored. This variable is typically set in the file .profile, in the user's login directory.

For Example: MAIL=/var/spool/mail/root

**IFS:** This variable contains internal field separators. These are the characters that separate the parameters and commands. These separators include space, tab and newline. Default separator is space.

**CS –22 Operating Systems Concepts With Unix / Linux**  
**SHELL:** This variable contains the pathname of login shell. Various shells available in unix system include korn shell, bash shell or c shell. User can select any one from them.

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

**TERM:** It indicates the Terminal type being used. The value of this variable can be utilised by VI command. The VI editor makes the use of the control file in the directory /usr/lib/terminfo.

**MAILCHECK:** It contains Interval between tests for new email. The default is 600 seconds. When the user logs in, shell checks mail file to determine if there is new mail. If mail is there, it display "You have a mail".

### 3. User defined variables

- The variable which is created by the user is known as user defined variables.
- 1. The variable name should not be the same as one of the predefined variables.
- 2. The name of the variable must start with an alphabet or underscore.
- 3. Unix is case sensitive, so SUM and sum are treated as different variables.

#### ➔ set, unset, and echo command with variables

- You can use set command to set any variable.
- You can use unset command to unset/remove any variable.
- You can use echo command to print the value of any variable.

#### One word question answer

##### Question

1. What is shell variable?
2. What is system variable?
3. How many prompts shell has?

##### Answer

Used to configure shell  
Used to configure system environment  
PS1 & PS2

**Topic: 4 Explain Positional parameters.**

**Mark 3**

**Ans.**

#### Detailed:

- A **positional parameter** is a variable within a shell program; its value is set from an argument specified on the command line that invokes the program.
- **Positional parameters** are numbered and are referred to with a preceding ``\$``: \$1, \$2, \$3, and so on.
- **Example:**

\$sh filename.sh	first	second	third	fourth	fifth	sixth	seventh	eighth	ninth
\$0	\$1	\$2	\$3	\$4	\$5	\$6	\$7	\$8	\$9

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

- Arguments are user supplied data that follow the file name on the command line and are input to the shell script.
- Positional Parameters are predefined memory variables in the shell scripts.
- There are nine positional parameters \$1, \$2..... \$9 that are used to store the user arguments.
- The file name is stores in \$0.
- When script is executed the shell puts the first argument in first positional parameter \$1, second argument in second positional parameter \$2 and so on.

## One word question answer

Question	Answer
1. How many positional parameters are available?	9
2. Which positional parameter store file name?	\$0

Topic: 5 Explain Decision statements in Linux.

Mark 5

Ans.

Detailed:

- While writing a shell script, there may be a situation when you need to adopt one path out of the given two paths. So, you need to make use of **conditional statements** that allow your program to make correct decisions and perform the right actions.
- Unix Shell supports **conditional statements** which are used to perform different actions based on different conditions.
- Unix shell provides decision making using if then else and case structure.

**If – then – fi**

**Syntax:**

```
If [condition]
then
execute command if condition is true
...
fi
```

**Example:**

```
NAME="Rohit"
if [ $NAME = "Rohit" ]
then
echo "True - my name is Rohit"
```



# GEETANJALI GROUP OF COLLEGES

*(Affiliated to Saurashtra University)*

fi

## **If – else – fi**

- If given condition is true, then command1 is executed else command2 is executed.

### **Syntax:**

If [condition]

then

execute command1 if condition is true

else

execute command2 if condition is false

fi

### **Example:**

```
NAME="Virat"
```

```
if [ $NAME = "Rohit" ]
```

```
then
```

```
echo "True - my name is Rohit"
```

```
else
```

```
echo "False"
```

```
echo "My name is $NAME"
```

```
fi
```

## **if...elif...else...fi**

- The if...elif...fi statement is the one level advance form of control statement that allows Shell to make correct decision out of several conditions.

### **Syntax:**

if [ condition 1 ]

then

Statement to be executed if condition 1 is true

elif [ condition 2 ]

then

Statement to be executed if condition 2 is true

else

Statement to be executed if no expression is true

fi

### **Example:**

```
NAME="Rohit"
```

```
if [ $NAME = "Virat" ]
```

```
then
```

```
echo "Virat Kohli"
```

```
elif [ $NAME = "Rohit" ]
```

```
then
```

```
echo "Rohit Sharma"
```

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

```
else  
echo "This is Dhoni"  
fi
```

## The case...esac Statement

- Unix Shell supports case...esac statement which handles exactly this situation, and it does so more efficiently than repeated if...elif statements.
- There is only one form of case...esac statement

### Syntax:

```
case "$variable" in  
"$condition1" )  
command...  
;;  
"$condition2" )  
command...  
;;  
esac
```

### Example:

```
mycase=1  
case $mycase in  
1) echo "You selected unix";;  
2) echo "You selected php";;  
3) echo "You selected java";;  
4) echo "You selected c++";;  
5) exit  
esac
```

### One word question answer

#### Question

1. How many conditional statements are in Unix?
2. How can we end if condition statement?
3. How can we end case condition statement?

