PGScIT S4 03
Exam. Code: M4 LSA

Linux System Administration

SEMESTER-IV

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

BLOCK-1

KRISHNA KANTA HANDIQUI STATE OPEN UNIVERSITY

Subject Experts

Prof. Anjana Kakati Mahanta, Gauhati University

Prof. (Retd.) Pranhari Talukdar, Gauhati University

Prof. Shikhar Kumar Sarma, Gauhati University

Dr. Jyotiprokash Goswami, Assam Engineering College

Programme Coordinator

Dr. Ridip Dev Choudhury, KKHSOU

Course Coordinators

Dr. Ridip Dev Choudhury, KKHSOU

Dr. Tapashi Kashyap Das, KKHSOU

Dr. Sruti Sruba Bharali, KKHSOU

SLM Preparation Team

UNITS	CONTRIBUTORS
1,2,3,4,5	Dr. Gautam Chakraborty, NERIM
6	Mr. Saurabh Sutradhar, Royal Global University
7,8	Dr. Ridip Dev Choudhury, KKHSOU

Editorial Team

Structue, Format & Graphics	**:**	Dr. Ridip Dev Choudhury, KKHSOU
Content	**:**	Dr. Khurshid Alam Borbora, KKHSOU

September, 2022 **ISBN NO :**

This Self Learning Material (SLM) of the Krishna Kanta Handiqui State Open University is made available under a Creative Commons Attribution-Non Commercial-Share Alike 4.0 License (international): http://creativecommons.org/licenses/by-nc-sa/4.0/

Printed and published by Registrar on behalf of the Krishna Kanta Handiqui State Open University.

Headquarters : Patgaon, Rani Gate, Guwahati - 781017
City Office : Resham Nagar, Khanapara, Guwahati- 781022 **Web: www.kkhsou.in**

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

Linux System Administration

Block-1

DETAILED SYLLABUS

Page No.

UNIT 1: Introduction to System Administration **7-18**

Introduction to System Administration, Role of System Administrator, Qualities of good System Administrator, System Administration Common administrative tasks, identifying administrative files - configuration and log files

UNIT 2: Introduction to LINUX Operating System **19-35**

Basic idea on Proprietary, Open Source, Free Software etc, Basic Features of the Linux operating system, Introduction of Various Linux Distribution (Red Hat Enterprise Linux, Cent OS, Fedora Projects, Debian Linux, Ubuntu, SUSE etc.)

UNIT 3: Installation of LINUX Operating System **36-65**

Installation Requirements, Partitioning the Hard drive in Linux, Installing the Linux system: Using Live CD, Virtual Machine, Direct Installation, Installing and Configuring software in Linux

UNIT 4: LINUX Kernel **66-82**

Linux kernel and device drivers, System Startup and Shutdown, Standard I/O, Standard error, Redirection and Piping

UNIT 5: Basics of LINUX File System **83-106**

Basics of Linux file system - File system types (ext3, ext4, xfs, jfs, ReiserFS, iso9660 etc.), Boot block, Super block, Inode table, Data blocks, How Linux access files, storage files, Linux standard directories, LILO, GRUB Boot Loader, three basic types of files (ordinary or regular, special or device and directory), I-nodes and file attributes, init and run levels

UNIT 6: File organization **107-123**

Files and File System (File Types and Permissions, Links, Size and Space, Date and Time); Working with Files: Reading Files, Searching for files, Copying, Moving, Renaming, Deleting, Linking, and Editing Files; Absolute and Relative

path names, File system Mounting and Unmounting, Organization of the file tree, Standard directories and their contents, Disk related commands, checking disk free spaces.

UNIT 7: LINUX File Handling Commands **124-136**

Files and Directory handling Commands - ls, cd, cp, mv, rm, mkdir, rmdir, pwd, Commands for Creating and Viewing ordinary files - cat, more, less

UNIT 8: LINUX Filter Commands **137-154**

Filter Commands - wc, head, tail, cut, tr, top, grep (with regular expressions),. Other commands -tar, unzip, nice, kill, netstat. Setting user and group ownership of files and Access permissions - chmod, chown, chgrp commands

COURSE INTRODUCTION

This is a course on **Linux System Administration.** Linux is an open-source operating system. The Linux OS was developed by Linus Torvalds in 1991. Nowadays, Linux is the fastest-growing operating system. Linux system administration is a process of setting up, configuring, and managing a computer system in a Linux environment. This course is designed to describe most of the skills needed for Linux system administration work.

The course consists of two blocks.

Block 1 deals with the basic concepts of Linux Operating System. The installation processes of the Linux operating system are explained. The file and directory structure of Linux OS are explained. At the end of this block, various file handling and filter commands are discussed.

Block 2 discusses about the Linux shells. It also discusses the processes, printing jobs and establishment of networking in Linux OS. In the last part of this block, the concept of network protocols and basic network security issues are explained.

BLOCK INTRODUCTION

This is the first block of the course **Linux System Administration.** This block contains eight units.

Unit 1 describes the role of system administrator and the common administrative task of a system administrator.

Unit 2 discusses about open Source and Free Software. It also describes various Linux Distributions like Red Hat Enterprise Linux, Cent OS, Fedora Projects, Debian Linux, Ubuntu, and SUSE Linux

Unit 3 describes step by step installation process of Linux operating system.

Unit 4 describes the Linux kernel and the system startup and shutdown procedures.

Unit 5 describes various file system types like ext3, ext4, xfs etc.

Unit 6 discusses how to work with Files. File system mounting and unmounting, Organization of the file tree, Standard directories and their contents are also discussed in this unit.

Unit 7 discusses various file handling commands with examples.

Unit 8 deals with the various Linux filter commands.

Each unit of these blocks includes "CHECK YOUR PROGRESS" questions. These have been designed to self-check your progress of study. It will be helpful for you if you solve the problems put in these boxes immediately after you go through the sections of the units and then match your answers with "ANSWERS TO CHECK YOUR PROGRESS" given at the end of each unit.

UNIT 1: INTRODUCTION TO SYSTEM ADMINISTRATION

UNIT STRUCTURE

1.1 Learning Objectives

1.2 Introduction

1.3 Role and Power of System Administrator

1.4 Qualities of Good System Administrator

1.5 Common Administrative Tasks

1.6 Identifying, Configuration and Log Files

1.7 Let Us Sum Up

1.8 Further Readings

1.9 Answer to Check Your Progress

1.10 Model Questions

1.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- describe the need of the system administrator
- explain the usage of su command
- describe the roles and privileges of a system administrator
- describe the skills required for being a system administrator
- learn some system administrative tasks
- learn about the LINUX configuration and log files

1.2 INTRODUCTION

System administration, required in a multiuser system, is the task of administering and monitoring the system when the system is accessed by multiple users. It is usually handled by a trusted person who is responsible for the administration of the whole system. The main job of a system administrator is to ascertain that there is little to complain about the system's performance or service availability. The system administrator operates in super user mode which is called root. The super user has the privilege to

issue any command, access any file, and perform all the functions in the system. root is the user-id for the super user to log on to the multiuser system. The use of root command must be limited due to the following reasons:

- Inexperienced users can cause severe damage to the system.
- The use of root command for non-privileged tasks is needless and may lead to attack.
- Since root can look at anyone's files, it may violate security and privacy of the system.

The screen shot of super user login into the system is depicted in figure 1.1.

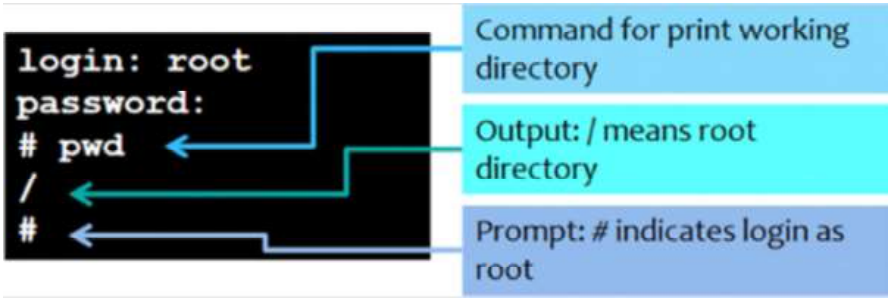


Fig. 1.1 Super user login

The fundamental tasks of the super user includes starting-up & shutting-down the system, creating accounts for new users with necessary environments. Administrative tasks also involve installation of new software, distributing user disk space, taking regular back-ups, keeping system logs, ensuring secure operations and providing network services and web access. The su command is used for becoming super user. It switches from non-privileged account to the super user account. The who am i command is used to view the login name.

The optional argument '-' is used to provide an environment similar to what the user had logged in directly. The su command without '-' argument executes command as root but retains user's current environment.

By using the command shutdown, super user can shut-down or restart the system. The option -h with shutdown command is used for a complete shutdown and the -r option is for restarting the system.

1.3 ROLE AND POWER OF THE SYSTEM ADMINISTRATOR

The System Administrator is responsible for installing the server of the system, updating configuration, operating and maintaining the hardware and software of the system. He/she has to manage failures in the system, troubleshoot and maintain the servers, the user accounts, security, etc. He/she has to accomplish the task of managing the disk space as well as the backup. Also, the job of system administrator is changing the contents of the file or its attributes like its ownership or permission etc. He/she also needs to reset new password and constantly monitor the special services and security of the servers. They are expected to remain informed with the recent knowledge in Linux system in order to assess the hardware and software technologies.

The key responsibilities that need to be taken care of by a Linux System Administrator are:

- To setup and maintain user account,
- To monitor system performance,
- To verify that peripherals are working properly,
- To quickly arrange to repair for hardware in occasion of hardware failure,
- To create and manage file systems,
- To install software and patching firmware and software,
- To create a backup and recovery policy,
- To monitor network communication,
- To update system as soon as new version of OS and application software,
- To Setup security policies for users,
- To manage network administration and Database administration

Further, Linux system administrator must participate in various cross training activities, so as to attain a comfort level that supports the database. They might also be involved in training the subordinate team members in system administration and software deployment responsibilities. They retain proper documentation of all the procedures, configurations, programs, etc.

They offer valuable support to the users by answering their technical queries. Many-a-time, they also provide the customer support through e-mail, telephone, knowledge-based programs and other resources.

The system administrator has the power to modify the contents or the attributes of any files owned by other users, initiate or kill any process that are essentially required for running the system, change the password of any user without even entering the old one, setting the system clock, control user's access to scheduling service and networking service.

1.4 QUALITIES OF GOOD SYSTEM ADMINISTRATOR

To become a successful Linux system administrator, one should be a bootstrapper who has high motivation and detail orientation capacity, and also be flexible and ready to follow the instructions given by their superior. The person should be prepared to acquire recent information/ updates and apply the up-to-date technologies that would help the organization to grow. He/she should hold excellent interpersonal skills and should be comfortable working in a team. Further, the quality of identifying the issues related to system and resolving them in an effective way should also be with the individual. He/she must be able to work well under pressure and able to respond and prioritize work according to the stringent deadlines. In concise, a good system administrator should have the following skills:

- Problem-solving skill:
  - Taken up the matter when a computer system malfunctions.
  - Quickly and correctly diagnose the crisis and produce the best solution.
- Analysing skill:
  - Understand the behavior of software.
  - Deployment of software.
  - Troubleshoot the Problems if occur.
- Programming skill:
  - Have the knowledge of different types of programming languages.
  - Scripting or automation of routine tasks.

- Soft-skills:
  - Users to the system should feel comfortable and know that their problems will be fixed in time.
  - Good and healthy interaction among colleagues/team members etc.

1.5 COMMON ADMINISTRATIVE TASKS

User account management: The system administrator can add accounts for new users and removes the accounts of users those are no longer active. The linux commands used for this are:

- useradd
- usermod
- userdel

useradd: Using this command, the administrator can create an account to represent a user who will be working on the system. Each user must authenticate to Linux with an identity that can be used to control their resource access and consumption. User accounts' detail are stored in the **/etc/passwd** file. It is used in the following way:

$ sudo useradd <accountname>

Also, we should not forget to set a password for the created account by using the passwd command.

$ sudo passwd <accountname>

Few common options are used with useradd command:

- --create -home (-m): Adds a home directory.
- --shell (-s): Sets the user's preferred shell if it's different from /bin/bash
- --uid (-u): Specifies a particular user ID (UID)
- --comment (-c): Populates the comment field (usually with the user's full name enclosed in quotes)

usermod:

This command is used to modify the settings of an existing user account. Modifications are necessary when users change their names, request different shells, or need updated password information.

Standard options for usermod includes:

- --comment (-c): Modifies the comment field.
- --home (-d): Modifies home directory information.
- --expiredate (-d): Changes account-expiration settings.
- --login (-l): Modifies the username.
- --lock (-L): Locks a user account.
- --unlock (-U): Unlocks a user account.

Syntax to use this command is:

$ sudo usermod -c <newName> <oldName>

userdel:

We can remove/delete an account representing a user whose role has changed or is no longer with the organization.

To delete a particular account, type:

$ sudo userdel <accountname>

However, before deleting an account, we also have to consider the resources such as the home directory or system mail related to that account.

Few common options associated with userdel that address these resources are:

- --force (-f): Deletes the account (including mail and home directory), even if the user is still logged in
- --remove (-r): Deletes the account (including mail and home directory), but the user must be logged out

Performing backups: The term backup refers to making copies of data that may be used to restore the original if data loss happens. In Linux, tar command is used to backup/restore files. It can also be used to archive a group of files into another file. The following options are used with tar command.

-c to create an archive

-t to display files on archive

-x to extract from archive

-v verbose -display filenames

-z to compress files

-f device name or filename to write to

There are some examples of using tar command as given below:

tar -cvf /dev/st0 backup the entire system to tape

tar -tvf /dev/st0 view the contents of the tape

tar -xvf /dev/st0 restore the entire contents of the file

1.6 IDENTIFYING, CONFIGURATION AND LOG FILES

A Configuration File, also known as a config file, is a unique type of plain text file that is used to control the operations of a program, utility and process. Linux configuration files contain the settings and instructions for different systems, utilities, applications and processes. As the name infers, configuration files contain information about the structure or arrangement of specific parts of the system. They are kept under the /etc directory.

These files are not essentially of any direct use to users. However, they do affect the way a system functions. For example, the fstab file is the filesystem configuration file that includes all the disk partitions and the purpose of each of the partitions. For regular partitions, the conventional mount point may also be listed. The /etc/fstab file is used by mount and other such commands. It is what allows file systems to be mounted automatically at boot time rather than having to mount them all by hand, one at a time.

As a part of system administration tasks, administrator has to frequently modify these files. For instance, every time a new user is added, the /etc/passwd and /etc/group configuration files must be modified.

/etc/passwd is a password file in which each line holds all the information of a single user. A line consists of seven fields, separated by a colon.

- Login name
- Encrypted password
- UID
- Default GID
- GCOS field including full name, office location, extension number, and home phone.

- Home directory
- Login shell

User-id's (UID's) are mapped to the names in this file. Every time a new user is added their information must be added to this file.

/etc/group contains the names of valid groups and the usernames of their members. This file is owned by root and only root may modify it. When a new user is added information on what groups they are a member of must be added here. Group-id's (GID's) are mapped to the group names kept in this file.

Other configuration files include:

/etc/inetd.config, /etc/hosts, /etc/inittab, /etc/hosts.equiv, /etc/export, and /etc/mtab.

/etc/inetd.config is the Internet daemon file. It holds the instructions used by /etc/inetd to manage the network service daemons.

/etc/hosts is the TCP/IP network configuration file. It is accessed by all commands that use the network. The file contains a list of IP addresses, hostnames and hostname aliases.

/etc/inittab is a system configuration file that helps regulate the operation of serial interfaces.

/etc/hosts.equiv contains a list of TCP/IP "trusted" hosts. If machine A is a trusted host of machine B, then all users on machine A may rlogin to machine B without supplying a password.

/etc/export lists all the file systems that are available to other machines via NFS.

/etc/mtab holds information on all mounted file systems.

Linux Log Files

Log files are the set of records that Linux system keeps so that the system administrator keep a track of important events. They contain messages about the server, including the kernel, services and applications running on it. These files can be helpful for troubleshooting system issues.

Most Linux log files are stored in a plain ASCII text file and are kept in the /var/log directory and subdirectory. Logs are generated by the Linux system daemon log, syslogd or rsyslogd.

Viewing Logs:

- First, open the Linux terminal as a root user. This will enable root privileges.
- Use the following command to see the log files:

  $ cd /var/log
- To view the logs, type the following command:

  $ ls

This command displays all Linux log files, such as kern.log and boot.log. These files contain the necessary information for the proper function of the operating system.

Log files are accessed using root privileges. By definition, root is the default account that can access to all Linux files.

Use the following command to access the respective file:

$ sudo less [logname].log

This command displays a timeline of all information related to that operation.

It is be noted that log files are stored in plain text and hence, they can be viewed by using the following standard commands:

zcat - Displays all the contents of logfile.gz

zmore - lists the file in pages, without decompressing the files

zgrep - Searches inside a compressed file

grep - Finds all occurrences of a search term in a file or filter a log file

tail - produces the last few lines of files

Log files can provide a lot of information about the operations of a system. Most directories can be grouped into one of four categories:

- System Logs
- Event Logs
- Application Logs
- Service Logs

  Many of these logs can be located in the var/log subdirectory.
- System Logs

  Systems log files contain the most significant amount of information

about system functionality. The most common log files include:

- /var/log/boot.log: System Boot log (the boot log stores all information related to booting operations)
- /var/log/auth.log: Auth logs (the authentication log stores all authentication logs, including successful and failed attempts)
- /var/log/httpd/: Apache access and error logs
- /var/log/mysqld.log: MySQL database server log file
- /var/log/debug: Debug logs (the debug log stores detailed messages related to debugging and is useful for troubleshooting specific system operations)
- /var/log/daemon.log: Daemon logs (the daemon log contains information about events related to running the Linux operation)
- /var/log/maillog: Mail server logs (the mail log stores information related to mail servers and archiving emails)
- /var/log/kern.log: Kernel logs (the kernel log stores information from the Ubuntu Linux kernel)
- /var/log/yum.log: Yum command logs
- Event logs

An event log refers to a "log book" that is analyzed and monitored for higher level network intelligence. It can capture many different types of information. For example, it can capture all logon sessions to a network, along with account lockouts, failed password attempts, etc.

- Application Logs

Application logs store information relevant to any application that is executed. These includes error messages, signs of system compromise, and browser identification string. Log files that fall into this category include CUPS Print System logs, Rootkit Hunter log, Apache HTTP server logs, Samba SMB server logs, and X11 server log.

- Service logs

Service logs provides diagnostic information about the resources in your occupancy. When you enable logging on resources, you receive information about the resource in a log file. This information allows you to analyze, optimize, and troubleshoot your resources.