#### Answer

If & case  
Fi  
esac

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

Topic: 6 Explain test command in Linux.

Mark 2

Ans.

**Detailed:**

- The test command is used to check file types and compare values. Test is used in conditional execution.

**Syntax:**

test condition

OR

test condition && true-command

OR

test condition || false-command

OR

test condition && true-command || false-command

**Example :**

- Type the following command at a shell prompt (is 5 greater than 2? ):

```
test 5 -gt 2 && echo "Yes"
```

```
test 1 -lt 2 && echo "Yes"
```

**Sample Output:**

```
Yes
```

```
Yes
```

**One word question answer**

**Question**

1. The test command is used to\_\_\_\_\_.

**Answer**

check file types and compare values

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

Topic: 7 Explain logical operators in Linux.

Mark 3

Ans.

Detailed:

- **Logical Operators:** They are also known as Boolean operators.
- These are used to perform logical operations.
- They are of 3 types:
  - **Logical AND (&& / -a):** This is a binary operator, which returns true if both the operands are true otherwise returns false.
  - **Logical OR (|| / -o):** This is a binary operator, which returns true if either of the operand is true or both the operands are true and returns false if none of them is false.
  - **Not Equal to (!):** This is a unary operator which returns true if the operand is false and returns false if the operand is true.

Example:

```
# take a number from the user
echo "Enter a number: "
read a

# check
if [ `expr $a % 2` == 0 -a $a -gt 10 ]
then
echo "$a is even and greater than 10."
else
echo "$a failed the test."
fi
```

## One word question answer

### Question

1. Logical Operators are also known as\_\_\_\_\_.
2. How many types of logical operators available?

### Answer

Boolean operators  
3 types

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

Topic: 8 Explain looping structure in Linux.

Mark 5

Ans.

Detailed:

- There is total 2 looping statements which can be used in bash programming

1. **while statement**
2. **for statement**

- To alter the flow of loop statements, two commands are used they are,

1. **break**
2. **continue**

- Their descriptions and syntax are as follows:

## ➔ **while statement**

- Here command is evaluated and based on the result loop will executed, if command raise to false then loop will be terminated.

**Syntax:**

```
while command
do
Statement to be executed
done
```

**Example:**

```
a=0
# -lt is less than operator
```

```
#Iterate the loop until a less than 10
while [ $a -lt 10 ]
do
# Print the values
echo $a
```

```
# increment the value
a=`expr $a + 1`
done
```

## ➔ **for statement**

- The for loop operate on lists of items.
- It repeats a set of commands for every item in a list.
- Here var is the name of a variable and word1 to wordN are sequences of characters separated by spaces (words).

# GEETANJALI GROUP OF COLLEGES

*(Affiliated to Saurashtra University)*

- Each time the for loop executes, the value of the variable var is set to the next word in the list of words, word1 to wordN.

## **Syntax:**

```
for var in word1 word2 ...wordn
do
Statement to be executed
done
```

## **Example:**

```
#Start of for loop
for a in 1 2 3 4 5 6 7 8 9 10
do
# if a is equal to 5 break the loop
if [ $a == 5 ]
then
break
fi
# Print the value
echo "Iteration no $a"
done
```

## ➔ **until loop**

- As an exit control loop, until loop is available.

## **Syntax:**

```
until [condition]
do
command1
command2
done
```

## **Example:**

```
a=10

until [ $a -lt 10 ]
do
echo $a
a=`expr $a + 1`
done
```

# GEETANJALI GROUP OF COLLEGES

*(Affiliated to Saurashtra University)*

## ➔ The break Statement

- The **break** statement is used to terminate the execution of the entire loop, after completing the execution of all the lines of code up to the break statement.
- It then steps down to the code following the end of the loop.

### Syntax:

The following **break** statement is used to come out of a loop –  
break

### Example:

```
a=0

while [ $a -lt 10 ]
do
echo $a
if [ $a -eq 5 ]
then
break
fi
a=`expr $a + 1`
done
```

## ➔ The continue statement

- The **continue** statement is like the **break** command, except that it causes the current iteration of the loop to exit, rather than the entire loop.
- This statement is useful when an error has occurred, but you want to try to execute the next iteration of the loop.

### Syntax:

continue

### Example:

```
NUMS="1 2 3 4 5 6 7"

for NUM in $NUMS
do
Q=`expr $NUM % 2`
if [ $Q -eq 0 ]
then
echo "Number is an even number!!"
continue
fi
echo "Found odd number"
done
```

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## One word question answer

### Question

1. The **break** statement is used to \_\_\_\_\_.
2. How many looping statements available?

### Answer

terminate the execution of the entire loop  
2

**Topic: 9 Explain Arithmetic in shell.**

**Mark 3**

**Ans.**

### Detailed:

- Operators are used for manipulating variables and constants in shell programs.
- They are required to perform mathematical operations.

### ➔ Shell Arithmetic Operators

- **Addition: +**
- **Subtraction: –**
- **Multiplication: \***
- **Division: /**
- **Modulus: %**

Each of these operators performs the operation on two integer variables or constants.

**For Example, the below program illustrates each of these operations:**

```
$ c=`expr $a + $b`  
$ echo "the value of addition=$c"  
$ d=`expr $a - $b`  
$ echo "the value of subtraction=$d"  
$ e= `expr $a \* $b`  
$ echo "the value of multiplication=$e"  
$ f=`expr $a / $b`  
$ echo "the value of division=$f"  
$ g= `echo `expr $a % $b`  
$ echo "the value of modulus=$c"
```

## One word question answer

### Question

1. Arithmetic operators are used to \_\_\_\_\_

### Answer

perform mathematic calculations



# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## Topic: 8 Explain Array in Linux.

- An array is a structured arrangement of similar data elements. Within shell scripting, an array is a variable that holds multiple values, whether they are of the same type or different types

### How to Declare Array in Shell Scripting?

#### 1. Indirect Declaration:

```
ARRAYNAME[INDEXNR]=value
```

#### 2. Compound Assignment:

```
ARRAYNAME=(value1 value2 .... valueN)
```

Alternatively, you can use index numbers to assign values explicitly:

```
ARRAYNAME=([1]=10 [2]=20 [3]=30)
```

#### Example:array.sh

```
#!/bin/bash
#Script to print the keys of the array
#Declaring the Array
declare -a example_array=( "Welcome""To""Geetanjali" )
#Printing the Keys
echo "${!example_array[@]}"
```

**Printing the keys of an Array :** `${!ARRAY_NAME[index]}`

**Finding Array Length :** `${#ARRAY_NAME[@]}`

**Adding Elements to an Array :** `ARRAY_NAME[index_n]="New Element"`

**Deleting an Element from an Array :** `unset ARRAY_NAME[index]`

**Deleting the Entire Array :** `unset array_Name`

#### Loop through the Array:

```
#!/bin/bash
#Script to print all keys and values using loop through the array
declare -a example_array=( "Welcome""To""Javatpoint" )
#Array Loop
for i in "${!example_array[@]}"
do
echo The key value of element "${example_array[$i]}" is "$i"
done
```

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

**Topic: 8 Explain Function in Linux.**

**Mark - 5**

With the help of functions, overall functionality of a function can be divided into smaller or logical parts, which can be called to perform their task.

- **To create a Function:**

**Syntax:**

```
function functionName () {  
    Commands to be executed  
}
```

**Example:**

```
#!/bin/bash  
demo () {  
    echo 'Welcome to Geetanjali'  
}  
demo
```

- **Passing Arguments**

```
#!/bin/bash  
#Script to pass and access arguments  
function_arguments()  
{  
    echo $1  
    echo $2  
    echo $3  
    echo $4  
    echo $5 }  
  
#Calling function_arguments  
function_arguments "We""welcome""you""on"Geetanjali."
```

- **Variable Scope**

Global variables are defined as the variables which can be accessed anywhere within the script regardless of the scope. By default, all the variables are defined as global variables, even if they are declared inside the function. We can also create variables as a local variable. Local variables can be declared within the function body with the `local` keyword when they are assigned for first time. They are only accessible inside that function. We can create local variables with the same name in different functions. To add a local variable, we can use the following syntax:

# GEETANJALI GROUP OF COLLEGES

*(Affiliated to Saurashtra University)*

local var\_name=<var\_value>

- **Example:**

```
#!/bin/bash
v1='A'
v2='B'
my_var () {
  local v1='C'
  v2='D'
  echo "Inside Function"
  echo "v1 is $v1."
  echo "v2 is $v2." }
echo "Before Executing the Function"
echo "v1 is $v1."
echo "v2 is $v2."
my_var
echo "After Executing the Function"
echo "v1 is $v1."
echo "v2 is $v2."
```

- **Return Value[Example 1]**

```
#!/bin/bash
print_it () {
  local my_greet="Welcome to Geetanjali."
  echo "$my_greet"
}
my_greet="$(print_it)"
echo $my_greet
```

- **Return Value[Example 2]**

```
#!/bin/bash
#Setting up a return status for a function
print_it () {
  echo Hello $1
  return 5 }
print_it User
print_it Reader
echo The previous function returned a value of $?
```

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## UNIT – 5 (Part-1)

### Getting Started with Linux, Linux Booting, Linux Admin (Ubuntu)

**Topic: 1** Give the introduction of Linux history.

**Mark 2**

**Ans.**

**Detailed:**

- Linux started in 1991.
- Linux was created by Linus Torvalds. Initially UNIX was available as operating system, but it was used on huge computers with business and universities in 1970.
- Linus had tried to develop one program especially for the PC with an 80386 processor and independent of any operating system.
- He was developing with use of MINIX (MINIX is kind of operating system which is clone of UNIX).