CHECK YOUR PROGRESS

Q 1: Choose the correct options from the followings:

i) The process of starting up a computer is known as

a) Boot Loading b) Boot Record c) Boot Strapping

d) Booting

ii) The shell used for Single user mode shell is:

a) bash b) csh c) ksh d) sh

iii) Which is the only partition mounted in Single user mode

a) boot b) usr c) root d) tmp

iv) Which command is used to restart the Linux system?

a) shutdown -h b) shutdown -r c) restart

d) none of these

v) Which Linux command is used for becoming a super user?

a) su b) man c) who d) None of these



1.7 LET US SUM UP

- System administrator is the person who is responsible for setting up and maintaining the system or server.
- The root account has full (unrestricted) access, so he/she can do anything with the system.
- The system administrator is responsible for user administration, maintaining system, verify whether peripherals are working properly or not, quickly arrange repair for hardware on the occasion of hardware failure, monitor system performance, create file systems, install software.
- Administrator can create a backup and recovery policy.
- The system administrator also monitors network communication, updates system as soon as new version of OS and application software comes out, implements the policies for the use of the computer system and network setup security policies for users.
- A system admin must have a strong grasp of computer security, e.g.

firewalls and intrusion detection systems.

- A system administrator should have problem solving skill, analyzing skill, programming skill and soft skills.
- A configuration file, also known as a config file, is a unique type of plain text file that is used to control the operations of a program, utility or process.
- Log files are a set of records that Linux keeps for the system administrators to keep track of important events.



1.8 FURTHER READINGS

1) S. Das (2012). Your UNIX/Linux: The Ultimate Guide. (Third Edition). (Science/Engineering/Math). McGraw-Hill.
2) Christopher Negus (2012). Linux Bible. (8th Edition). Wiley.



1.9 ANSWERS TO CHECK YOUR PROGRESS

Ans to Q No 1: i) c ii) d iii) c iv) b v) a



1.10 MODEL QUESTIONS

Q 1: What is called System administration and why is it required for multiuser system?

Q 2: Explain the responsibility of a system administrator in detail.

Q 3: Mention the skills required for a good administrator.

Q 4: Write the application of sudo and who am I commands.

Q 5: What is the purpose of configuration file and log file?

Q 6: What are the Linux commands used for user management?

*** ***** ***

UNIT 2:INTRODUCTION TO LINUX OPERATING SYSTEM

UNIT STRUCTURE

2.1 Learning Objectives

2.2 Introduction

2.3 Basic idea on Proprietary, Open Source, Free Software

2.4 Linux Operating System

- 2.4.1 Components of Linux System
- 2.4.2 Architecture of Linux operating System

2.5 Basic Features of Linux Operating System

2.6 Popular Linux Distributions

- 2.6.1 Red Hat Enterprise Linux/CentOS
- 2.6.2 Ubuntu
- 2.6.3 Debian
- 2.6.4 Fedora
- 2.6.5 SUSE

2.7 Let Us Sum Up

2.8 Further Readings

2.9 Answer to Check Your Progress

2.10 Model Questions

2.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- learn about proprietary, open source and free software
- define an operating system
- describe the Linux Operating System
- explain the internal architecture of Linux
- list the basic features of the Linux Operating System
- get brief overview of various Linux distributions like Fedora, Debian, SUSE, etc.

2.2 INTRODUCTION

In the first unit, we have learnt about the need of a system administrator in multiuser operating system. The role and power of a system administrator are also discussed in that unit. In this unit, we concentrate on about LINUX operating system. An operating system is a vital part of a computer system. In general, a computer system works in the presence of three general components: the hardware, the operating system and the applications. The hardware refers to various electronic as well as electro-mechanical parts such as a central processing unit (CPU), a keyboard, a hard drive, a mouse, a printer, etc. Applications are those which are used to operate the computer in order to perform the desired task. The operating system is the component that manages and controls the hardware on one side and manages the applications on other side.

2.3 BASIC IDEA ON PROPRIETARY, OPEN SOURCE AND FREE SOFTWARE

Proprietary Software

Proprietary Software, also known as Closed Source Software, is the software product that legally remains the property of an organisation, a group, or an individual who developed it. The organisation that owns the rights to the product usually does not release the source code, and may insist that only those who have purchased a special licence key can use it. This software is under restrictive copyright licensing and the source code is usually hidden from the users. The products developed and marketed by Microsoft such as Windows 2010, Oracle DBMS, etc. are some of the examples of proprietary software.

Free Software

Free Software, also called Freeware, is licensed at no cost or for an optional fee. It is usually closed source. According to Richard Stallman's definition and adopted by the Free Software Foundation (FSF), free software can be defined as a matter of liberty not price, and it upholds the Four

Essential Freedoms which include the following:

- The freedom to run the program as you wish, for any purpose (freedom 0).
- The freedom to study how the program works, and change it so it does your computing as you wish.
- The freedom to redistribute copies so you can help others.
- The freedom to distribute copies of your modified versions to others.

Open-Source Software

Open-source software is free and openly available to everyone. People who create open-source products, publish the code and allow others to use and modify it. Communities of programmers often work together to develop the software and to support users. Open-source products are usually tested in public by online contributors.

Large companies such as Twitter, Facebook and the BBC make use of open-source technology. For example, the BBC makes use of MySQL and it creates open-source software, such as the program to improve the compatibility of iPlayer on smart TVs.

2.4 LINUX OPERATING SYSTEM

Linux Operating System is one of the UNIX-like operating system, which is designed to offer a free or low-cost operating system for personal computer users. It has gained the status of a fast performing and very efficient operating system. This is a remarkably considered a complete operating system, which includes a GUI (Graphical User Interface), TCP/IP, the Emacs editor, X Window System, etc. Linux is based on the Linux kernel, an operating system kernel first released on September 17, 1991 developed by Linus Torvalds, a then-student at the University of Helsinki. Linux is typically packaged in a Linux distribution which includes the Linux kernel and supporting system software and libraries, many of which are provided by the GNU Project. Many Linux distributions use the word "Linux" in their name, but the Free Software Foundation uses the name GNU/Linux to emphasize the importance of GNU Software. Due to its open-

source licensing, Linux is freely available to anyone. However, the trademark on the name "Linux" rests with its creator, Linus Torvalds. The source code for Linux is under copyright by its many individual authors, and licensed under the GPLv2 license.

Popular Linux distributions include Debian, Fedora, CentOS and Ubuntu. Commercial distributions include Red Hat Enterprise Linux and SUSE Linux Enterprise Server. Desktop Linux distributions include a windowing system such as X11 or Wayland, and a desktop environment such as GNOME or KDE Plasma. Distributions intended for servers include a solution stack such as LAMP. Linux was originally developed for personal computers based on the Intel x 86 architecture, but later it has been ported to more platforms. Linux is the leading operating system on servers and other big iron systems such as mainframe computers and is the only OS used on TOP 500 supercomputer.

Further, Linux also runs on embedded systems, i.e. devices whose operating system is typically built into the firmware and is highly tailored to the system. This includes routers, automation controls, televisions, digital video recorders, video game consoles, and smart watches. Many smartphones and tablet computers run Android and other Linux derivatives. Because of the dominance of Android on smart phones, Linux has the largest installed base of all general-purpose operating systems.

Linux is one of the most prominent examples of free and opensource software collaboration which offers the source code be used, modified and distributed-commercially or non-commercially by anyone under the terms of its respective licenses, such as the GNU General Public License.

2.4.1 Components of Linux System

Linux Operating System is comprised of the following three components as shown in figure 2.1:

- **Kernel:** Kernel is the central part of Linux OS. All major activities of Linux OS are executed under the control of Kernel. It consists of various modules and by using these modules kernel interacts

directly with the underlying hardware. Kernel provides the required abstraction to hide low level hardware details to system or application programs.

- **System Library:** System libraries are special functions or programs by using which application programs or system utilities can access features of the Kernel. These libraries implement most of the functionalities of the operating system.
- **System Utility:** System Utility refers to specialized programs which are responsible for doing specialized individual level tasks.

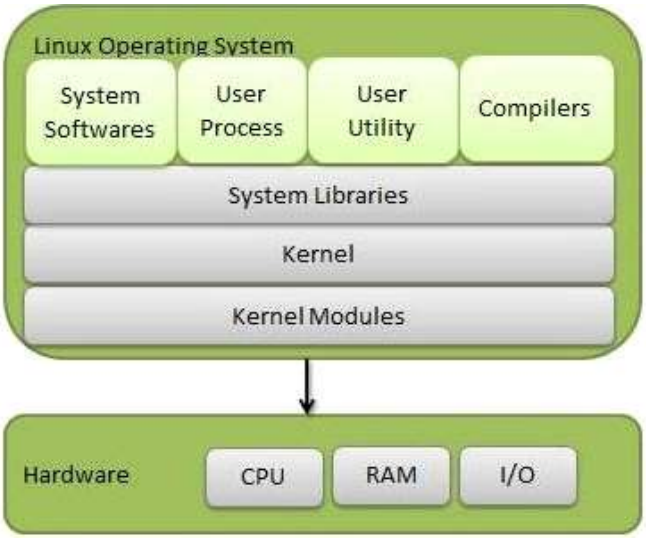


Fig. 2.1 Components of Linux OS

Kernel Mode vs User Mode

Kernel code executes in a special privileged mode called kernel mode with full access to all resources of the computer. This code represents a single process, executes in single address space and does not require any context switch. Kernel runs each process and provides system services to processes. It also provides protected access to hardware for processes.

in User Mode which does not have access to system hardware and kernel code. User programs/utilities use System Libraries to access Kernel Functions to get system's low-level tasks.

2.4.2 Architecture of Linux operating System

The architecture of the Linux operating system, depicted in figure 2.2, consists of four layers.

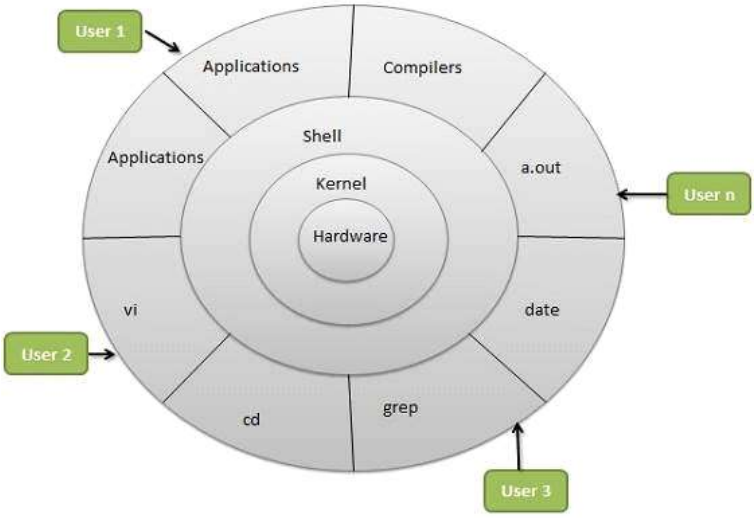


Fig. 2.2 Architecture of Linux Operating system

- Hardware layer - It consists of all peripheral devices such as RAM, HDD, CPU, etc.
- Kernel - It is the core component of Operating System, interacts directly with hardware, and provides low level services to upper layer components.
- Shell - It acts as an interface to kernel, hiding complexity of kernel's functions from users. The shell accepts commands from the user and sends to the kernel to execute.
- Utilities - Utility programs provide the user most of the functionalities of an operating systems.

2.5 BASIC FEATURES OF THE LINUX OPERATING SYSTEM

Linux Operating System provides the following basic features:

- **Portability:** This feature allows software to work on different types of hardware in the same way. Linux kernel and application programs support 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 provides a multiuser environment which means multiple users can access system resources like memory / RAM / application programs at the same time.
- **Multiprogramming:** This feature allows multiple applications to run at the same time.
- **Hierarchical File System:** Linux provides a standard tree form file structure in which system files/ user files are arranged.
- **Security:** It provides security in three ways namely authenticating (by assigning password and login ID), authorization (by assigning permission to read, write and execute) and encryption (converts file into an unreadable format).
- **Live CD/USB:** Almost all Linux distributions have Live CD/USB feature by which user can run the OS even without installing it on the system.
- **Graphical user interface (X Window System):** Though Linux is usually considered as a command-based OS, but Linux has packages which can be installed to make the whole OS graphic based.
- **Application Support:** Linux has its own software repository from where users can download and install thousands of applications just by issuing a command in Linux Terminal or Shell.

2.6 POPULAR LINUX DISTRIBUTIONS

Technically, Linux is an Operating System Kernel, the core part of an Operating System. The Linux kernel is commonly bundled with many other components to form a complete Linux Distribution. When most people refer to Linux, they are talking about this complete set of programs needed to run a computer, not just the kernel. Because Linux is an open-source operating system, anyone can extend and distribute the system. Several organizations put together the Linux kernel along with a host of programs

and their own special setup utility and distribute the whole package. These are called Linux distributions. Linux users usually obtain their operating system by downloading one of the Linux distributions, which are available for a wide variety of systems ranging from embedded devices (for example, OpenWrt) and personal computers to powerful supercomputers (for example, Rocks Cluster Distribution).

A typical Linux distribution comprises a Linux kernel, GNU tools and libraries, additional software, documentation, a window system (the most common one the X Window System), a window manager, and a desktop environment. Most of the included software is free and open-source software made available both as compiled binaries and in source code form, which allow modifications to the original software. Usually, Linux distributions optionally include some proprietary software that may not be available in source code form, such as binary blobs required for some device drivers. A Linux distribution may also be defined as a particular collection of application and utility software (various GNU tools and libraries, for example), packaged together with the Linux kernel in such a way that its functionalities meet the needs of many users. The software is usually adapted to the distribution and then packaged into software packages by the distribution's maintainers. The software packages are available online in so-called repositories, which are storage locations usually distributed around the world.

Some commercially backed Linux distributions include Fedora Linux, Red Hat Enterprise Linux, openSUSE (SUSE) and Ubuntu, etc. and entirely community-driven distributions include Debian, Slackware, Gentoo and Arch Linux, etc.

2.6.1 Red Hat Enterprise Linux/CentOS

Red Hat Enterprise Linux (RHEL) is a commercial Linux distribution intended for servers and workstations. Red Hat Enterprise Linux is a Linux distribution developed by Red Hat and targeted toward the commercial market. Red Hat Enterprise Linux is released in server

versions for x86-64, Power Architecture, ARM64, and IBM Z, and a desktop version for x86-64. All of Red Hat's official support and training, together with the Red Hat Certification Program, focuses on the Red Hat Enterprise Linux platform. Red Hat Enterprise Linux is often abbreviated to RHEL. The first version of Red Hat Enterprise Linux originally came into the market as "Red Hat Linux Advanced Server". In 2003, Red Hat rebranded Red Hat Linux Advanced Server to "Red Hat Enterprise Linux AS", and added two more variants, Red Hat Enterprise Linux ES and Red Hat Enterprise Linux WS.

Red Hat uses strict trademark rules to prevent their official Red Hat Enterprise Linux software from being redistributed, and yet freely provides its source code. Third-party derivatives can be built and redistributed by stripping away the non-free components like Red Hat's trademarks. CentOS is a community project that uses the Red Hat Enterprise Linux code, removes all Red Hat's trademarks, and makes it available for free use and distribution. It's a free version of RHEL, so it's good if you want a stable platform that will be supported for a long time. Nowadays, CentOS is now part of Red Hat itself. A screen shot of Desktop window of RHEL is shown in Figure 2.3.

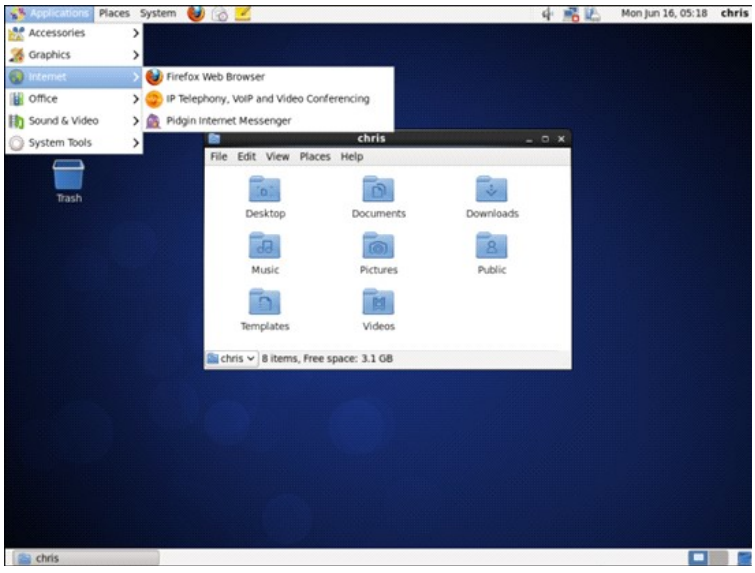


Fig. 2.3 Desktop window of RHEL

2.6.2 Ubuntu

Ubuntu is a free and open-source Linux distribution based on Debian. Ubuntu is officially released in three editions: Desktop, Server, and Core. Ubuntu is a popular operating system for cloud computing, with support for OpenStack. Ubuntu is developed by Canonical who provide security updates and support for each Ubuntu release, starting from the release date and until the release reaches its designated end-of-life (EOL) date. Ubuntu is named after the African philosophy of ubuntu, which Canonical translates as "humanity to others" or "I am what I am because of who we all are". It came into existence in 2004 by Canonical and quickly became popular for it's a graphical Linux desktop without the use of command line. It is the most well-known Linux distribution. Ubuntu is a next version of Debian and easy to use for beginners. It is comprised of lots of pre-installed apps and easy to use repositories libraries. A screenshot of Desktop window of Ubuntu is depicted in Figure 2.4.

A default installation of Ubuntu contains a wide range of software that includes LibreOffice, Firefox, Thunderbird, Transmission, and several lightweight games such as Sudoku and chess. Many additional software packages are accessible from the built in Ubuntu Software as well as any other APT-based package management tools. Many additional software packages that are no longer installed by default, such as Evolution, GIMP, Pidgin, and Synaptic, are still accessible in the repositories still installable by the main tool or by any other APT-based package management tool. Cross distribution snap packages and flatpaks are also available, that both allow installing software, such as default file manager is GNOME Files, formerly called Nautilus.



Fig. 2.4 Desktop window of Ubuntu

Ubuntu operates under the GNU General Public License (GPL) and all of the application software installed by default is free software. In addition, Ubuntu installs some hardware drivers that are available only in binary format, but such packages are clearly marked in the restricted component. Ubuntu aims to be secure by default. To increase security, the sudo tool is used to assign temporary privileges for performing administrative tasks, which allows the root account to remain locked and helps prevent inexperienced users from inadvertently making catastrophic system changes or opening security holes. Polkit is also being widely implemented into the desktop. Most network ports are closed by default to prevent hacking.

A built-in firewall allows end-users who install network servers to control access. Ubuntu also supports full disk encryption as well as encryption of the home and private directories.

2.6.3 Debian

Debian is a Unix-like operating system having entirely of free software. Ian Murdock started the Debian Project on August 16, 1993. Debian 0.01 was released on September 15, 1993, and the first stable version, 1.1, was marketed on June 17, 1996. The Debian stable branch is the most popular edition for personal computers and network servers, and is used as the basis for many other distributions.

Debian is one of the earliest operating systems based on the Linux kernel. The project work of Debian is carried out over the Internet by a team of volunteers guided by the Debian Project Leader and three foundational documents: the Debian Social Contract, the Debian Constitution, and the Debian Free Software Guidelines. New distributions are updated continually, and the next candidate is released after a time-based freeze. Debian has been openly developed and freely distributed according to the principles of the GNU Project founded by Richard Stallman. This philosophy drew the support of the Free Software Foundation, which sponsored the project from November 1994 to November 1995. When the sponsorship ended, the Debian Project formed Software in the Public Interest to continue financially supporting development. Debian has access to online repositories that contain over 51,000 packages making it the largest collection of software in the world. Debian officially contains only free software, but non-free software can be downloaded and installed from the Debian repositories. Debian includes popular free programs such as LibreOffice, Firefox web browser, Evolution mail, K3b disc burner, VLC media player, GIMP image editor, and Evince document viewer. Debian is a popular choice for servers. Desktop window of Debian Linux is presented in Figure 2.5.



Fig. 2.5 Desktop window of Debian Linux

2.6.4 Fedora

Fedora is a Linux distribution developed by the community supported Fedora Project and sponsored by Red Hat. Fedora contains software distributed under various free and open-source licenses and aims to be on the leading edge of such technologies. Fedora is considered the upstream source of the commercial Red Hat Enterprise Linux distribution.

Since the release of Fedora 21, different editions of Fedora are currently available: Workstation, focused on the personal computer, Server for servers, and Atomic focused on cloud computing.

Fedora has a reputation for focusing on innovation, integrating new technologies early on and working closely with upstream Linux communities. Making changes upstream instead of specifically for Fedora ensures that the changes are available to all Linux distributions. The default desktop environment in Fedora

is GNOME and the default user interface is the GNOME Shell. The other desktop environments include KDE Plasma, Xfce, LXDE, MATE and Cinnamon.

Fedora comes preinstalled with a wide range of software such as LibreOffice and Firefox. Additional software is available from the software repositories and can be installed using the DNF package manager or GNOME Software. Additionally, extra repositories can be added to the system, so that software not available in Fedora can be installed easily.

2.6.5 SUSE

SUSE Linux Enterprise Server is an operating system which is built on top of the free and open-source Linux kernel. It is distributed with system and application software from other open-source projects.

SUSE Linux is of German origin, basically an acronym of "Software und System-Entwicklung" (software and systems development), and was mainly developed in Europe. The first version was released in early 1994, making SUSE one of the oldest existing commercial distributions. It is known for its YaST configuration tool.

SUSE Linux is available under two brands, openSUSE and SUSE Linux Enterprise. openSUSE is a free, community distribution driven by the openSUSE Project. It includes some of the latest Linux technologies and is designed for home users and enthusiasts. SUSE Linux Enterprise is the tested and certified open-source solution for major enterprises. openSUSE is a freely available, community project that releases versions on a comparatively frequent basis, and generally uses the latest versions of the various open- source projects that it includes.

SUSE Linux Enterprise is the commercial version of SUSE Linux, which enables it to offer support more effectively for enterprise and production deployments. It is certified for a wide variety of

enterprise applications and offers a number of special enterprise features including, High Availability and Point of Sale extensions. SUSE historically uses a heavily-tested subset of packages from openSUSE Linux as the basis for SUSE Linux Enterprise.



CHECK YOUR PROGRESS

Q 1: Linux is free and open-source operating system. (True/False)

.....

Q 2: Which type of software product has the legal right to use?

.....

Q 3: Shell acts as an interface between the .....and

.....

Q 4: The Linux system is a single-user or multi-user operating system?

.....

Q 5: How many layers are there in Linux OS architecture and what are these?

.....



2.7 LET US SUM UP

- An operating system is an important part of a computer system. A computer system can be viewed as being built from three general components: the hardware, the operating system and the applications.
- The operating system is the component that on one side manages and controls the hardware and on the other, manages the applications.
- Linux operating system is one of the popular versions of the UNIX operating system, which is designed to offer a free or low-cost operating system for personal computer users.
- Linux is a family of free and open-source software operating systems based on the Linux kernel; an operating system kernel was first released on September 17, 1991 by Linus Torvalds. Linux is typically packaged

in a Linux distribution.

- Linux Operating System has primarily three components: kernel, system library and system utilities.
- The architecture of Linux Operating System has the following layers- hardware, kernel, shell, and utilities.
- Some basic features of Linux are: portability, multi-user, multiprogramming, hierarchical file system, shell and security.
- Linux also provides many additional features like live CD/USB, graphical user interface, support most national or customized keyboards and application support.
- A typical Linux distribution comprises a Linux kernel, GNU tools and libraries, additional software, documentation, a window system (the most common being the X Window System), a window manager, and a desktop environment.
- Some popular Linux distributions are - Red Hat Enterprise Linux, Ubuntu, Fedora, SUSE and Debian.



2.8 FURTHER READINGS

1) Das, S. (2012). Your UNIX/Linux: The Ultimate Guide. (Third Edition). (Science/ Engineering/Math). McGraw-Hill.
2) Negus, C. (2012). Linux Bible. (Vol. 772). John Wiley & Sons.



2.9 ANSWERS TO CHECK YOUR PROGRESS

Ans to Q No 1: True

Ans to Q No 2: Proprietary software

Ans to Q No 3: Kernel, user

Ans to Q No 4: Multi user operating system

Ans to Q No 5: Four layers. Kernel, shell, Hardware and Utilities



2.10 MODEL QUESTIONS

Q 1: Define Operating system. What is Linux? What are the core components of Linux?

Q 2: Distinguish between Proprietary, free and open-source software.

Q 3: Discuss about different layers of Linux architecture.

Q 4: What are the basic features of the Linux Operating System?

Q 5: What is a Linux distribution? Name some of the popular Linux distributions.

Q 6: What is Ubuntu? What are the features of Ubuntu?

Q 7: What is Fedora? What are the features of Fedora?

Q 8: Discuss about RHEL system with various features.

*** ***** ***

UNIT 3:INSTALLATION OF LINUX OPERATING SYSTEM

UNIT STRUCTURE

3.1 Learning Objectives

3.2 Introduction

3.3 Installation Requirements

3.4 Partitioning the Hard Drive in Linux

- 3.4.1 Partitions
- 3.4.2 Extended and Logical Partitions
- 3.4.3 Partitioning the Hard Drive in Linux

3.5 Installing the Linux System (Red Hat Enterprise Linux 6.0)

3.6 Installing and Configuring Software in Linux

- 3.6.1 Debian, Ubuntu: apt
- 3.6.2 Fedora, Red Hat: yum

3.7 Installing the Ubuntu 20.04

3.8 Let Us Sum Up

3.9 Further Readings

3.10 Answer to Check Your Progress

3.11 Model Questions

3.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- learn the hardware and software requirements for installing Linux operating system
- know about hard disk partitions
- describe logical partitioning
- install Linux system on a computer
- install software in Linux Operating System

3.2 INTRODUCTION

In the previous unit, we have learnt about the Linux Operating System. In this unit, step-by-step installation of Linux OS will be discussed.

The Linux OS is a freely distributable; cross-platform operating system based on Unix and can be installed on PCs, laptops, mobile and tablet devices, video game consoles, servers, supercomputers and more. Certain hardware requirements are to be met before installing Linux Operating System on a computer. Once this is met, the hard drive requires to be partitioned for installing the operating system. The installation steps for Ubuntu 20.04 LTS will also be discussed.

3.3 INSTALLATION REQUIREMENTS FOR RED HAT ENTERPRISE LINUX

Both of the Red Hat Enterprise Linux versions for 32-bit and 64-bit platforms have the same system requirements as listed below (Table 3.1).

Criteria	**Requirements**
Operating System	Red Hat Enterprise Linux 4 or 5 with the latest patches and upgrades
CPU Type	Pentium 4 or higher; 2 GHz or higher
Memory/RAM	1-2 GB minimum
Hard Disk	4 to 5 GB minimum

3.4 PARTITIONING THE HARD DRIVE IN LINUX

3.4.1 Partitions

A hard disk can be divided into several segments called partitions. Each partition operates as if it were a separate hard disk. The idea behind this is that if we want to install two operating systems in one hard disk, we can divide the disk into two partitions. Each operating system uses its own partition area without interfering the other ones. In this way, the two operating systems can co-exist on the same hard disk. The information about how a hard disk has been partitioned is stored in its first sector (that is, the first sector of the first track on the first disk surface). The first sector is called the master boot record (MBR) of the disk. BIOS reads in MBR and

starts when the machine is first booted. The MBR contains a small program that reads the partition table, and checks which partition is active or bootable, and then reads the first boot sector of that partition (the MBR is also a boot sector, but it has a special status and therefore a special name). This boot sector contains another small program that reads the first part of the operating system stored on that partition, and then it starts it.

There are three types of partitions available namely primary partition, extended partition and logical partition.

Primary partition is a partition which is used to save and boot an operating system. Usually, the operating system is installed in a primary partition which can be set "active" so that the computer can locate the operating system that needs starting. If there are more than one primary partition on the hard disk, only one of them can be visible and active at a time. This active partition is the partition from which the operating system is booted when computer starts. There are at most four primary partitions or three primary partitions and an extended partition on a hard disk.

The partitioning scheme is not built into the hardware, or even into the BIOS. It is only a convention that many operating systems use. Some operating systems support partitions, but they occupy one partition on the hard disk, and use their internal partitioning method within that partition. The latter type like Linux exists with other operating systems and does not require any special measures. An operating system that doesn't support partitions cannot co-exist on the same disk with any other operating system.

3.4.2 Extended and Logical Partitions

Extended partition: Extended partition is a kind of special primary partition which can be subdivided into logical partitions. It can be seen as a container for logical partitions. A hard disk can contain only one extended partition and it is not formatted or assigned a

drive letter, for it is used to create logical partitions.

Logical partition: Logical partitions are partitions created in extended partition. Logical partitions are similar to primary partitions. However, the number of logical partitions that can exist on a hard disk is unlimited.

An extended partition does not directly hold data and logical partitions are mainly used for storing data. Once created, logical partitions must be formatted, and each may be assigned a drive letter. It is a good idea to have several partitions for one operating system. For example, swap space is usually best put in its own partition for Linux instead of in the main Linux partition for reasons of speed. To overcome this design problem, extended partitions were invented. This trick allows partitioning a primary partition into sub partitions. The primary partition thus subdivided is the extended partition; the sub-partitions are logical partitions. They behave like primary partitions but are created differently.

A sample partitioned structure of a hard disk might look like that in Figure 3.1. The disk is divided into three primary partitions, the second of which is divided into two logical partitions. Part of the disk is not partitioned at all. The disk as a whole and each primary partition has a boot sector.

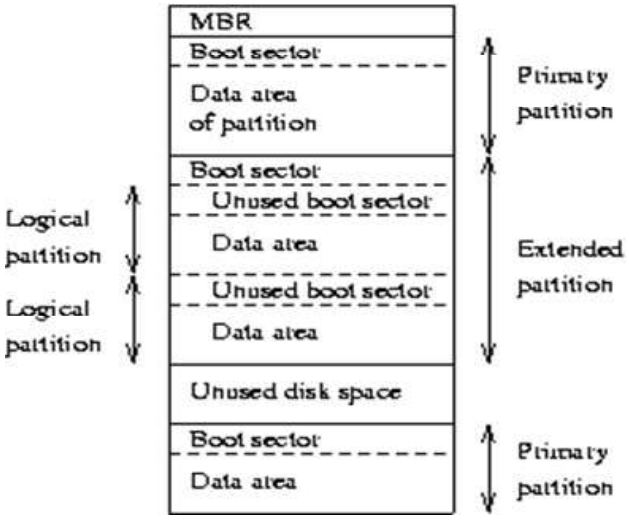


Fig. 3.1: A sample hard disk partitioning.

3.4.3 Partitioning the Hard Drive in Linux

Partitioning disks (Hard Drive) is generally the first step before installing Linux operating system. Before we can create any files, a filesystem must exist. Similarly, we can't make a file system unless there is a partition on the disk.

Partitions are used to divide raw storage space into chunks (slices). This provides the means for organization and hosting of filesystems. Besides that, partitions also help to isolate the filesystem faults.

For a proper Linux system installation in a single-user desktop system, three partitions are recommended: swap, root, and home.

swap: This is a partition that will be used as Linux swap space. This space on the hard drive can be used as virtual memory. Virtual memory allows the computer to run large programs and perform complex tasks even if it does not have enough physical RAM to do the job. There are no fixed guidelines for the necessary amount of swap space that is required. To keep it simple, create one swap partition that is 50% to 100% the size of the physical RAM in the system.

root (/): The root file system is represented by a forward slash (/). It is the top of the directory tree and contains Linux and everything that we install with Linux. We must create a partition for the root directory. The size of the root partition will vary depending on what to install or plan to install. It is to check distribution's documentation and reserve enough space for a maximum installation, plus at least 100MB more for temporary space and installation of new software. If we plan to download and try out lots of software, more space is required. If hard drive is in small size, we can trim back on our installed packages to save space. In general, root partition is kept in between 2GB and 8GB.

/home: The third and final partition is created to hold /home directory. This is the place where all the user-specific files are stored. On a multi-user system, each user will have their own directory under /home. Strictly speaking, it is not necessary to create a separate partition for /home. If we do not, it will reside on the root partition like everything else.



CHECK YOUR PROGRESS

Q 1: What is the minimum RAM and Hard disk space required for Red Hat Enterprise Linux installation?

.....

Q 2: What is master boot record (MBR)?

.....

Q 3: Mention the type of partitions recommended for a proper Linux installation in a single-user desktop system?

.....

Q 4: What is the top of the directory tree in Linux?

.....

3.5 INSTALLING THE LINUX SYSTEM (Red Hat Enterprise Linux 6.0)

The steps which are to be followed to install Red Hat Enterprise Linux are as below:

Step 1: Boot Computer Using Red Hat 6 Installation CD/DVD.

Then Select Install or upgrade an existing system option on Grub Menu (Fig. 3.2)



Fig. 3.2

2. Select a language from the language option.



Fig. 3.3

Generally, English option is used in India (Fig. 3.3).

3. Choose a keyboard type (Fig. 3.4).



Fig. 3.4

Note: To change the keyboard layout type after the completion the installation, use the Keyboard Configuration Tool. Type the system-config-keyboard command in a shell prompt to launch the Keyboard Configuration Tool. If the user is not admin, it prompts him for the root password to continue.

4. Choose an installation media from the screen (Fig. 3.5).



Fig. 3.5

5. Skip DVD media test (or select media test, if you want to test installation media before installation) (Fig. 3.6).



Fig. 3.6

6. Red Hat 6 graphical installer starts, select next (Fig. 3.7).



Fig. 3.7

7. Accept Pre-Release Installation (Fig. 3.8).

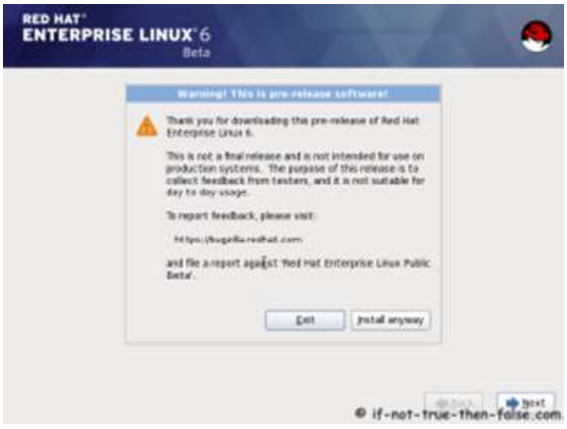


Fig. 3.8

8. Select storage devices.

We can install Red Hat Enterprise Linux on a large variety of storage devices - either basic or specialized storage devices. Basic Storage Devices are hard drives or solid-state drives connected directly to the local system wherein Specialized Storage Devices include Storage area networks (SANs), Direct access storage devices (DASDs), Firmware RAID devices and Multipath devices (Fig. 3.9)



Fig. 3.9

9. Insert computer name (Fig. 3.10).



Fig. 3.10

10. Select time zone (Fig. 3.11).

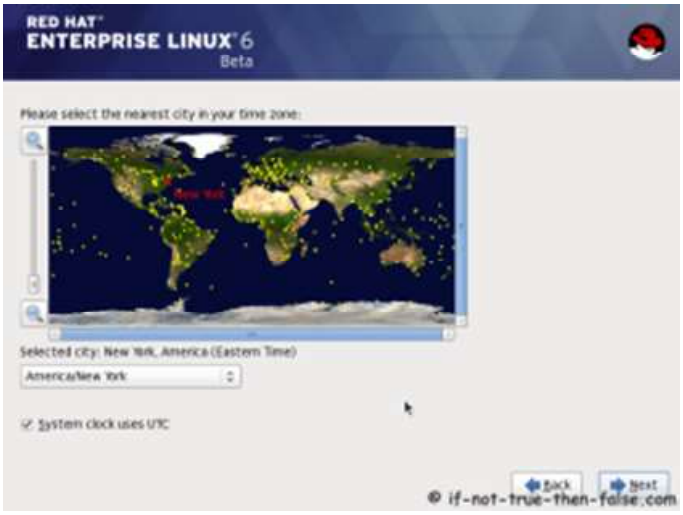


Fig. 3.11

11. Enter a password for root user (Fig. 3.12).



Fig. 3.12

Setting up a root account and password is one of the most important steps during the installation process. The root account is used to install packages, upgrade RPMs, and perform most system maintenance. Logging in as root gives us complete control over the system.

12. Select type of installation (Fig. 3.13).

Read every options info carefully. And select encrypting if needed and option to review and modify partition layout.



Fig. 3.13

Partitioning allows us to divide the hard drive into isolated sections, where each section behaves as if it is a separate hard drive. Partitioning is particularly useful if multiple operating systems are to be installed. In the above screen we can choose to create the default partition layout using one of four different ways or choose to partition storage devices manually to create a custom layout. The first four options allow us to perform an automated installation without having to partition our storage devices. If we do not feel comfortable with partitioning our system, we can choose one of these options and let the installation program partition the storage devices for us. Depending on the option we select, we can still control what data (if any) is removed from the system.

Type of installation can be chosen from:

Use All Space: Select this option to remove all partitions on our hard drives (this includes partitions created by other operating systems such as Windows VFAT or NTFS partitions).

Replace Existing Linux System(s): Select this option to remove only partitions created by a previous Linux installation. This does not remove other partitions we may have on our hard drives (such

as VFAT or FAT32 partitions).

Shrink Current System: Select this option to resize our current data and partitions manually and install a default Red Hat Enterprise Linux layout in the space that is freed.