#### One word question answer

##### Question

1. Linux was developed by\_\_\_\_\_.

##### Answer

Linus Torvalds

**Topic: 2** Explain GNU and GPL concept.

**Mark 2**

**Ans.**

**Detailed:**

- To prepare free UNIX like operating system, Richard Stallman has started a project called GNU General Public License (GPL).
- In 1990 enough software were available to create full operating system then also GNU was not successful.
- The GNU software was distributed for free to anyone who wanted the source code was also available so user can modify the code according to their need.
- GNU and Linux developers worked to integrate GNU components with Linux kernel to make full functional and free operating system.

#### One word question answer

##### Question

1. GPL stands for.

##### Answer

General Public License

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

**Topic: 3 Explain Open source and freeware.**

**Mark 2**

**Ans.**

**Detailed:**

- Open source means the software code is available with licensed in which the copyright holder provides the rights to study, modify, and redistribute the software to anyone without any cost and with any purpose.
- Freeware means software is available without cost.
- Freeware generally available with some restricted rights.
- There is not proper definition for freeware and open source, but it can be identified in following terms.
- Anyone can join the open-source project, where in freeware a closed development group is developing the software.
- In open-source code is available so new version can be created from the existing software, where in free ware code is private.

## One word question answer

### Question

1. What is Freeware?
2. Open-source means?

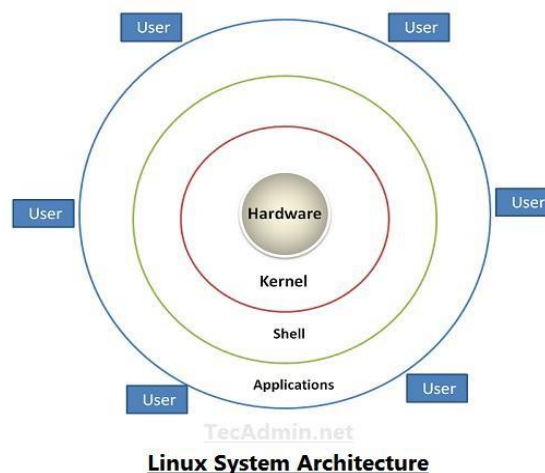
### Answer

Software available without cost  
Code is available online

**Topic: 4 Explain structure & features of Linux.**

**Mark 5**

**Ans.**



# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## Detailed:

### Linux Architecture

- The Linux system works basically on 4 layers.
- First view the diagram above, which is showing the architecture of a Linux System.
- **Hardware** – Hardware consists of all physical devices attached to the System. For example: Hard disk drive, RAM, Motherboard, CPU etc.
- **Kernel** – Kernel is the core component for any (Linux) operating system which directly interacts with the hardware.
- **Shell** – Shell is the interface which takes input from Users and sends instructions to the Kernel, Also takes the output from Kernel and send the result back to output shell.
- **Applications** – These are the utility programs which runs on Shell. This can be any application like Your web browser, media player, text editor etc.

### Linux Features

- Following are some of the important features of Linux Operating System.
- **Portable** – Portability means software's can works on different types of hardware's in same way. Linux kernel and application programs supports their installation on any kind of hardware platform.
- **Open Source** – Linux source code is freely available, and it is community-based development project. Multiple teams work in collaboration to enhance the capability of Linux operating system and it is continuously evolving.
- **Multi-User** – Linux is a multiuser system means multiple users can access system resources like memory/ ram/ application programs at same time.
- **Multiprogramming** – Linux is a multiprogramming system means multiple applications can run at same time.
- **Hierarchical File System** – Linux provides a standard file structure in which system files/ user files are arranged.
- **Shell** – Linux provides a special interpreter program which can be used to execute commands of the operating system. It can be used to do various types of operations, call application programs etc.
- **Security** – Linux provides user security using authentication features like password protection/ controlled access to specific files/ encryption of data.

### One word question answer

#### Question

1. What is Kernel?
2. What is shell?
3. Write any two features of Linux.

#### Answer

Core component of OS  
Interface between user and kernel  
Portable, shell

# GEETANJALI GROUP OF COLLEGES

*(Affiliated to Saurashtra University)*

**Topic: 5 Explain installation and configuration of Linux using ubuntu.**

**Mark 5**

**Ans.**

**Detailed:**

## **Installation & configuration of Linux**

### **Step 1 – Boot from the CD-ROM**

- Installation of UBUNTU requires an installation (bootable) CDROM.
- To make CDROM as first boot device.
- When computer boots up it shows the options to enter the BIOS Setup. Generally, F2 or delete key is pressed to enter the CMOS (BIOS) setup.
- As CDROM is the first boot device, the computer will first boot from the CDROM and it would find the installation CD in the CDROM, and it will ask the user whether computer should be booted using CDROM. Press enters when it is asked to boot from CDROM.
- The installation process starts, and it initially greets the user with a welcome screen. For default installation, press Enter and for server version type server and press enter.