Use Free Space: Select this option to retain the current data and partitions and install Red Hat Enterprise Linux in the unused space available on the storage drives. Ensure that there is sufficient space available on the storage drives before we select this option.

Create Custom Layout: Select this option to partition storage devices manually and create customized layouts.

13. Review partition layout (Fig. 3.14).

Modify if needed. Default setup with ext4 and LVM looks good for desktop machine.



Fig. 3.14

14. Accept write changes to disc (Fig. 3.15).



Fig. 3.15

15. Writing changes (creating partitions) to disc (Fig. 3.16).



Fig. 3.16

16. Configure boot loader options (Fig. 3.17).

Select device to install bootloader and check/create boot loader operating system list.



Fig. 3.17

17. Select software to install and enable repositories.

In this case, we have to select Software Development Workstation and enable Red Hat Enterprise Linux 6.0 Beta Repository and select Customize now (Fig. 3.18).

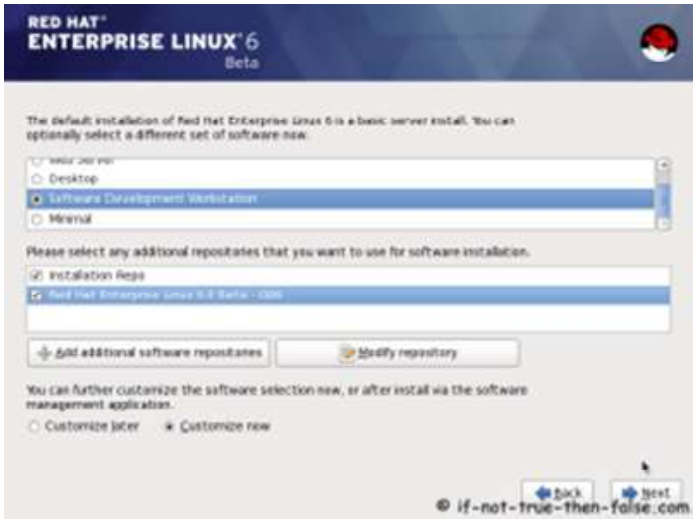


Fig. 3.18

18. Customize package selection.

By default, the Red Hat Enterprise Linux installation process loads a selection of software that is suitable for a system deployed as a basic server. Note that this installation does not include a graphical environment. To include a selection of software suitable for other roles, click the radio button that corresponds to one of the following options:

Basic Server: This option provides a basic installation of Red Hat Enterprise Linux for use on a server.

Database Server: This option provides the MySQL and PostgreSQL databases.

Web Server: This option provides the Apache web server.

Enterprise Identity Server Base: This option provides OpenLDAP and Enterprise Identity Management (IPA) to create an identity and authentication server.

Virtual Host: This option provides the KVM and Virtual Machine Manager tools to create a host for virtual machines.

Desktop: This option provides the OpenOffice.org productivity suite, graphical tools such as the GIMP, and multimedia applications.

Software Development Workstation: This option provides the necessary tools to compile software on your Red Hat Enterprise Linux system.

Minimal: This option provides only the packages essential to run Red Hat Enterprise Linux. A minimal installation provides the basis for a single-purpose server or desktop appliance and maximizes performance and security on such an installation.

Select PHP and Web Server to installation (Fig. 3.19).

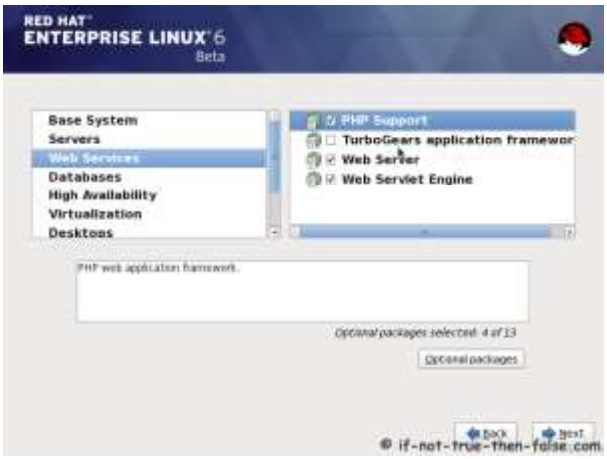


Fig. 3.19

Select MySQL and PostgreSQL Databases (Fig. 3.20).

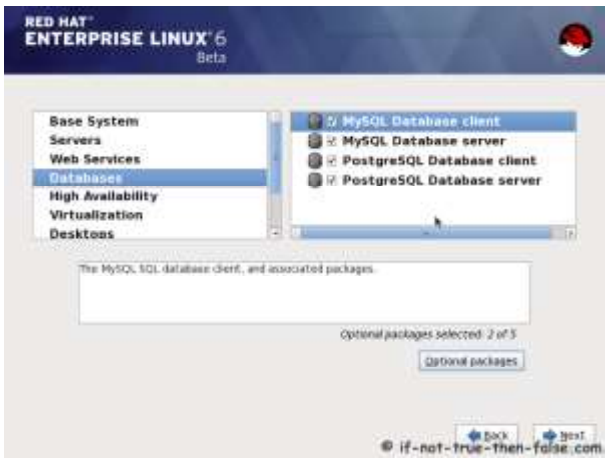


Fig. 3.20

Select set of Development tools like Eclipse IDE (Fig. 3.21).



Fig. 3.21

19. Checking dependencies for installation (Fig. 3.22).

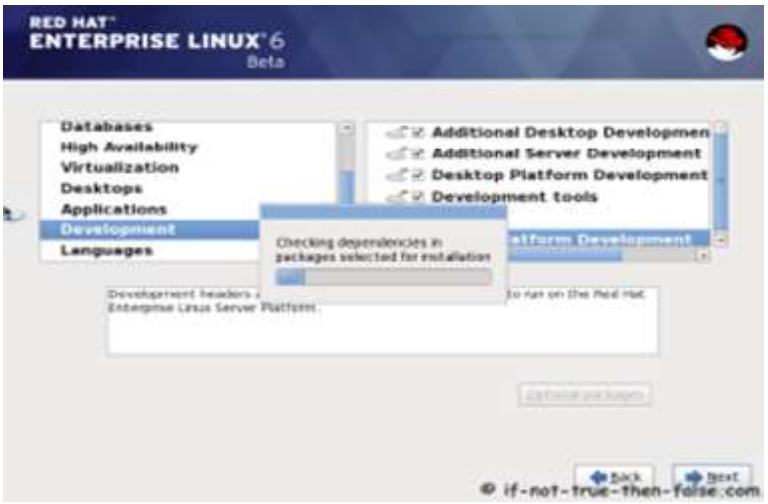


Fig. 3.22

20. Starting installation process (Fig. 3.23).



Fig. 3.23

21. Installing packages (Fig. 3.24 & 3.25).



Fig. 3.24



Fig. 3.25

22. Installation is complete (Fig. 3.26).

Click reboot computer and remove installation media.



Fig. 3.26

3.6 INSTALLING AND CONFIGURING SOFTWARE IN LINUX

The standard Linux package format is called RPM. RPM stands for Red Hat Packet Manager. It is originally developed by Red Hat and widely used in the Linux community Distributions which include Fedora, Mandriva, Red Hat (naturally), and SUSE. An RPM package file normally will be named something like program-version-other.rpm. Another popular package format is DEB, the Debian software package. Debian packages and the Advanced Packaging Tool (APT) have introduced several advanced features such as automatic dependency resolution and signed packages. Debian packages are used by Debian GNU/Linux, and distributions based on it, including Ubuntu, Knoppix, and Mepis. A Debian package file is normally named something like program-version-other.deb. It is to be noted that to install software packages, user needs to become SuperUser.

3.6.1 Debian, Ubuntu: apt

There is an array of tools for working with DEB packages, but the one the commonly used is apt-get, arguably the easiest of Linux package management tools. apt-get is easy to use because it not only keeps track of what packages are installed, but also what other packages are available. We can install software package by the following way.

apt-get install ${packagename}

To remove any software package, we can use the following.

apt-get remove ${packagename}

Although the repositories that contain installable packages might live on the Internet or on a disc somewhere, APT keeps a local database on your hard drive with a list of all available packages and where to find them. This database needs to be explicitly updated which is done in the following way.

apt-get update

Moreover, upgradation of all the packages that have patches or security updates is also required. The following command will do this all at once.

apt-get update

apt-get upgrade

3.6.2 Fedora, Red Hat: yum

Like what apt-get does for Debian packages, yum does for RPM packages. yum can download and install software packages from a configured repository by the following way.

yum install ${packagename}

Software can be removed by using the following command.
yum remove ${packagename}

yum does not keep a local copy of your package database by default, so normally there is no need to update it. To install all available security patches and bug fixes, use the following command:

yum update

We can also explicitly update a single package in the following way:

yum update ${packagename}



CHECK YOUR PROGRESS

Q 5: Linux cannot be installed on Direct access storage devices. (State TRUE or FALSE)

Q 6: Which account gives you total control of the Linux system?

.....

Q 7: Which package group provides MySQL and PostgreSQL databases?

.....

Q 8: Which package group provides OpenOffice.org productivity suite?

.....

Q 9: What needs to be installed to boot the system without boot media?

.....

Q 10: What is the command to install software in Debian?

.....

3.7 INSTALLING THE UBUNTU 20.04

Ubuntu 20.04 was released on April 23, 2020 and it is known as a Long-Term Support (LTS) release, meaning that it can receive support and updates until April 2025. Ubuntu 20.4 offers some new features which include the following.

- **Linux Kernel 5.4 -** This version of the Linux kernel adds support for a wider range of processors. It also includes improvements for boot speed, power-saving, and USB-C, plus several security features.
- **Gnome 3.36 -** Gnome has been updated and improved. System animations should be smoother and have a lower impact on CPU usage.
- **ZFS 0.8.3 file system -** Improved performance and native encryption.
- **Dark theme**
- **Redesigned login screen**
- **Newer versions of popular programming languages -** Supports the languages like Python 3.8, OpenJDK 11, PHP 7.4, Glibc 2.31, GCC 9.3, Ruby 2.7.0, Perl 5.30, Golang 1.13.

To install Ubuntu 20.04 in a computer system, the following requirements are essential:

- 2 GHz dual-core processor
- 4 GB memory
- 25 GB available disk space for storage
- DVD drive or USB port
- At least a 4GB USB drive

Steps involved to install Ubuntu 20.04 are as follows.

Step 1: Download the Installation Media

1. In a web browser, visit the Ubuntu download page and pick the Ubuntu version suitable for your machine. The most popular versions

include:

- Ubuntu Desktop
- Ubuntu Server
- Ubuntu Derivatives

2. Once you find the version you need, click the green Download button (Fig. 3.27). It will take back to a thank-you page, and download will start. The downloaded file will be an ".iso" file. You can use it to create a bootable USB drive.

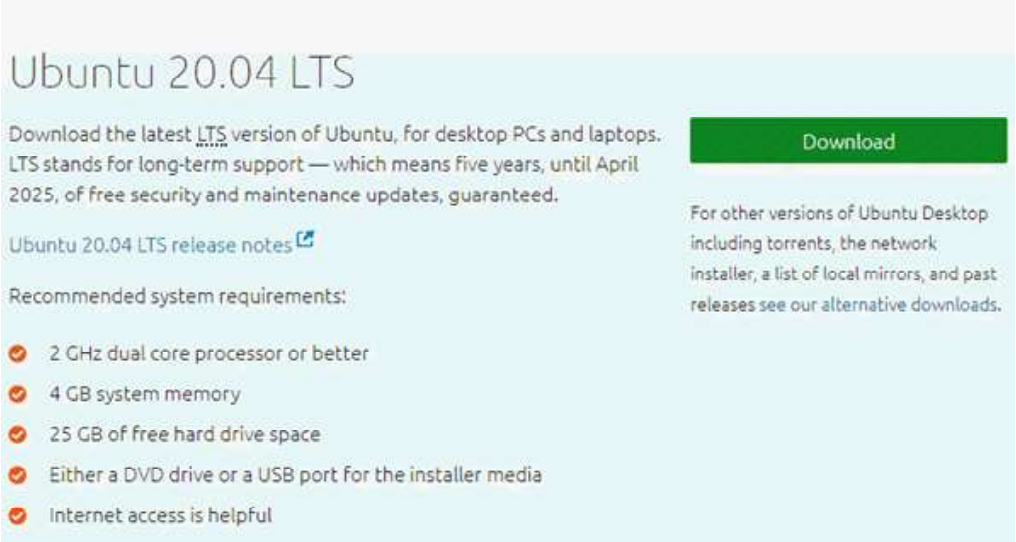


Fig. 3.27

3. Save the file to a location of your choice.

Step 2: Create Bootable USB

You will need a USB drive with 4GB or more. This process will delete all data on the USB drive. Make sure to backup any existing data on the USB drive.

Option 1: Create a Bootable USB Drive on Ubuntu

Use the Create Start-up disk tool:

- Open a search dialog, and type create startup.
- If it's not installed, the Software Center will offer the option to install it - choose the option for USB drive, then open the utility.
- In the top pane, click Other, then browse and select the Ubuntu 20.04 .iso file you downloaded.
- In the bottom pane, select your USB drive.

- Click Make startup disk.

Option 2: Create Bootable USB Drive on Windows

You'll need to install a third-party utility called Rufus to create a USB bootable drive.

1. Download the Rufus utility. Scroll down to the download section and click the link to download the latest version of Rufus (Fig. 3.28).



Fig. 3.28

2. Run the file once downloaded.

3. A pop-up dialog opens. You will be prompted whether you want to check for online updates. Select No. See fig. 3.29.

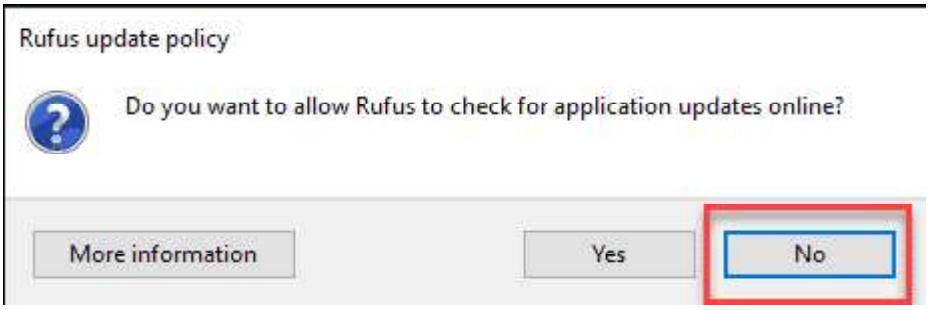


Fig. 3.29

4. The Rufus utility launches. Plug in the USB drive - you should see the drive pop up in the device field (Fig. 3.30).

- Set the USB as the device you wish to write to.
- In the Boot Selection drop-down, click Disk or ISO Image.

- Click the Select button to the right.
- Browse and select the .iso Ubuntu file you downloaded earlier.

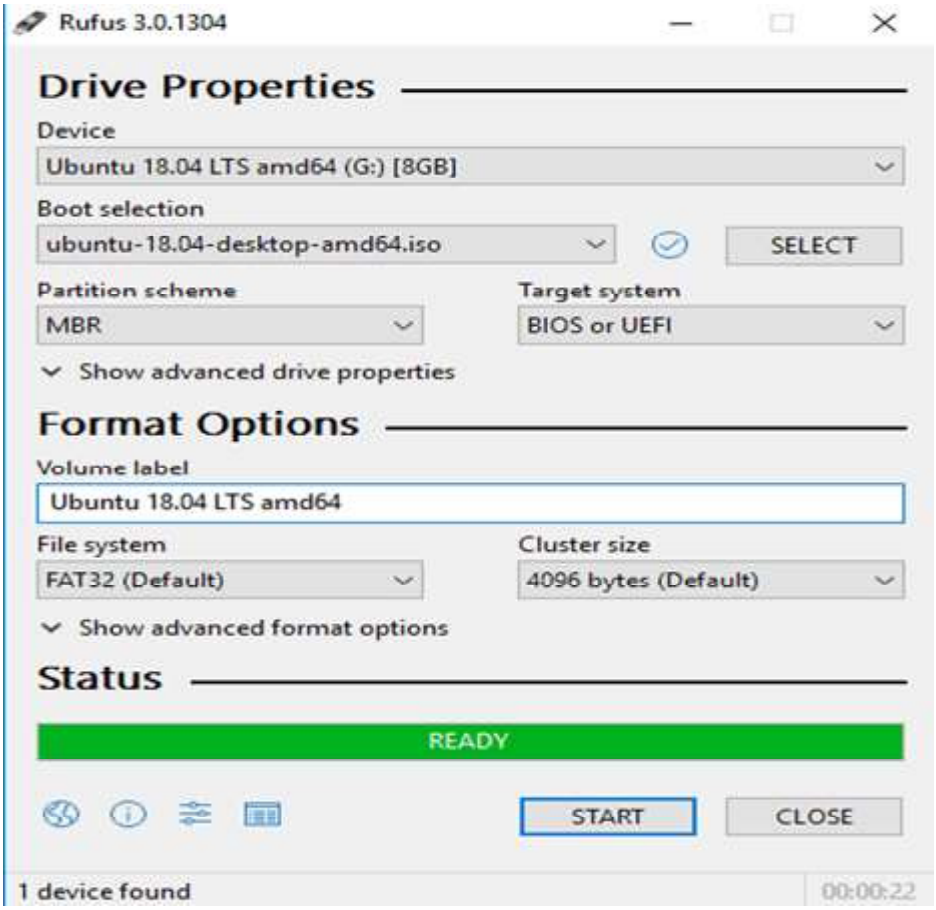


Fig. 3.30

5. Click Start.

Step 3: Boot up Ubuntu from USB

1. Turn off your system. Make sure you remove all other USB devices, such as printers, memory cards, etc.

2. Insert the Ubuntu USB drive into the system and turn on your machine.

There are two possible scenarios:

- The computer boots the USB drive automatically.
- You need to manually configure USB booting in the Boot Menu or BIOS/ UEFI.

3. To manually configure the boot order, tap the boot menu key about once or twice per second as soon as the computer powers on.

The boot menu key may be different depending on your computer manufacturer. Below is a list of common boot keys associated to a brand (Table 3.2):

Asus	F8 or Esc
Acer	F12, F9 or Esc
Compaq	F9 or Esc
Dell	F12
Fujitsu	F12
HP	F9 or Esc
Lenovo	F8, F10 or F12
Samsung	F2, F12 or Esc
Toshiba	F12

4. Once you see your boot menu, use the arrows to pick the Ubuntu media to boot from. For a DVD, the entry will usually have DVD or Optical in the name. USB is usually labelled USB.

Your system should start loading the Ubuntu live disc menu.

Step 4: Run Ubuntu

You can test Ubuntu 20.04 before you commit to installing it. The .iso includes a live mode that only runs in memory.

Launch this mode by clicking Try Ubuntu. See fig. 3.31.

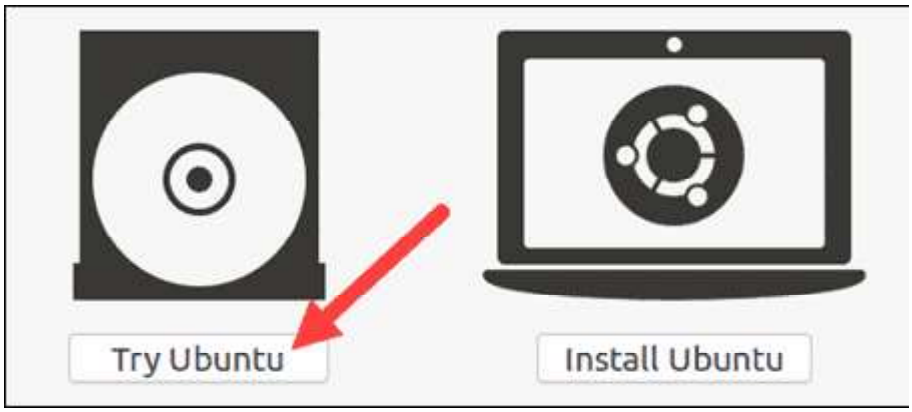


Fig. 3.31

Step 5: Install Ubuntu 20.04 LTS Desktop

To begin the installation, click Install Ubuntu (Fig. 3.32).

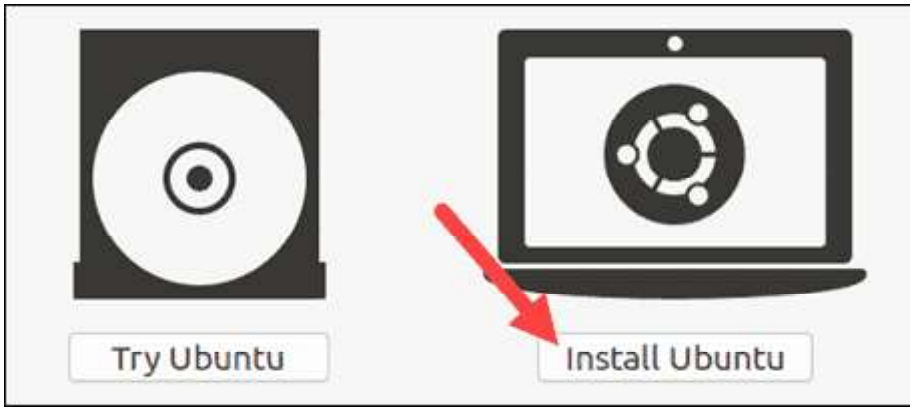


Fig. 3.32

Choose Keyboard Layout

By default, the system will select English (US).

If you have a non-standard keyboard, you can select it in the list. Alternately, click Detect Keyboard Layout and the system will automatically choose your keyboard. If you need to test your keyboard, use the labelled field.

When you're ready, click Continue. See fig. 3.33.

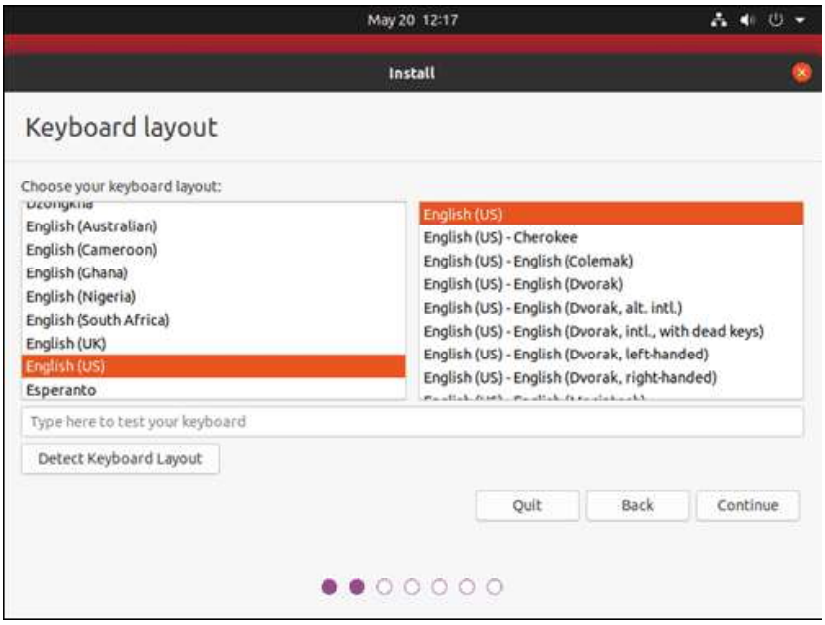


Fig. 3.33

Choose Starting Applications

- **Normal Installation -** This is the full Ubuntu Desktop experience, with office software, games, and media players.

- **Minimal Installation -** Choose this to save disk space, especially if you won't be using media players or productivity software.
  You will also be asked to confirm other options:
- **Download updates while installing Ubuntu -** This does the work of downloading large package files during the installation. Once the installation finishes, the packages will be ready to apply as updates.
- **Install third-party software for graphics and Wi-Fi hardware and additional media formats -** Some hardware, like graphics cards and wi-fi cards, do not have open-source driver support. Also, some media formats, such as ".wmv", do not fall under the GPL license. If you need support for these, you'll need to agree to additional terms of use.

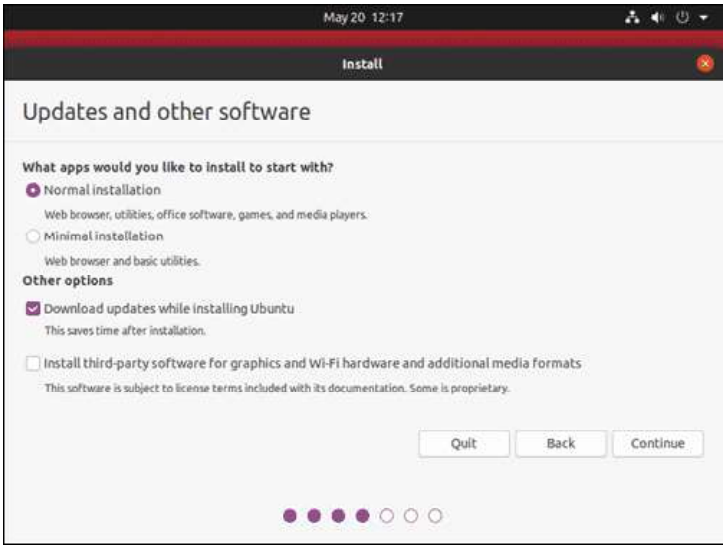


Fig. 3.34

Disk Partitioning

Next, an Installation Type dialog will appear. You can wipe the hard drive clean prior to installing Ubuntu by clicking Erase disk and install Ubuntu. If you go way, skip ahead to the next step.

Advanced users may want to edit Advanced Features. Use this to specify your own disk partitions or set other advanced options:

- **Use LVM with the new Ubuntu installation:** LVM stands for Logical Volume Management. This is a tool for dynamically managing different virtual drives on your system. It's much like an enhanced version of the gparted tool.

- **Encrypt the new Ubuntu installation for security:** This will encrypt the drive's contents. You'll choose a security key, which will be required to decrypt and use the drive.
- **Experimental: Erase disk and use ZFS:** ZFS refers to Zettabyte File System, but it has grown into a hybrid file system and volume manager. Since it's still being tested, avoid this setting on mission-critical production systems.

If you'd rather create your own hard drive partitions, click Something Else.

The next screen will allow you to create your own partition table and logical drives. This lets you divide a physical hard drive into different partitions. The operating system sees partitions as individual drives.

Note: Some users create their /home directory on a separate partition. If the operating system needs to be reinstalled, the partition with the /home directory is unaffected.

Click Continue to apply your changes to the drive partitions.

You'll be asked to Write changes to disks? None of the options you've selected are permanent until you click Continue on this screen. Click Continue to proceed.

Select Time Zone

Once the system formats the disk partitions, the installer will ask Where are you?

Type the nearest large city into the box, and the system will set your local time zone.

Click Continue.

Create User Account

Next, you'll need to configure a user account. Fill in the following fields:

- Name: Your actual name.
- Computer name: This is the hostname or network name.
- Username: The user account name you want to use.
- Password: Enter and confirm a strong password - the installer will automatically evaluate your password strength.

- **Log in automatically:** This is not recommended for publicly accessible servers.
- **Require my password to log in:** This is recommended for publicly accessible servers.

Click Continue to install Ubuntu.

Once the installer finishes, remove the Ubuntu installation media. You'll be prompted to Restart Now (Fig. 3.35).

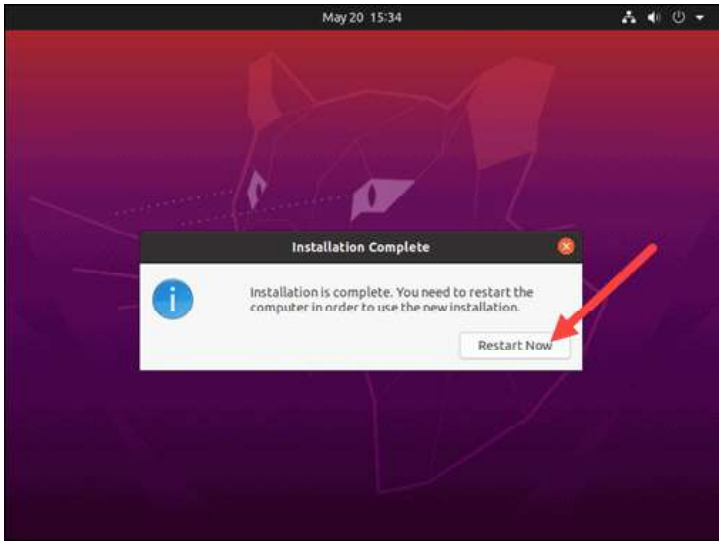


Fig. 3.35

The system should boot into your fresh install of Ubuntu 20.04.



3.8 LET US SUM UP

- Both Red Hat Enterprise Linux versions on both 32-bit and 64-bit platforms need the same system requirements for installation: Pentium 4 or higher CPU of 2Ghz or higher speed, 1 GB minimum RAM and 4 GB minimum hard disk.
- A hard disk can be divided into several partitions. Each partition functions as if it were a separate hard disk.
- The information about how a hard disk is partitioned is stored in its first sector which is named as the master boot record (MBR) of the disk; this is the sector that the BIOS reads in and starts when the machine is first booted.

- For a proper Linux installation in a single-user desktop system, three partitions are recommended: swap, root, and home. The swap partition is the space on the hard drive that can be used as virtual memory. Virtual memory allows the computer to run large programs and perform complex tasks even if it does not have enough physical RAM to do the job.
- The root (/) partition is the top of the directory tree and contains Linux and everything that are installed with Linux.
- The home directory is the place where all the user-specific files are stored. On a multi-user system, each user will have their own directory under/home.
- We can install Red Hat Enterprise Linux on a large variety of storage devices- either basic or specialized storage devices. Basic Storage Devices include hard drives or solid-state drives connected directly to the local system. Specialized Storage Devices include Storage area networks (SANs), Direct access storage devices (DASDs), Firmware RAID devices and Multipath devices.
- The root account is used to install packages, upgrade RPMs, and perform most system maintenance. Logging in as root gives us complete control over your system.
- By default, the Red Hat Enterprise Linux installation process loads a selection of software that is suitable for a system deployed as a basic server.
- To boot the system without boot media, we usually need to install a boot loader. A boot loader is the first software program that runs when a computer starts. It is responsible for loading and transferring control to the operating system kernel software. The kernel, in turn, initializes the rest of the operating system.
- To install packages on Debian and Ubuntu Systems the command apt-get install ${packagename} is used. To remove software, we must use apt-get remove ${packagename}.
- To install packages on Fedora and Red Hat Linux systems, the command yum install ${packagename} is used.



3.9 FURTHER READINGS

1) Das, S. (2012). Your UNIX/Linux: The Ultimate Guide. (Third Edition). (Science/Engineering/Math). McGraw-Hill.
2) Negus, C. (2012). Linux Bible. (Vol. 772). John Wiley & Sons.



3.10 ANSWERS TO CHECK YOUR PROGRESS

Ans to Q No 1: minimum 1 GB RAM and minimum 4 GB hard disk.

Ans to Q No 2: MBR is the sector that the BIOS reads in and starts when the machine is first booted.

Ans to Q No 3: swap, root, and home.

Ans to Q No 4: root

Ans to Q No 5: FALSE

Ans to Q No 6: root

Ans to Q No 7: Database Server

Ans to Q No 8: Desktop

Ans to Q No 9: boot loader

Ans to Q No 10: apt-get install ${packagename}



3.11 MODEL QUESTIONS

Q 1: What are the system requirements for installing Red Hat Enterprise Linux?

Q 2: What are partitions? What are extended and logical partitions?

Q 3: What are the different recommended partitions for a proper Linux installation in a single-user desktop computer?

Q 4: What are the different kinds of storage devices where Linux can be installed?

Q 5: What are the different options available in the Disk Partition Setup of Linux installation?

Q 6: What are the different package groups available during Linux installation?

Q 7: What are the system requirements for installing Ubuntu 20.04?

Q 8: Mention the steps of installing Ubuntu 20.04?

*** ***** ***

UNIT 4: LINUX KERNEL

UNIT STRUCTURE

4.1 Learning Objectives

4.2 Introduction

4.3 Linux Kernel Concepts

- 4.3.1 Major Functions of Linux Kernel
- 4.3.2 Architecture of Linux Kernel
- 4.3.3 Features of Linux Kernel

4.4 Device Drivers

- 4.4.1 Block and Character Devices
- 4.4.2 Major and Minor Numbers

4.5 System Startup

4.6 System Shutdown

4.7 Redirection

- 4.7.1 Standard Input
- 4.7.2 Standard Output
- 4.7.3 Standard Error

4.8 Piping

4.9 Let Us Sum Up

4.10 Further Readings

4.11 Answer to Check Your Progress

4.12 Model Questions

4.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- learn about Linux kernel
- describe the architecture of the Linux kernel
- describe the role of the kernel in the Linux system
- define device driver and its types
- explain how a device is represented in Linux system
- know about redirection and piping

4.2 INTRODUCTION

From the previous units, we have already learned the basic concepts along with its architecture and the basic features of the Linux operating system. We are also accustomed with the basic ideas about the various Linux Distributions. In the last unit, the concept of the linux installation process & requirements for the installation, how to partition the hard drive in linux, how to install the linux etc. were described. Besides these, we have also learned to install and configure different software or application in linux in the previous units.

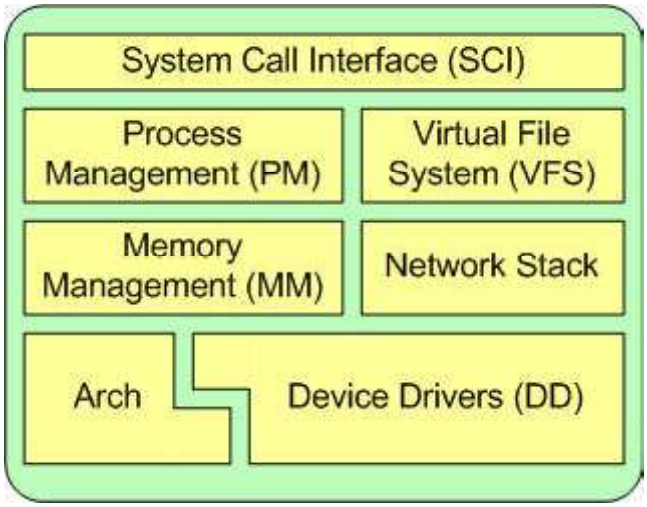
In this unit, we are going to discuss about the Linux Kernel, the role of the kernel in the linux system and the architecture of the Linux kernel. By going through this unit, we will be able to know about the concepts of device drivers in linux system in addition to booting and shutting down process of the system. The important features such as Redirection & Piping and how they work will also be learnt in this unit besides knowing about the standard input, standard output and the standard error in linux system.

4.3 LINUX KERNEL CONCEPTS

Kernel is a special code written in C language which directly interacts with hardware. It is at the core of the Linux system. Kernel runs simultaneously a number of processes. It acts as a manager for various resources like CPU, memory, process etc. Linux kernel is efficient in terms of both memory and CPU utilization. Kernels are of two types - monolithic kernel and micro kernel.

Linux kernel is a monolithic kernel because it accumulates all of the basic services. The linux kernel is layered into a number of distinct subsystems. The kernel is conceptually composed of six major subsystems: System Call Interface, the Process Management, the Memory Management, the Virtual File System, the Network Interface, and the Device Driver.

The major subsystems of the kernel are presented in the figure 4.1.



4.3.1 Major Functions of Linux Kernel

The major functions performed by kernel are described below.

- **Process Management:** The kernel handles the creation and destruction of the processes and their connection to the input and output systems. The inter process communication, i.e., the communication among different processes is also handled by the kernel. Process management can provide the information about what is happening with a process, file allocated to process, state of process (like running, waiting, stop), etc.
- **Memory Management:** The memory management is the most important function of kernel. Kernel assigns the address space to processes and applications, basically assigning virtual memory instead of physical memory, whereby later the actual address space in RAM is assigned and the conversion of physical address to virtual address is done by MMU (Memory Management Unit).
- **File System Management:** In Linux, almost everything is considered as a file. The Linux file system is a hierarchically data structured tree starting from root followed by other subdirectories which organizes the file system. A directory is a group of files. Two types of directories available in Linux system include: 1) root and 2) Sub directory. The root is the main

directory and sub directories can be created or renamed by the user under the root directory. The Linux file system supports a number of file types like ext2, ext3, device file, block file, networking file system etc. and each file system is stored in different disk partition.

- **System Call Interface:** The system call interface accomplishes function calls from user space into the kernel. It can be architecture dependent, even within the same processor family.
- **Device controlling:** This subsystem which maps to the different physical devices need to perform device control operations which are performed by code that is specific to the device being addressed. Those codes are called device drivers. For every peripheral connected to a system, there is a device driver embedded into the kernel.
- **Network Management:** This function is responsible for delivering data packets across program and network interfaces. The kernel handles the routing and address resolution issues in networking.

4.3.2 Architecture of Linux Kernel

The architecture of the Linux kernel is divided into two levels - User Space and Kernel Space. The architecture is depicted in the figure 4.2.