### **Step 2 – Selection of Language**

- Ubuntu supports many languages, and this step in the installation is to select one of the languages from the given list. The default language is English.

### **Step 3 – Keyboard Layout**

- Next, it asks to configure the keyboard layout. Generally, it will be automatically selected, so you can press Enter to ensure the selection. “Test your keyboard” option allows checking whether it is properly configured.

### **Step 4 – Hardware and Networking Auto Discovery**

- Next, Ubuntu configure your hardware and network connection at this stage.
- If there is any network related error then we have option for “Do not configure the network at this time”, so installation will not be interrupted due to this error.

### **Step 5 – Set a Hostname**

- Installation asks for setting a host name for the computer.

### **Step 6 – Partition the Disk**

- Partition has four option:
  1. Resize IDEI master (IDE) and use free space
  2. Erase entire disk: IDE Master

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

3. Erase entire disk using LVM
4. Use target continuous free space
5. Manually edit partition table

➤ The Users can partition the disk manually by providing the appropriate space for the partitions.

## Step 7 – Wait during Automated package copying

### Step 8 – Set the Time Zone

➤ Depending on your location, set the time zone for your locality.

### Step 9 – Set your Real Name, Username and Password

➤ To login to the system, users require username and password, which needs to be unique.

### Step 10 – Install the GRUB Boot Loader

- If windows is also installed on the same hard disk as Ubuntu, It will verify that GRUB boot loader is to be installed.
- The GRUB boot loader displays the list of operating systems installed on the computer when computer starts booting.

## Step 11 – Reboot for the Second Installation Phase

## Step 12 – Wait During Package Installation

### Step 13 – Configure your Monitor

➤ After package are installed, set the resolution for monitor

### Step 14 – Boot for the first time

- After rebooting it display the Ubuntu login screen, where user can enter the password and enter the system.

#### One word question answer

Question	Answer
1. Which will be the first installation screen?	language selection
2. What is GRUB?	Boot loader



# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

Topic: 6 Explain startup, shutdown, and boot loader in Linux.

Mark 3

Ans.

Detailed:

## shutdown

- If you are going to initiate a shutdown of your system and there are other users logged in to your system, it is always polite to notify them first.
- To do this, we would use the **shutdown** command. The shutdown command will accept times and a broadcast message:

shutdown [options] time [warning\_message]

- shutdown and reboot immediately

```
$ shutdown -r now
```

## Linux's boot process

- The Linux boot process is the name given to the startup procedures/order that your system goes through to load its operating system.
- In the following example, I will be referring to booting Linux on x86 architecture.

## Boot Loader

- The Ubuntu installation routine writes a new boot sector (also known as boot loader).
- The boot loader is a separate program called GRUB (Grand Unified Boot Loader).
- If more than one operating system is installed, GRUB boot loader with list of all installed operating system appears.
- The first option will be selected automatically within 10 seconds.
- In the boot loader screen, all pre-installed operating systems are listed, apart from that an entry ending in "(recovery mode)".
- This is a safe mode within windows, in that the system boots with safe system settings.

## One word question answer

### Question

1. Which command is used to shut down and reboot immediately?
2. GRUB stands for.

### Answer

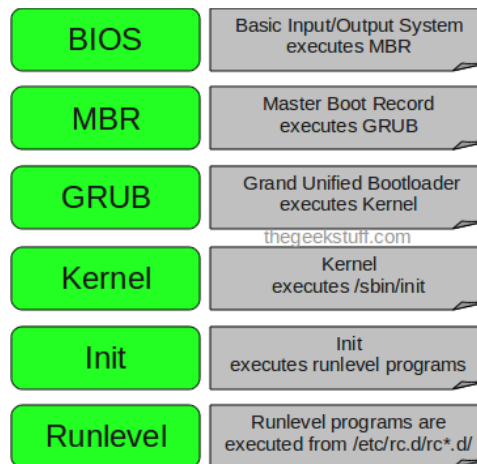
\$ shutdown -r now  
Grand Unified Boot Loader

## UNIT – 5 (Part-2)

Topic: 1 Explain Linux booting process.

Mark 3

Ans.



Detailed:

- The linux booting process is series of 6 stages which is also known as start-up sequence.
- The booting process is initiated and completed during the time you press the power button and until the login screen appears.

### 1. BIOS

- BIOS stands for Basic Input/Output System
- Performs some system integrity checks
- Searches, loads, and executes the boot loader program.

### 2. MBR

- MBR stands for Master Boot Record.
- It is located in the 1st sector of the bootable disk.
- MBR is less than 512 bytes in size.
- So, in simple terms MBR loads and executes the GRUB boot loader.