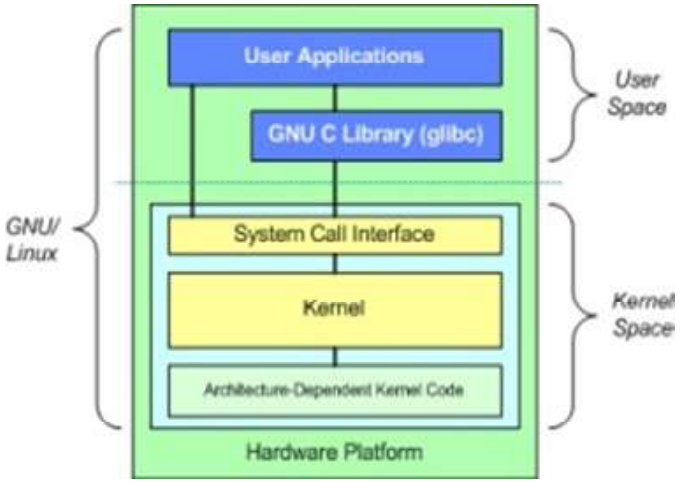


Fig. 4.2: Architecture of Linux Kernel

User Space: All the user programs and applications are executed in User Space. This Space cannot directly access the memory and hardware. It accesses those through kernel space. Hence, it needs to be switched user space application to kernel space and the switching mechanism is provided by the GNU C library.

Kernel Space: Kernel Space is for executing kernel programs. It can access full part of memory and directly interact with the hardware. The kernel space is further subdivided into three levels - System Call Interface, Kernel and Architecture-Dependent Code. System Call Interface sits in between the user space and kernel space and it executes the basic functions such as read and write. Kernel is architecture-independent code which is common to all hardware processors. The architecture-dependent code serves as the processor and platform specific code for the given architecture.

To summarise, the kernel space is the memory area that is reserved to the kernel while user space is the memory area reserved to a particular user process. The kernel space is accessed protected so that user applications cannot access it directly, while user space can be directly accessed from code running in kernel mode.

4.3.3 Features of Linux Kernel

As discussed in previous units, Linux is an open-source operating system. It supports a large number of networking protocols, including the typical TCP/IP. Linux kernel is a dynamic kernel that supports the addition and removal of software components on the fly by employing dynamically loadable kernel modules. And they can be inserted at booting time when they are required. Though much of Linux is independent of the architecture on which it runs, but there are some elements which are considered for normal operation and for efficiency reason. Low-level Scheduling of Processes, Inter-

process Communication, Process Synchronization and Context Switching are some other features of Linux kernel.

4.4 DEVICE DRIVERS

The software that handles or manages a hardware controller is known as a device driver. The Linux kernel device drivers are, essentially, a shared library of privileged, memory resident, low-level hardware handling routines. It is Linux's device driver that handles the peculiarities of the devices they are managing. The device drivers have a special role in the kernel of the Linux system. All the devices are represented by files. Nearly all system operations map to different physical devices and all device control operations are performed by some routines that are specific to a device. Those routines are referred to as device driver. The kernel calls the appropriate device driver, whenever a particular device is accessed and then the kernel passes some parameter for it to act properly. The device files are stored in /dev directory or in its sub-directories. The device files contain no data. The following Linux command lists device names connected to the system.

```
$ ls -l /dev
total 0
crw----- 1 root root 10, 55 Feb 3 18:11 acpi_thermal_rel
crw--- 1 root root 10, 235 Feb 3 18:11 autofs
drwxr-xr-x 2 root root 400 Feb 3 18:10 block
drwxr-xr-x 2 root root 80 Feb 3 18:10 bsg
crw--- 1 root root 10, 234 Feb 3 18:10 btrfs-control
drwxr-xr-x 3 root root 60 Feb 3 18:10 bus
lrwxrwxrwx 1 root root 3 Feb 3 18:11 cdrom -> sr0
lrwxrwxrwx 1 root root 3 Feb 3 18:11 cdrw -> sr0
drwxr-xr-x 2 root root 3900 Feb 3 18:11 char
crw--- 1 root root 5, 1 Feb 3 18:11 console
lrwxrwxrwx 1 root root 11 Feb 3 18:10 core -> /proc/kcore
drwxr-xr-x 2 root root 60 Feb 3 18:10 cpu
```

```
crw--- 1 root root 10, 59 Feb 3 18:11 cpu_dma_latency
crw--- 1 root root 10, 203 Feb 3 18:10 cuse
drwxr-xr-x 8 root root 160 Feb 3 18:10 disk
drwxr-xr-x 2 root root 80 Feb 3 18:10 dri
crw--- 1 root root 246, 0 Feb3 18:11 drm_dp_aux0
crw--- 1 root root 246, 1 Feb 3 18:11 drm_dp_aux1
crw--- 1 root root 246, 2 Feb 3 18:11 drm_dp_aux2
lrwxrwxrwx 1 root root 3 Feb 3 18:11 dvd -> sr0
lrwxrwxrwx 1 root root 3 Feb 3 18:11 dvdrw -> sr0
crw--- 1 root root 10, 61 Feb 3 18:11 ecryptfs
crw-rw--- 1 root video 29, 0 Feb 3 18:11 fb0
lrwxrwxrwx 1 root root 13 Feb 3 18:10 fd -> /proc/self/fd
crw-rw-rw- 1 root root 1, 7 Feb 3 18:11 full
```

4.4.1 Block and Character Devices

The Linux system classifies most of the hardware devices into two forms: block or character devices. These devices are indicated by the first character (either b or c) in the permission field. The character b indicates that it is a block device file and if it is c then it means that it is a character device file. A block device can store or transfer a block of data at a time, wherein a character device can display, process or store a character at a time.

The floppy drive, CD-ROM and the hard disk drive are the block special devices. All data are read from or written into these devices in blocks and use the buffer cache. The terminal, tape drive and printer are the character special devices. The read/write operations for character devices do not use the buffer cache and access the device directly.

4.4.2 Major and Minor Numbers

The fifth field in the file list representing the size for the other files, consists of a pair of two numbers, separated by a comma. These

numbers called the major and minor device numbers respectively. The major number represents the type of device and the minor number represents the special characteristics of the device. For example, tty0 and tty1 have the same major number (4) but different minor numbers (0 and 1) which is presented below.

crw-w-- 1 root tty 4, 0 Feb 3 18:11 tty0

crw-w-- 1 root tty 4, 1 Feb 3 18:11 tty1

4.5 SYSTEM START UP

To start-up the system, we have to power on the machine such that the power is supplied to the peripherals those are immediately required. When the kernel is loaded, these peripherals are also to be loaded. The start-up or booting process follows through a number of stages.

At first, the kernel is loaded into the memory. It is represented by the file /vmlinuz. BIOS (Basic Input/Output System) then performs some system integrity checks. It searches, loads, and executes the boot loader program from floppy, CD-ROM, or hard drive. In simple terms, BIOS loads and executes the MBR (Master Boot Record) which is located in the 1st sector of the bootable disk. The MBR then loads and executes the GRUB (Grand Unified Bootloader) that has the knowledge of the file system. The GRUB configuration file is /boot/grub/grub.conf. This file loads and executes kernel and then the kernel starts spawning further processes, among which the most important is init. The kernel executes init process by executing the program /sbin/init. Since, init is the first process to be executed by the kernel, its PID (process id) is 1. The init then in turn spawns further processes. The init controls a number of modes which is called the run level. The available run levels are-

- 0 - halt
- 1 - Single user mode
- 2 - Multiuser, without NFS
- 3 - Full multiuser mode
- 4 - unused

- 5 - X11
- 6 - reboot

If we look at the /etc/inittab file to decide the Linux run level, it may observe that init identifies the default init level from /etc/inittab and uses that to load all appropriate program. Depending on the default init level, the system will execute the programs from one of the following directories.

o Run level 0 - /etc/rc.d/rc0.d/

o Run level 1 - /etc/rc.d/rc1.d/

o Run level 2 - /etc/rc.d/rc2.d/

o Run level 3 - /etc/rc.d/rc3.d/

o Run level 4 - /etc/rc.d/rc4.d/

o Run level 5 - /etc/rc.d/rc5.d/

o Run level 6 - /etc/rc.d/rc6.d/

4.6 SYSTEM SHUTDOWN

The shutdown command is used to bring the system down in a secured way. The /sbin/shutdown command controls the shutdown sequence. This shutdown job is executed by signalling the init process, asking it to change the run level. The activities performed by shutdown are-

- Informs the users that the system is going down.
- Notifies all the running processes that the system is going to be shut down by sending the signals so that the processes can terminate normally.
- Unmounts all the secondary file system.

Different options used with shutdown command can be explained with following example.

Example 4.1: Shutdown examples

$ shutdown 8:00

Schedule the system to shut down at 8 A.M.

$ shutdown 20:00

Schedule the system to shut down at 8 P.M.

$ shutdown now

Bring down the system immediately.

$ shutdown -r now

Bring down the system immediately, and automatically reboot it.

$ shutdown -P now

Bring down the system immediately, and automatically power off the system.

4.7 REDIRECTION

Redirection is a special feature of Linux system through which one can direct the input or output to some other stream other than the standard ones. The redirection operators in Linux include >, < and >> where < means the input redirection, > specifies the output redirection operator and >> operator appends output to an existing file.

The input redirection operator (<) instructs the shell to redirect the input. It can be used to take input from a file. Suppose the thisfile.txt contains the following texts.

This is a text file.

It is used to test the cat and the wc commands.

The redirection and the piping can also be tested by using this file.

And this is the end of the content of the file.

The wc command will count and display the number of lines, words and characters in the file thisfile.txt.

$ wc < thisfile.txt

4 40 187

Here, 4 is the number of lines, 40 is the number of words and 187 is the number of characters.

The output of any command can be redirected by using the output redirection operator. For example, the command:

$ cat thisfile.txt > thatfile.txt

will write the contents of the file into another file thatfile.txt.

The input and output redirection operators can also be combined to

redirect both the input and output. For example,

$ wc < thisfile.txt > count_file.txt

command will save the number of lines, word and characters of the file ''thisfile.txt'' into another file ''count_file.txt''.

$ cat count_file.txt

4 40 187

When a command begins running, it usually expects that three files are already open: standard input, standard output, and standard error. A number, called a file descriptor, is associated with each of these files, as follows:

File descriptor 0 means Standard input

File descriptor 1 means Standard output

File descriptor 2 means Standard error (diagnostic) output

A child process normally inherits these files from its parent. All these three files are initially assigned to the peripherals like 0 to the keyboard, 1 and 2 to the display. The shell permits them to be redirected elsewhere before control is passed to a command.

When you enter a command and no file name is given then your keyboard is the standard input, denoted as stdin. When a command finishes, the results are displayed on your screen. This is called the standard output, denoted as stdout. By default, commands take input from the standard input and send the results to standard output. Standard Error, denoted as stderr, is where error messages go. By default, this refers to monitor screen.

4.7.1 Standard Input

The standard input is a character stream that comes from a particular source. A source may be a device or a file or a program or command. Some commands (like cat and wc) can operate on a character stream without knowing the source of the stream. If we write cat and wc command without any argument as follows-

cat

wc

Then these commands will receive character stream from the default source (i.e., from the keyboard). This stream is called standard input. The sources from which the standard input can come from are-

- the keyboard, which is the default source.
- a file (given by using the redirection operators >, <, >>)
- a program or command (given by using pipe)

For example,

$ wc
Hello, I am wc command.
can count the number of lines, words and characters.
<Ctrl d>
2 15 79
$ wc < thisfile.txt
4 40 187

4.7.2 Standard Output

Standard output writes the output generated by a program or command in the form of a character stream. When the standard output is not redirected, it will send the output to the terminal, which is the default destination.

For example, if you write the cat and wc command as follows-

cat
wc

Then these commands will send the output as a character stream to the terminal. This stream is called the standard output. Here, the shell sets up a connection between the command and the terminal to display the output on the monitor screen. Again, the standard output can be redirected.

For example,

$ wc thisfile.txt > thatfile.txt

This command redirects the number of lines, number of words and the number of characters in the file thisfile.txt to

the file thatfile.txt.

The standard output can also serve as input to another program. Thus, the standard output has three destinations - the terminal (which is default destination), a file (redirected by using redirection operator) and another command or program (given by suing the concept of piping).

4.7.3 Standard Error

The standard error writes error messages generated by a program or command that fails at some point in its execution. It writes the error message to the terminal. Like standard output, the default destination of standard error is the terminal display. The output redirection operator can also be used to redirect the error to a file. But it needs to prefix the file descriptor. A file descriptor is a number assigned to a standard file that identifies the standard file. The file descriptors 0,1 and 2 represent the standard input, standard output and standard error respectively.

For example,

$ cat testfile.txt > newfile.txt

cat: testfile: No such file or directory

$ cat testfile 2> newfile

Here, standard error has been redirected by using file descriptor.

$ cat newfile

cat: testfile: No such file or directory

4.8 PIPING

A pipe is a form of redirection (transfer of standard output to some other destination) that is used in Linux. Pipes are used to redirect a standard input or standard output from one command or program to another for further processing. The shell enables the standard output of one command to be connected as a standard input to another. This is called piping.

Piping is done by using a special operator "|" called pipe that acts

as the connector of two commands. Here, the output of one command acts 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. Consider, there are four text files which are listed as:

$ ls

file1.txt file2.txt file3.txt file4.txt

$ ls | wc

4 4 40

Here, the ls command passes its output directly to the input of wc then the wc command gives the output as above.

$ ls file1.txt | wc

1 1 10

$ cat file1.txt | wc

1 5 19

Here, the cat command passes its output (i.e., the content of the file file1.txt) to the input of wc command and wc command gives the final output which is displayed on the screen.

Pipes are unidirectional i.e., data flows from left to right through the pipeline. The syntax is as follows:

command_1 | command_2 | command_3 | .... | command_N

For example, to list all files and directories and give it as input to more command, the following command with pipe displays the output as given below.

$ ls -l | more

Output is (Fig. 4.3):

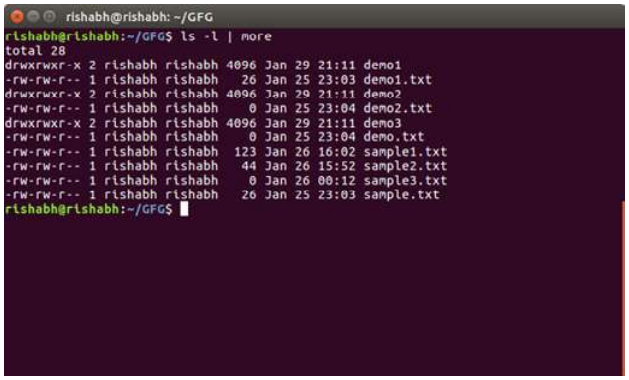


Fig. 4.3: Output of the command "ls -l | more"



CHECK YOUR PROGRESS

Choose the correct option and fill up the blanks:

Q 1: Redirection is a process of switching of the standard stream of data.

a) True b) False

Q 2: How many files are used for representing different standard streams?

a) 1 b) 2 c) 4 d) 3

Q 3: Which stream is connected to the display?

a) standard input b) standard output

c) standard error d) None of these

Q 4: The command wc < sample.txt will count data from the file sample.txt.

a) True b) False

Q 5: Which of the following symbol(s) can be used to redirect the output to a file or another program?

a) | b) > c) >> d) |, > and >>

Q 6: Device Drivers reside in which environment?

a) Kernel space b) Device space

c) Application space d) User space

Q 7: Which command is used to display the operating system name

a) os b) unix c) kernel d) uname

Q 8: A ___ is a special program that can control a specific type of hardware. (Pick the right option)

a) Shell b) Device driver c) Complier d) Database

Q 9: _____ directory has system configuration files, network configuration files, application configuration files and other installed software. (Pick the right option)

a) /var b) /mnt c) /opt d) /etc

Q 10: In Linux, devices are placed in _____ directory. (Fill up the blank)



4.9 LET US SUM UP

- Kernel is the core component of Linux system that interact with the hardware.
- Each device is considered a file.
- A device driver is a routine of codes specific to a particular device that performs the control operation for that device.
- In Linux, the system start-up process goes through a series of stages. At first, the kernel is loaded into the memory. BIOS (Basic Input/Output System) then performs some system integrity checks. It searches, loads, and executes the boot loader program from floppy, CD-ROM, or hard drive.
- Shutdown command is used with several options for shutting down a system.
- Redirection is used to redirect the output to some other stream other than the standard inputs and outputs.
- The standard inputs, standard outputs and the standard errors are the character streams.
- Piping is the technique of redirecting a standard input or output from one command to another. Piping is done by using a special operator "|" called pipe that acts as the connector of two commands.



4.10 FURTHER READINGS

1) Das, S. (2000).UNIX: Concepts and Applications. Tata McGraw-Hill.
2) Negus, C. (2012). Linux Bible. (Vol. 772). John Wiley & Sons.



4.11 ANSWERS TO CHECK YOUR PROGRESS

Ans to Q No 1: Q.1) a Q.2) d Q.3) b Q.4) a
Q.5) c Q.6) a Q.7) d Q.8) b Q.9) d
Q.10) /dev



4.12 MODEL QUESTIONS

Q 1: Define Kernel? Describe the functions and the architecture of the Linux kernel.

Q 2: What do you mean by device driver? How can you identify whether a device file is block or character device?

Q 3: Describe the phases performed by the start-up process of Linux system.

Q 4: Describe the various options used with the shutdown command in Linux.

Q 5: What are redirection and piping? Explain the functionality of these two techniques.

Q 6: What do you mean by file descriptor? State the difference between standard input, standard output and the standard error.

*** ***** ***

UNIT 5: BASICS OF LINUX FILE SYSTEM

UNIT STRUCTURE

5.1 Learning Objectives

5.2 Introduction

5.3 Basics of Linux File System

- 5.3.1 ext
- 5.3.2 ext2
- 5.3.3 ext3
- 5.3.4 ext4
- 5.3.5 xfs
- 5.3.6 jfs
- 5.3.7 Reiser FS
- 5.3.8 ISO 9660

5.4 Linux File System Internal Structure

5.5 How Linux Access Files

5.6 Storage of Files

5.7 Linux Directory Structure

5.8 LILO, GRUB Boot Loader

5.9 Basic Types of Files

- 5.9.1 Regular Files
- 5.9.2 Derectory Files
- 5.9.3 Special or Device Files

5.10 Other Common Types of Files

5.11 File Attributes

5.12 Let Us Sum Up

5.13 Further Readings

5.14 Answer to Check Your Progress

5.15 Model Questions

5.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- define file and file system

- describe different file system types
- explain the features of different file system and boot loader
- learn about boot block, super block, inode and data block
- describe the journaling concept in file system
- know about various file types used in Linux system
- describe different file attributes.

5.2 INTRODUCTION

In the previous units, the features of Linux Operating System and the working of Linux Kernel were described. Also, the redirection and piping concepts in Linux Operating System were discussed. In this unit, we will study the Linux File System and the related properties. Subsequently, the functions of different file systems in Linux will be explained here in this unit. Further, we will study the different types of files, I-node and the file attributes.

5.3 BASICS OF LINUX FILE SYSTEM

In Linux, all data and programs are stored in files. No restriction is imposed on the structure of a file. The size of a linux file is the exact size of the contents that stored inside the file. The files in a file system are connected logically. A file system manages how the files are stored and can be retrieved easily. Without a file system, it is not possible for an operating system to know the positioning of a file i.e., where a file starts and where ends and thus the operating system keeps track of how files are organised in a disk partition so that it can be found easily. There are different file systems available in linux operating system. They are different from each other based on the structure, logic and other important properties like flexibility, speed, size etc. Some of the file systems that are being commonly used for better understanding of the functionalities of a file system are described as follows.

Any Linux distribution has different kinds of file system to partition a disk. Some of the commonly used file systems include the following:

- ext

- ext2
- ext3
- ext4
- xfs
- JFS
- ReiserFS
- ISO9660 etc.

5.3.1 ext

ext stands for extended file system. It is considered the first file system used in Linux. It was developed by Remy Card in April, 1992 especially for Linux Operating System. It is the first version in the series of extended file systems. It was replaced by ext2 in 1993. The ext file-system uses a system called inodes to track information about the files stored in the virtual directory. The inode system creates a separate table on each physical device, called the inode table, to store the file information. Each stored file in the virtual directory has an entry in the inode table. Linux maintains each inode in the inode table with a unique number (called the inode number), assigned by the file system.

5.3.2 ext2

ext2 is meant for second extended file system created for Linux OS and it is an improved version of ext. It was also designed by Remy Card. It is one of the most popular and commonly used file systems in Linux and has been used for many years. Here, the data stored in files are kept in data blocks. All the data blocks have same length. However, the length is not fixed in all ext2 file systems. The size of the file system is depended on the block size which may vary from 1 KB to 8 KB. The maximum size of the ext2 file system is from 2 TB to 32 TB whereas the maximum file size is from 16 GB to 2TB. The size of the filename may be of maximum of 255 characters.

The limitation of ext2 file system is that it takes very long time to recover after an unclean shutdown which may occur due to situations like power failure or system crash. This problem leads to development of the improved version of ext2.

5.3.3 ext3

ext3 is the third extended file system in Linux. It is an extended version of ext2 file system which was introduced by Stephen Tweedie in November 2001 with Linux. It overcomes the limitation of ext2 file system by introducing the concept of journaling. Moreover, ext3 also adds two more features which are the growth of online file system and Tree indexing for larger directories. An ext3 file system works similar to ext2 file and an ext2 file system can easily be upgraded to ext3. It uses the same inode table structure as the ext2 filesystem, but adds a journal file to each storage device to journal the data written to the storage device. This Ext3 file system is used as default file system in many standard Linux distributions.

A journaling file system handles a special file known as a journal. It is used to keep track of changes that are not committed to the file system completely. It is used for correcting the inconsistencies that may occur at the time of system crash or power failure. It may keep track of both stored data and metadata. This file system keeps the information in a log area. After the updating of log, the system wrote the actual data in the appropriate areas of the file system. If a file system is crashed out, it can be repaired very quickly back online using the journal log. So, there is a very little chance of data loss in journaling file system.

ext2 file system runs a program called e2fsck (ext2 file system check) at the time of rebooting after an unclean shutdown. It checks whether any problem occurs during the improper shutdown and tries to fix the problems before the file system be mounted. Thus, ext2 tries to maintain the data integrity. But the problem is

that e2fsck checks the entire file system which may take a long time to recover. The main advantage of ext3 file system is that it supports journaling. This file system does not need any checking in the file system at the time of rebooting after an unclean shutdown unlike ext2 file system since it guarantees the internal consistency of the system.

The size of the file system and file size is similar to file system of ext2. The maximum size of the ext3 file system is from 4 TB to 32 TB whereas the maximum size of a file is 16 GB to 2 TB. The filename size is also same as ext2. The disadvantage of the ext3 file system is that it occupies more system memory to store the information and can make the I/O system little slower. In spite of this, many of the distributions use ext3 file system for its faster recovery.

5.3.4 ext4

The result of expanding the ext3 file system is the ext4 file system. The ext4 file system was officially supported in the Linux kernel in 2008, and is now the default file system used in most popular Linux distributions, such as Fedora and Ubuntu. It was developed over ext3 in order to increase the size of the file and entire file system to improve the performances. The various features of ext4 file system are as follows:

- **Compatibility:** Any ext3 file system can be upgraded to ext4 easily. So, we can easily increase the size of the file system and improve the performance without doing any formatting or reinstallation of the operating system. Ext4 uses the new data structures for the new data only. The old data can be modified whenever required.
- **Extents:** ext2 and ext3 file systems use an indirect block mapping scheme to keep track of blocks used for particular data. It is not efficient for the larger files. An extent is a collection

of contiguous physical blocks. A single extent can have a contiguous block size of 128 MB where size of a single block is 4 KB.

- **Multiblock Allocation:** In ext file system, a program block allocator decides which free block will be used at the time of writing a new data. But it allows only one block at a time. So, it is not efficient technique for writing a file of large data size. To improve the performance, ext4 uses multiblock allocator which allows to use more than one free block at a single call.
- **Journal Checksumming:** ext4 file system checksums the journal data as it is one of the most used files in the disk. It is used to check whether any journal block has corrupted or not.
- **Delayed Allocation:** In ext4, blocks are not allocated to data immediately. There is a delay in allocating the block unlike ext3 and the files are kept in cache. In this case, block allocator can optimize the situation.
- **No Journal Mode:** A journaling file system keep a track of information of the current changes. For this, it occupies little storage. Some programs may have special requirement where it can run without journal. Accordingly, journaling option can be disabled in ext4 file system.
- **Larger File System and File Sizes:** ext4 file system has significantly larger sizes than ext3 file system. A file in ext4 can have a maximum size of 16 TB and on the other side, the file system has maximum size of 1 EiB or 1 ExbiBytes [1 EiB = 260 bytes].

5.3.5 xfs

xfs file system is a high-performance and highly scalable 64 bit system. It supports journaling and thereby it ensures the consistency of the data. It was created by Silicon Graphics and introduced in 1994 for its commercial IRIX Unix system. It was used in Linux

kernel in 2001. This file system is supported by almost all Linux distributions. The xfs file system uses the writeback mode of journaling, which provides high performance and also allows online resizing of the file system. For space allocation, extents with stored data structure in B+ tree is used. One of the unique features of xfs file system is pre-allocation of input-output bandwidth at previously determined rate which may be beneficial for real time applications. xfs file system has block size of 512 bytes to 64 KB.

5.3.6 jfs

JFS is considered one of the oldest journaling file systems. It is a 64-bit file system created by IBM. It's first version was developed for AIX operating system of IBM in 1990 known as jfs1. JFS2 was introduced in 2001 with Linux OS. The term "journaled" depicts that the file system keeps the record of ongoing modification of files in a log file. This log file is used to recover the loss that may occur during an improper shutdown. JFS uses the ordered journaling method, storing only the inode table data in the journal, but not removing it until the actual file data is written to the storage device. In addition, JFS file system uses extent-based file allocation, allocating a group of blocks for each file written to the storage device. This method provides for less fragmentation on the storage device. It uses B+ tree to search in directories. Since ext4 shows better performance compared to JFS, so it is not widely used in Linux.

5.3.7 ReiserFS

In 2001, Hans Reiser created the first journaling file system for Linux, called ReiserFS. The ReiserFS file system only supports writeback journaling mode, writing only the inode table data to the journal file. Due to this feature, the ReiserFS file system is considered one of the fastest journaling file systems in Linux. Two interesting features available with the ReiserFS file system refer

the following.

- We can resize an existing file system while it's still active, and
- It uses a technique called tail packing, which stuffs data from one file into empty space in a data block from another file and thereby reduces internal fragmentation.

5.3.8 ISO 9660

ISO 9660 is a file system created for optical media. It is published by International Standard Organization (ISO). The main aim of this file system is that it can support in any other standard operating system like Linux, Windows and Mac etc. for exchanging the data between the media and the system. This file system standard evolved from the High Sierra Format file system. It has three different levels for files:

- **Level 1** - supports only 8.3 filenames (eight characters with a three-character extension).
- **Level 2** - supports longer 32-character filenames.
- **Level 3** - supports longer 32-character filenames.

5.4 LINUX FILE SYSTEM INTERNAL STRUCTURE

Linux files have disk partitions or blocks containing file data. Linux File System is divided mainly into 4 blocks.

- Boot block
- Super block
- inode
- Data Block

Partition is divided into data blocks with of fixed size which are used to store content of file. In case if file is of greater size than of block then it is stored in multiple blocks.

| Boot Block | Super Block | Inodes | | | ... | Data Blocks |

Boot Block: It is located in the first few sectors of a file system. The boot block contains the initial bootstrap program used to load the

operating system. Typically, the first sector contains a bootstrap program that reads in a larger bootstrap program from the next few sectors, and so forth. The boot block is usually a part of the disk label, a special set of blocks containing information on the disk layout. The boot block holds the loader to boot the operating system.

Super Block: Each filesystem has one super block. The superblock contains the basic information about the entire file system. This includes the size of the file system, the list of free and allocated blocks, the name of the partition, and the modification time of the filesystem.

Inodes: Information about each file in the filesystem is kept in a special kernel structure called an inode. The inode contains a pointer to the disk blocks containing the data in the file, as well as other information such as the type of file, the permission bits, the owner and group, the file size, the file modification time, and so on. The inode does not, however, contain the name of the file. The name of each file is listed in the directory the file is associated with. Every file has one inode and vice versa. In linux, every file is recognized with integer number known as inode number. Each inode has an entry in inode table. An inode provides following information about a file:

- File Types (i.e., regular, directory, device file or other)
- File Permission or access modes (i.e., read, write or execute to different users)
- User ID of the owner
- Group ID of the group
- Last access, last modification detail
- Number of soft links and hard links
- Size of the file
- Location or path of the file
- Other related data

We can find the inode number of a particular file using the following Linux command:

ls -i <filename>

When a program or a user is referred by its name, the operating

system maps the filename with its corresponding inode. Then, through the inode further operations in the file is performed.

Inodes include pointers to the data blocks. Each inode contains 15 pointers:

- the first 12 pointers point directly to data blocks
- the 13th pointer points to an indirect block, a block containing pointers to data blocks
- the 14th pointer points to a doubly-indirect block, a block containing 128 addresses of singly indirect blocks
- the 15th pointer points to a triply indirect block (which contains pointers to doubly indirect blocks, etc.)

Data Blocks: Data blocks start at the end of the inode list and contain file data and directory blocks. An inode can directly or indirectly reference three kinds of data blocks. All referenced blocks must be of the same kind. The three types of data blocks are:

- Plain data blocks
- Symbolic-link data blocks
- Directory data blocks

Plain data blocks contain the information stored in a file. Symbolic-link data blocks contain the path name stored in a symbolic link. Directory data blocks contain directory entries. fsck can check the validity only of directory data blocks.

Directories are distinguished from regular files by an entry in the mode field of the inode. Data blocks associated with a directory contain the directory entries. Directory data blocks are checked for inconsistencies involving:

- Directory inode numbers pointing to unallocated inodes
- Directory inode numbers greater than the number of inodes in the file system
- Incorrect directory inode numbers for "." and ".." directories
- Directories disconnected from the file system

5.5 HOW LINUX ACCESS FILES

Linux file system is generally a built-in layer of Linux operating system used to handle the data management of the storage. It helps to arrange the file on the disk storage. It manages the file name, file size, creation date, and much more information about a file. A file system is designed in a way so that it can manage and provide space for non-volatile storage data. All file systems required a namespace that is a naming and organizational methodology.

In Linux, the file system creates a tree structure. All the files are arranged as a tree and its branches. The topmost directory called the root (/) directory. All other directories in Linux can be accessed from the root directory.

Some key features of Linux file system are as following:

- **Specifying paths:** Linux does not use the backslash (\) to separate the components; it uses forward slash (/) as an alternative. For example, as in Windows, the data may be stored in C:\ My Documents\ Work, whereas, in Linux, it would be stored in /home/ My Document/ Work.
- **Partition, Directories, and Drives:** Linux does not use drive letters to organize the drive as Windows does. In Linux, we cannot tell whether we are addressing a partition, a network device, or an "ordinary" directory and a Drive.
- **Case Sensitivity:** Linux file system is case sensitive. It distinguishes between lowercase and uppercase file names. Such as, there is a difference between test.txt and Test.txt in Linux. This rule is also applied for directories and Linux commands.
- **File Extensions:** In Linux, a file may have the extension '.txt,' but it is not necessary that a file should have a file extension. While working with Shell, it creates some problems for the beginners to differentiate between files and directories. If we use the graphical file manager, it symbolizes the files and folders.
- **Hidden files:** Linux distinguishes between standard files and hidden files, mostly the configuration files are hidden in Linux OS. Usually, we

don't need to access or read the hidden files. The hidden files in Linux are represented by a dot (.) before the file name. To access the files, we need to change the view in the file manager or need to use a specific command in the shell.

5.6 STORAGE OF FILES

File systems are defined based on where they are used. There are file systems defined for operating systems, networks, databases, and other special-purpose file systems. A filesystem is designed to provide space for non-volatile storage of data. All filesystems need to provide a namespace with a name and organizational methodology. This defines how a file can be named, specifically the length of a filename and the subset of characters that can be used for filenames out of the total set of characters available. It also defines the logical structure of the data on a disk, such as the use of directories for organizing files instead of just lumping them all together in a single, huge conglomeration of files.

Once the namespace has been defined, a metadata structure is necessary to provide the logical foundation for that namespace. This includes the data structures required to support a hierarchical directory structure; structures to determine which blocks of space on the disk are used and which are available; structures that allow for maintaining the names of the files and directories; information about the files such as their size and times they were created, modified or last accessed; and the location or locations of the data belonging to the file on the disk. Other metadata is used to store high-level information about the subdivisions of the disk, such as logical volumes and partitions. This higher-level metadata and the structures, it represents, contain the information describing the filesystem stored on the drive or partition, but is separate from and independent of the filesystem metadata.

File systems also require an Application Programming Interface (API) that provides access to system function calls which manipulate filesystem objects like files and directories. APIs provide for tasks such as

creating, moving, and deleting files. It also provides algorithms that determine things like where a file is placed on a filesystem. Such algorithms may account for objectives such as speed or minimizing disk fragmentation. Modern filesystems also provide a security model, which is a scheme for defining access rights to files and directories. The Linux filesystem security model helps to ensure that users only have access to their own files and not those of others or the operating system itself. The final building block is the software required to implement all of these functions. Linux uses a two-part software implementation as a way to improve both system and programmer efficiency.

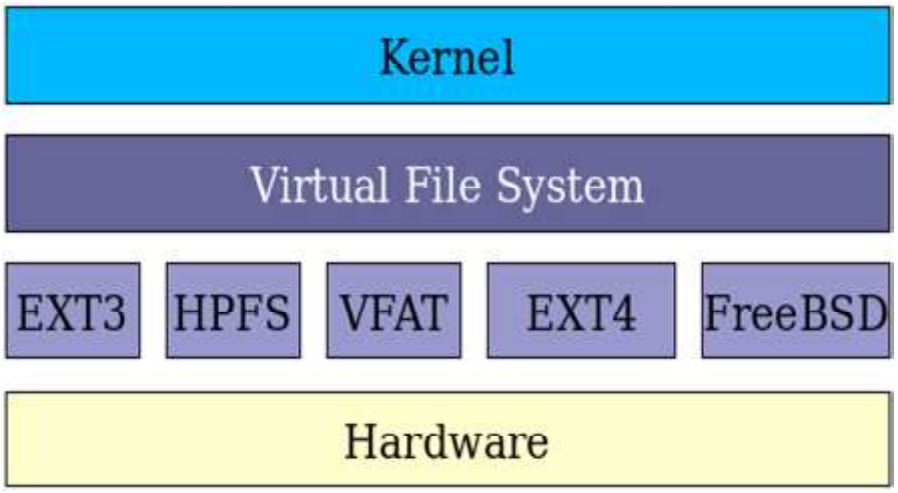


Fig. 5.1: The Linux two-part filesystem software implementation.

The first part of this two-part implementation is the Linux Virtual File System. This virtual file system provides a single set of commands for the kernel, and developers, to access all types of file systems. The virtual file system software calls the specific device driver required to interface to the various types of file systems. The file system-specific device drivers are the second part of the implementation. The device driver interprets the standard set of file system commands to ones specific to the type of file system on the partition or logical volume.

5.7 LINUX DIRECTORY STRUCTURE

The File System Hierarchy Standard (FHS) defines the directory structure and directory contents in Linux distributions. It is maintained by

the Linux Foundation. In the FHS, all files and directories appear under the root directory /, even if they are stored on different physical or virtual devices. Some of these directories only exist on a particular system if certain subsystems, such as the X Window System, are installed.

Most of these directories exist in all Unix-like operating systems and are generally used in much the same way. However, the descriptions here are those used specifically for the FHS, and are not considered authoritative for platforms other than Linux. Some standard Linux directories with their description are listed in the following table (Table-5.1).

Directory	Description
/	Primary hierarchy root and root directory of the entire file system hierarchy
/bin	Essential command binaries that need to be available in single user mode; for all users, *e.g.*, cat, ls, cp.
/boot	Boot loader files, *e.g.*, kernels, initrd.
/dev	Device files, *e.g.*, /dev/null , /dev/disk0 , /dev/sda1 , /dev/tty , /dev/random.
/etc	Host-specific system-wide configuration files.
/home	Users' home directories, containing saved files, personal settings, etc.
/sys	Contains information about devices, drivers, and some kernel features.
/usr	contains the majority of (multi-)user utilities and applications
/var	Contains Variable files such as logs, spool files, and temporary e-mail files.

Table 5.1: Few standard Linux directories with description

What the file system looks like is represented graphically in the following figure 5.2. Some of these directories have standard files as well like /bin, /etc and /sbin specifically.