### 3. GRUB

- GRUB stands for Grand Unified Bootloader.
- If you have multiple kernel images installed on your system, you can choose which one to be executed.

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## 4. Kernel

- Mounts the root file system as specified in the “root=” in grub.conf
- Kernel executes the /sbin/init program

## 5. Init

- Looks at the /etc/inittab file to decide the Linux run level.
- Following are the available run levels
  - 0 – halt
  - 1 – Single user mode
  - 2 – Multiuser, without NFS
  - 3 – Full multiuser mode
  - 4 – unused
  - 5 – X11
  - 6 – reboot

## 6. Run level programs

- When the Linux system is booting up, you might see various services getting started.
- For example, it might say “starting send mail .... OK”.
- Those are the run level programs, executed from the run level directory as defined by your run level.

### One word question answer

#### Question

1. How many stages of the Linux booting process?
2. BIOS stands for.
3. MBR stands for.

#### Answer

- 6  
Basic Input/Output System  
Master Boot Record

**Topic: 2 Explain LILO Configuration.**

**Mark 3**

**Ans.**

Detailed:

- LILO (Linux Loader) is a boot loader for Linux and was the default boot loader for most Linux distributions.
- Today, many distributions use GRUB as the default boot loader, but LILO is still in wide use.
- To work with LILO, the administrator needs to edit the file /etc/lilo.conf to set a default partition to boot, the time-out value, where LILO should be installed and other information.
- After that the administrator must update the loader by running the LILO command.

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## LILO Configuration File

➤ The following is a sample /etc/lilo.conf

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
message=/boot/message
lba32
default=linux
```

### One word question answer

#### Question

1. LILO stands for.
2. Which is lilo configuration file?

#### Answer

Linux Loader  
lilo.conf

**Topic: 3 Explain GRUB Configuration.**

**Mark 3**

**Ans.**

#### Detailed:

- GRUB is a boot loader designed to boot a wide range of operating systems from a wide range of filesystems.
- GRUB is becoming popular due to the increasing number of possible root filesystems that Linux can reside upon.
- GRUB is documented in a GNU info file.
- The **GRUB** boot loader uses the configuration file /boot/grub/grub.conf.

## GRUB Configuration File

```
default=0
timeout=10
splashimage=(hd0,0)/grub/splash.xpm.gz
```

- The 'default =' option tells GRUB which image to boot by default after the timeout period.
- The 'splashimage' option specifies the location of the image for use as the background for the GRUB GUI.

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## One word question answer

### Question

1. GRUB is a \_\_\_\_\_.
2. Which is grub configuration file?

### Answer

Boot Loader  
grub.conf

**Topic: 4 Explain user interface GUI and CUI.**

**Mark 3**

**Ans.**

### Detailed:

- User Interface is the junction between a user and a computer program.
- An interface is a set of commands or menus through which a user communicates with a program.
- A command-driven interface is one in which you enter commands.
- A menu-driven interface is one in which you select command choices from various menus displayed on the screen.
- The user interface is one of the most important parts of any program because it determines how easily you can make the program do what you want.
- A powerful program with a poorly designed user interface has little value.
- Graphical user interfaces (GUIs) that use windows, icons, and pop-up menus have become standard on personal computers.

### GUI

- A GUI is a graphical (rather than purely textual) user interface to a computer.
- The term came into existence because the first interactive user interfaces to computers were not graphical; they were text-and-keyboard oriented and usually consisted of commands you had to remember and computer responses that were infamously brief.
- The command interface of the DOS operating system (which you can still get to from your Windows operating system) is an example of the typical user-computer interface before GUIs arrived.
- Today's major operating systems provide a graphical user interface.
- Applications typically use the elements of the GUI that come with the operating system and add their own graphical user interface elements and ideas.
- It is very user-friendly interface. Example: Windows, Linux

### CUI

- CUI stands for Character User Interface or Command User Interface.
- In CUI user must interact with the application by making use of commands, in CUI only one task can run at a time.
- Everything is done by using commands, Example: DOS, UNIX

# GEETANJALI GROUP OF COLLEGES

*(Affiliated to Saurashtra University)*

## One word question answer

### Question

1. User Interface is \_\_\_\_\_.
2. GUI is a \_\_\_\_\_
3. CUI is a \_\_\_\_\_

### Answer

junction between a user and a computer program  
graphical user interface  
command user interface

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## UNIT – 5 (Part - 3)

**Topic: 1** Explain how to create Linux user account and password.

**Mark 3**

**Ans.**

**Detailed:**

### Creating Linux User Account and Password on Ubuntu

To create a new user in Ubuntu, use the adduser command. The following is the syntax to create.

**sudo adduser newusername**

- The sudo is a program for computer operating system such as UNIX, Linux that allows users to run programs with the security privileges of another user, by default the Superuser.
- The name is combination of “Do as Su” where “Su” is for “Super user”.

After typing that command following process started by Ubuntu:

- Type and confirm password.
  - Enter in the user’s information. This is optional. Press enter key to automatically fill the field with the default information.
  - Press Y (or Enter key) when asked to confirm the information.
- You have just added a new user to your Ubuntu operating system.

### Deleting a User

- If any time you want to delete a specific user on the virtual private server, you can delete them with the following command:

**sudo user del username**

- You should also delete the user’s home directory. This can be done by following simple command :

**sudo rm -rf /home/username**

### One word question answer

Question	Answer
3. To create a new user in Ubuntu which command is used?	sudo adduser newusername
4. The name “sudo” is combination of _____	Do as Su

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

Topic: 2 Explain Samba server in detail.

Mark 3

Ans.

Detailed:

- Samba is a software package, released in 1992, gives network administrators flexibility and freedom in terms of setup, configuration and choice of systems and equipment.
- Because of its ease and flexibility, Samba has become extremely popular and continues to do so.
- Samba runs on UNIX platforms, but can communicate to Windows clients just like the native platform.
- It allows a Unix system to move into a Windows Network without causing any mix-up.
- Windows users can easily access file and print services without having any problem.

## Installing and Managing Samba Server

- If you want to share files between your Ubuntu and Windows computers, the best option is the Samba.
- Download latest version of samba from the site: <https://www.samba.org/samba/ftp/samba-latest.tar.gz>
- To install, first open a terminal window and enter the following command:  
**sudo apt-get install samba smbfs**
- After the successful execution of this command, the Samba will get installed, now you must configure it to make it accessible.
- To configure it, open the configuration file using the following command in an editor:  
**sudo gedit /etc/samba/smb.conf**

Find the following section in the file:

**# security = user**

- Remove the comment before the security line and add the following line just below it:  
**security = user**  
**username map = /etc/samba/smbusers**
- This will set Samba to use the smbusers file for looking up the user list.

## Creating Samba User

- There are two steps to creating a user.
- First we will run the smbpasswd utility to create a samba passwd for the user.

**sudo smbpasswd -a <username>**

- Next, we will add the username to the smbusers file.  
**sudo gedit /etc/samba/smbusers**



# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

- Add the following line, replacing the username with the one you want to give access to.
- The format is

<UbuntuUserName> = "<SambaUserName>".

## One word question answer

### Question

1. To create a new user in Ubuntu which command is used?
2. The name 'sudo' is combination of \_\_\_\_\_
3. If you want to share files between your Ubuntu and Windows computers, the best option is

### Answer

sudo adduser newusername  
Do as Su  
Samba

**Topic: 3 Explain Apache server in detail.**

**Mark 3**

**Ans.**

### Detailed:

- The Apache Web Server is one of the most popular open-source HTTP servers.
- It is powerful, secure, and fully featured web server.

### Installing Apache HTTP Web Server

- First, you should have the Apache server downloaded or download the Apache Server for Linux.
- Now extract the Apache Files by following commands:

```
gunzip -d httpd-2_0_NN.tar.gz  
tar xvf httpd-2_0_NN.tar
```

- To install Apache server on Ubuntu, use following commands:

```
sudo apt-get update  
sudo apt-get install apache2
```

- The Apache server is installed.
- To check if Apache is installed, go to your browser, enter your server's IP address.
- The page should display the message: "It Works!"

### Managing Apache Server

- You must understand the main configuration files of Apache which can be found in the "/etc/apache2" folder.
- You can explore this folder by following command:

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

**cd /etc/apache2**

**ls**

you will see file listing. Following configuration files are necessary to understand:

**apache2.conf** : This is the main configuration file for the server. You can do almost all configurations within this file.

**ports.conf** : You can specify the ports from where virtual hosts can listen.

**conf.d/ (Directory)** : This directory is used for controlling specific features of the Apache configuration. For example, It is frequently used to define SSL configuration.

**sites-available/ (Directory)** : This directory contains all of the virtual hosts files that define different web sites.

**sites-enabled/ (Directory)** : This directory defines which virtual host definitions are actually being used.

**mods-[enabled,available]/ (Directory)** : These directories are similar in function to sites directories, but they define modules that can be loaded optionally.

## One word question answer

### Question

1. To install Apache server on Ubuntu which command is used?
2. Where can we find the main configuration files of apache?
3. The main configuration file for the apache server is\_\_\_\_\_.

### Answer

sudo apt-get install apache2  
"/etc/apache2" folder  
apache2.conf

**Topic: 4 Explain how to optimize LDAP Services.**

**Mark 3**

**Ans.**

**Detailed:**

### Optimizing LDAP Services

- LDAP (Lightweight Directory Access Protocol) is a protocol for managing information from a centralized location over the use of a file and directory hierarchy.
- It can be used to organize and store any kind of information.
- LDAP is mainly used for centralized authentication.

### Install LDAP

- The OpenLDAP server is in Ubuntu's default package "slapd", so we can install it easily with apt-get.

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

- We will also install some additional utilities.

**sudo apt-get update**

**sudo apt-get install slapdldap-utils**

You must enter administrator password for the admin LDAP account.

## Reconfigure slapd

- When the installation is complete, we need to reconfigure the LDAP package.  
Type the following to reconfigure:

**sudo dpkg-reconfigure slapd**

You must answer some questions about the configuration such as:

**OpenLDAP Server configuration? Reply No.**

**DNS Domain Name? test.com**

And so on...

## One word question answer

### Question

1. LDAP stands for.
2. Which command is used to install LDAP package?
3. Command to reconfigure the LDAP package.

### Answer

Lightweight Directory Access Protocol  
sudo apt-get install slapdldap-utils  
sudo dpkg-reconfigure slapd

**Topic: 5 Explain DNS optimizing.**

**Mark 2**

**Ans.**

**Detailed:**

- We can also optimize DNS (Domain Name System) services on Linux (Ubuntu).
- The DNS is, as the name suggests, a simple naming system for your server so that you can remember the address by its name rather than its IP address.

## One word question answer

### Question

1. DNS stands for.

### Answer

Domain Name System

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

**Topic: 6 Explain FTP optimizing.**

**Mark 2**

**Ans.**

**Detailed:**

- FTP servers are a common way of sharing files between individuals with or without authentication.
- FTP is superior to HTTP (web) if the goal is to transfer files, not display information.
- After successful setup of an FTP server, you may need to optimize its services to get the better performance every time.
- For this, you need its proper configuration.
- FTP is file transfer protocol; security must be for safe file transfer.

## One word question answer

### Question

1. FTP stands for.

### Answer

File Transfer Protocol

**Topic: 7 Explain how to optimize web services.**

**Mark 2**

**Ans.**

**Detailed:**

- Apache is a powerful and capable web server.
- To make initial setup as easy as possible, this makes it a great choice for new projects when you need to be productive quickly.
- Here are the few steps to optimize the web service performance of Apache Web Server.

### 1. Unload Unneeded Modules

- In Ubuntu based system, you will see a folder called /etc/apache2/mods-enabled and /etc/apache2/mods-available/.
- The mods-available folder is a list of all the modules installed on a particular server.
- You should disable following modules if you do not need them:

PHP

SSL

Rewrite

Perl

Python

Rack / Ruby / Passenger

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

## 2. Limit the Number of Apache Processes.

### One word question answer

#### Question

1. From where we can Unload Unneeded Modules?
2. Steps to optimize web service\_\_\_\_\_

#### Answer

/etc/apache2/mods-enabled  
Unload Unneeded Modules,  
Limit the Number of Processes

Topic: 8 Explain how to configure firewall in ubuntu.

Mark 3

Ans.

#### Detailed:

- Firewall is considered as the first method of defense in securing your cloud server.
- Ubuntu includes its own (built-in) Firewall known as UFW.
- UFW stands for “Uncomplicated Firewall”.
- Its main goal is to provide an easy-to use interface.
- The firewall is disabled by default. To enable the firewall, run the following commands from a terminal.

#### **sudo ufw enable**

- To install Firewall:

**sudo aptitude install ufw**

Or

**sudo apt-get install ufw**

- To check the status, use following command:

**sudo ufw status**

#### **Setup rules**

- You should first define some rules for the firewall for allowing and denying connection.
- To set the defaults used by UFW, use the following commands

**sudo ufw default deny incoming**

And

**sudo ufw default allow outgoing**

- If you want to deny the outgoing request also, use the following command:

**sudo ufw default deny outgoing**

- Enable the SSH connection, use this command:

**CS –22 Operating Systems Concepts With Unix / Linux**

99 | Page

# GEETANJALI GROUP OF COLLEGES

(Affiliated to Saurashtra University)

**sudo ufw allow ssh**

## Deleting rules

- There are two option to delete rules.

**sudo ufw delete allow ssh**

- If you need to reset your cloud server's rules to their default settings, you can do this :

**sudo ufw reset**

## One word question answer

### Question

1. What is UFW?
2. Which command is used to enable the firewall?
3. How to enable the SSH connection?

### Answer

Uncomplicated Firewall  
sudo ufw enable  
sudo ufw allow ssh

**Topic: 9 Explain WINE.**

**Mark 3**

**Ans.**

### Detailed:

- As we know that Linux is a great operating system, If you like a Window game or other app, you need to use Wine to run it right on your Ubuntu desktop.

### Installing Wine

- You will find the Wine download in the Ubuntu Software Center in all version.
- Download and install it.

### Running an Application

- After successful installation of Wine, you can download an application's EXE file to install it by simply double-clicking it.
- You can also use terminal to install an application
- Use following command:

**wine /path/to/application.exe**

- After installation of the application, you can find it's shortcut icon on Desktop as we find it on Windows OS.

## One word question answer

### Question

1. Wine is used for.

### Answer

Window game or other app