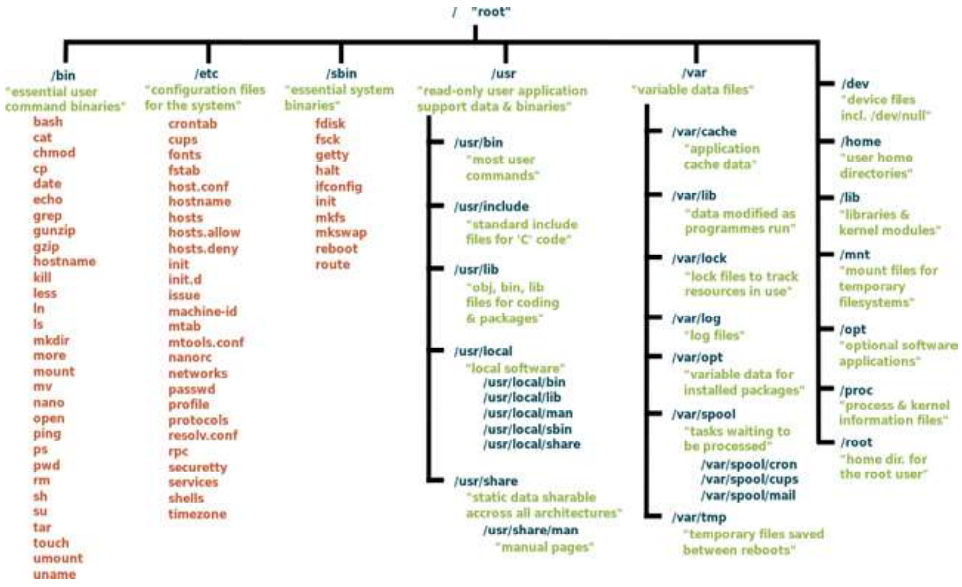


Fig. 5.2 Linux Directory structure

5.8 LILO, GRUB BOOT LOADER

A boot loader refers to a program which is responsible for loading the Linux kernel with optional kernel parameters and the Linux initial RAM disk, known as initrd. Linux kernel is the core of the Linux operating system, and it initiates the init process immediately after being loaded. The Linux initial RAM disk provides a temporary storage space for loading critical files into memory before the real root file system can be mounted. On older computers with BIOS (Basic Input/Output System), a boot loader resides in the MBR (Master Boot Record), which occupies the first 512 bytes on a disk, but newer computers with UEFI (Unified Extensible Firmware Interface) store it in a special partition called EFI System Partition.

A boot loader is loaded by BIOS (Basic Input Output System) after a successful POST (Power-On Self-Test), which is a self-test process performed immediately after a computer or other digital electronic device is powered on. There are several boot loaders in Linux system. Among them, two mostly used Linux loader programs are described here.

LILO (Linux Loader)

LILO once used to be the most popular Linux boot loader, but it has gradually fallen from beauty because it lacked support for multi-boot

environments. The development of LILO was stopped in December 2015, and the limited support for modern file systems and non-existing support for UEFI both reflect this reality.

GRUB (GRand Unified Bootloader)

GRUB is the most popular and the most feature-packed boot loader for the Linux operating system. It is based on the obsolete GRUB Legacy, which was created in 1995 by Erich Boleyn for the operating system GNU/ Hurd. GRUB supports both BIOS and UEFI, and it can handle all popular Linux file systems, including ext4, ReiserFS, VFAT, and XFS. Compared to GRUB Legacy, modern GRUB is cleaner, more powerful, and safer. GRUB has been designed to be compatible with the multiboot specification which allows GRUB to boot many versions of Linux and other free operating systems.

GRUB can also allow the user to choose to boot from among several different kernels for any given Linux distribution. This affords the ability to boot to a previous kernel version if an updated one fails somehow or is incompatible with an important piece of software. GRUB can be configured using the /boot/grub/grub.conf file.

5.9 BASIC TYPES OF FILES

Linux operating systems have different type of files. Most of the commonly used file types are regular files, directory files and device files.

5.9.1 Regular Files

Regular files or ordinary files are the mostly used files in Linux system. All the text files, binary files, executable program files, image files, shared libraries etc. are known as regular files. Regular files are denoted by a dash ('-') symbol. Regular files can be created using touch, vi, gedit commands.

For listing the regular files following Linux command is used:

ls - l | grep ^-

Where grep is a filter which will find all the regular files which are

denoted by '-' symbol.

5.9.2 Directory Files

Linux stores all the data and programs in files. These files are contained in directories. A directory contains all the regular files, special files, other folders stored in a physical device. These files are denoted by symbol 'd'. Directories can be listed using the following Linux command:

ls - l | grep ^d

Some of the important commands related with directories are given below:

mkdir : used for creating a new directory.

Syntax: mkdir <directory name>

rmdir: used for removing or deleting any empty directory from the system.

Syntax: rmdir <directory name>

cd: used for changing the directory.

cd <directory name>

5.9.3 Special or Device Files

Device files or special files in Linux are the files that work as an interface to device drivers. It allows an application to interact with a device via any device driver. Most of the device files are stored in / dev directory in Linux. Each device in the system should have an entry in /dev. These files are created during installing the system and device drivers. There are two different types of device files present in Linux operating systems. These are: character special files and block special files.

- **Character Special Files:** They are also known as character devices. Their behaviour is like serial ports, pipes etc. They provide a direct access to hardware devices. They do not allow any program to perform read or write operation. It is up to the

device. Sometimes character devices are known as raw devices also. Character devices are symbolically denoted by 'c'. To list all the character files locate in /dev directory, following Linux command can be used:

ls -l /dev | grep ^c

- **Block Special Files:** They are also known as block devices. These files differ from the character devices in the fact that a block device allows a program to read or write a block of any size. These are buffered access to hardware devices whereas character devices are unbuffered access. Block devices are denoted by 'b'.

To list all the symbolic link files located in /dev directory, following Linux

command can be used:

ls -l /dev | grep ^b



CHECK YOUR PROGRESS

Q 1: Choose the correct options from the followings:

i) Which among the following allows fast file system recovery?

a) Ext2 b) Journaling c) Caching d) Sysfs

ii) Filesystem for CDROM is:

a) Ext2 b) Ext3 c) ISOfs d) Procfs

iii) Which file contains the filesystems to be automatically mounted during boot?

a) /etc/mount b) /etc/fstab c) /etc/inittab d) /etc/boot

iv) Which command is used to check filesystem usage in a system?

a) mount

b) df c) du d) dd

v) In which of the following directory does the configuration files are present?

a) /bin/ b) /root/ c) /etc/ d) /dev/

5.10 OTHER COMMON TYPES OF FILES

Apart from these three types of files, Linux system offers some other type of files which are:

- **Symbolic Links:** A symbolic link is a reference to another file located in a disk. This file is stored as referenced file's path. These files are denoted by 'l', also known as soft link. Each soft link contains a different inode value which directs to the original file. A hard link has the same inode value as the original file. For this, it remains linked even the original file is moved to another location unlike a soft link file. To list all the block files located in /dev directory, following Linux command can be used:

  ls -l /dev | grep ^b

- **Named Pipes:** It performs communication between two or more processes locally with each other using file. This file acts as a pipe among the processes. These are also known as FIFO. Named pipes are denoted using 'p'. To list all the named pipe files located in /dev directory, following Linux command can be used:

  ls -l /dev | grep ^p

- **Socket Files:** Socket files are used for inter process communication. It is used to pass the information between applications. The main difference between socket and named pipe is that a socket can perform communications between processes which are running on two different systems. But named pipes perform communication locally. A socket file can be created using a system call named socket (). Socket files are denoted by 's'.

To list all the named pipe files located in /dev directory, following Linux command can be used:

ls -l /dev | grep ^s

5.11 FILE ATTRIBUTES

Linux system is a typical multi-user system. Different users in a

different position, with different privileges can access Linux OS. File attributes are used along with the computer files as settings. They are used to either allow or restrict certain operations for a user or for the operating system. To protect the security of the system, Linux system provides file permissions for different users to do different provisions to access the same files (including the catalogue file). The files and directories in Linux can have following attributes:

- **a - append only:** this attribute allows a file to be added to, but not to be removed. It prevents accidental or malicious changes to files that record data, such as log files.
- **c - compressed:** it causes the kernel to compress data written to the file automatically and uncompress it when it is read back.
- **d - no dump:** it makes sure the file is not backed up in backups where the dump utility is used
- **e - extent format:** it indicates that the file is using extents for mapping the blocks on disk.
- **i - immutable:** it makes a file immutable, which goes a step beyond simply disabling write access to the file. The file can't be deleted, links to it can't be created, and the file can't be renamed.
- **j - data journaling:** it ensures that on an Ext3 file system the file is first written to the journal and only after that to the data blocks on the hard disk.
- **s - secure deletion:** it makes sure that recovery of a file is not possible after it has been deleted.
- **t - no tail-merging:** Tail-merging is a process in which small data pieces at a file's end that don't fill a complete block are merged with similar pieces of data from other files.
- **u - undeletable:** When a file is deleted, its contents are saved which allows a utility to be developed that works with that information to salvage deleted files.
- **A - no atime updates:** Linux won't update the access time stamp when you access a file.

- **D - synchronous directory updates:** it makes sure that changes to files are written to disk immediately, and not to cache first.
- **S - synchronous updates:** the changes on a file are written synchronously on the disk.
- **T - and top of directory hierarchy:** A directory will be deemed to be the top of directory hierarchies for the purposes of the Orlov block allocator.

Linux provides us the access control by file and directory permissions on three levels which are user, group and other. These file permissions provide the basic level of security and access control. In Linux, we can use the ls -l command to display user and group attributes of a file and the file belongs to, such as:

```
[root@www /]# ls -l
total 64
dr-xr-xr-x   2 root root 4096 Dec 14  2012 bin
dr-xr-xr-x   4 root root 4096 Apr 19  2012 boot
```

Example, the first attribute in bin and boot file with the "d" represents that it is a directory file.

In Linux, the first character represents the file as a directory, the ordinary file or linked files or device file such as

- the [d] is the directory
- the [-] is the ordinary file;
- If it is [l] then it refers to a linked document (link file);
- If it is [b], it indicates that it is a block device;
- the [c] represents the character device such as a keyboard, a mouse (one reading device).

The next character in groups of three, and are a combination of "rwx" of the three parameters. Wherein, [r] represents read permission, [w] represents write and [x] means execute permission. It should be noted that the position of these three rights will not change. If there is no authority, there will be a minus sign [-] only.



CHECK YOUR PROGRESS

Q 2: Choose the correct options from the followings:

i) Which of the following is true about files?

A) All data in Unix is organized into files.

B) All files are organized into directories.

C) The directories are organized into a tree-like structure called the filesystem.

D) All of the above

ii) Each entry of directory file has component(s) namely _____.

A) filename B) inode number C) file size

D) Both A and B

iii) Which of the following symbol represents the current directory?

A) Double dot (..) B) Dollar dot($.) C) Single dot (.)

D) Hash dot(#.)

iv) The directory files have:

A) File names & File location B) File names & Inode Numbers

C) File names & Permissions

D) File names & Address v) Device files have following two types?

A) Character & Block B) Character & Socket

C) Block & FIFO D) Only A



5.12 LET US SUM UP

- A file is a collection of related data stored in a computer.
- A file system is used to store and organize all the files in a Linux system.
- ext2 file system runs e2fsck program for recover the file after any unclean shutdown.
- ext3 is an extended version of ext2 file system. Journaling concept is introduced in ext3.
- A journal is used to correct the inconsistencies which can occur in case improper or unclean shutdown.

- ext4 file system has some improvised features such as multiblock allocation, delayed allocation, no journaling mode, extents etc.
- ISO 9660 is a file system created for optical media.
- A regular file contains the entire text, binary, image, program and other files.
- Directory contains all the files and folders, sub folders. We can create, delete directories using mkdir and rmdir command respectively.
- cd command is used to change the current working directory of the system.
- A special file called device file is located in /dev directory of the file system. Character device and block device are two types of special files.
- A socket and named pipe file are used for inter process communication.
- Symbolic link file is a reference to a file located in the disk.
- An inode or index node contains all the details about a file. The inode files can be displayed by using ls -i command.
- File attributes are some setting options which allow or restrict specific operations on a file.



5.13 FURTHER READINGS

1) Das, S. (2000).UNIX: Concepts and Applications. Tata McGraw-Hill.
2) Bovet, D. P. & Cesati, M. (2005). Understanding the Linux Kernel:from I/O ports to process management. "O'Reilly Media, Inc."



5.14 ANSWERS TO CHECK YOUR PROGRESS

Ans to Q No 1: i. b ii. c iii. b iv. b v. c

Ans to Q No 2: i. D ii. D iii. C iv. B v. D



5.15 MODEL QUESTIONS

Q 1: What are the purposes of using file system in Linux?

Q 2: Name the different series of extended file system.

Q 3: Compare the features of ext2, ext3 and ext4 file system.

Q 4: What is the limitation of ext2 file system? How it is overcome by ext4 file system?

Q 5: What is journaling file system?

Q 6: Explain the features of ext4 file system.

Q 7: Explain any three commonly used file types in Linux.

Q 8: Differentiate between following:

- Hard link vs. Soft link
- Named pipe file vs. Socket files
- Character files and Block files

Q 9: Explain the functions of inode in Linux file system.

Q 10: Define file attributes. List the different type of attributes used in Linux with their functions.

Q 11: Describe different types of file systems used in Linux.

Q 12: Write short notes on following file systems:

- xfs
- JFS
- ReiserFS.

Q 13: Define boot loader. Explain various boot loaders used in Linux system.

Q 14: Discuss about Linux directory structure.

*** ***** ***

UNIT 6: FILE ORGANIZATION

UNIT STRUCTURE

6.1 Learning Objectives

6.2 Introduction

6.3 Files and File System

6.4 Working with Files

6.5 File System Mounting and Unmounting,

6.6 Organization of the File Tree

6.7 Disk Related Commands

6.8 Let Us Sum Up

6.9 Further Readings

6.10 Answer to Check Your Progress

6.11 Model Questions

6.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- to distinguish different files system,
- to cope up with different commands to work on file,
- understand basic file organization of Linux,
- to learn different disk related commands.

6.2 INTRODUCTION

In the previous unit, we have discussed the Linux File System and the related properties. The functions of different file systems in Linux are explained. We have also discussed the different types of files, I-node and the file attributes. In this unit, we will discuss about the file organisation. File is collection of information, and a file system is used to control how data is stored and retrieved. Without a file system, data placed in a storage medium would be a random assortment of some facts only, without having any specific beginning or ending for a particular data. Unix system treat everything inside it as files. Hence it has thousands of files. Unix file attributes

like name, size and access permission are stored separately rather than putting in the file like in Windows system.

6.3 FILES AND FILE SYSTEM

In general, we can categorize files into 3 categories:

- **Ordinary files:** These are regular files which contain stream of characters as data. This can be further subdivided it into Text file and Binary File
- **Directory files:** Unix system treat directories are as files. A directory file contains an entry for every file and subdirectory it has. Each entry consists of two entry the file and unique identification number (inode number) for the file or directory
- **Device files:** In Unix all devices and peripherals are also treat as files. To perform any operation on devices we must invoke associated with files. Device fil name are generally store under single directory structure called, /dev.

A filename in Unix can consist of up to 255 characters. Filename can be any ASCII character except the / and NULL character. But we should avoid using control characters in naming files.

All files in UNIX are related to one another and organized in hierarchical inverted tree structure. Every UNIX file system has a reference point at top of the tree structure which is called root and represented as / (forward slash). Under root we have a directory called home directory. Whenever we log on to UNIX system it automatically places us at home directory. for example, if you log into your system and open Unix Terminal (ctrl+Alt+T for Ubuntu users) and type pwd after the dollar symbol you will get a result like this:

/home/host_name

Here in the above, first slash "/" is synonym for root directory, inside the root directory we have home directory inside it we have our own directory here directory name is Ubuntu, Home directory and ubuntu directory is separated by the delimiter "/". The pwd (print working directory) command shows you the current working directory.



Fig 6.1: Snapshot of pwd command

We can use ls (listing command) to list out all the content present in the current directory. It will show a complete list of filenames in the current directory in ASCII collating sequence (number first, uppercase then lower case). It shows directories also.



Fig 6.2: Snapshot of ls command

At above picture it is showing contents of ubuntu (user directory) where all are directories except test0.txt which is text file

ls Options

ls command has different options which syntax is like ls -Options. We will present some important them.

Type ls -F in a terminal window to add "/" at the end of each directory. This command will help you to distinguish directories from files. See image below for further details:

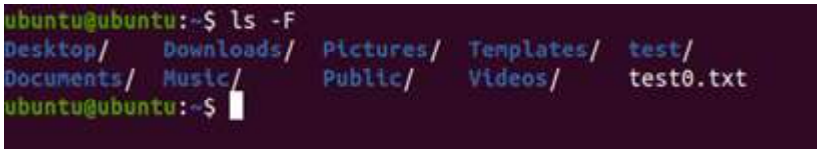


Fig 6.2: Snapshot of ls-F command

When using the basic ls command, you can't see hidden files and files starting with ".". But the ls -a command to display them as seen in the image below:



Fig 6.3: Snapshot of ls -a command

The get information about the user, size of the file, and date and time of modification. We can use ls -l command. In the picture bellow.

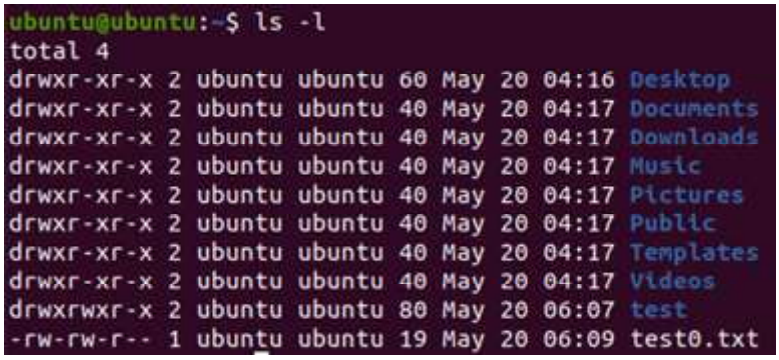


Fig 6.4: Snapshot of ls-l command

Let's us discuss this output little bit more elaborately. If we take the last line of the output, it looks like this:

-rw-rw-r-1 ubuntu ubuntu 19 May 20 06:09 test0.txt

1st Character - File Type: First character specifies the type of the file. In the example above the hyphen (-) in the 1st character indicates that this is a normal file. Following are the possible file type options in the 1st character of the ls -l output.

Field Explanation

- - normal file
- d : directory
- s : socket file
- l : link file
- **Field 1 - File Permissions:** Next 9 character specifies the files permission. The every 3 characters specifies read, write, execute permissions for user(root), group and others respectively in order. Taking above example, -rw-rw-r- indicates read-write permission for user(root) , read permission for group, and no permission for others respectively. If all three permissions are given to user(root), group and others, the format looks like -rwxrwxrwx
- **Field 2 - Number of links:** Second field specifies the number of links for that file. In this example, 1 indicates only one link to this file.
- **Field 3 - Owner:** Third field specifies owner of the file. In this example, this file is owned by username 'ubuntu'.
- **Field 4 - Group:** Fourth field specifies the group of the file. In this example, this file belongs to" ubuntu' group.

- **Field 5 - Size:** Fifth field specifies the size of file in bytes. In this example, '19' indicates the file size in bytes.
- **Field 6 - Last modified date and time:** Sixth field specifies the date and time of the last modification of the file. In this example, 'May 20 06:09' specifies the last modification time of the file.
- **Field 7 - File name:** The last field is the name of the file. In this example, the file name is test0.txt



CHECK YOUR PROGRESS

Q 1: What are the 3 types of file present in Linux?

.....

Q 2: Why ls command is used?

.....

6.4 WORKING WITH FILES

In this section we will try to explore different operations like creating, reading, searching, copying, moving, renaming, deleting, linking, and editing Files. First, we will go through operations of directory then we will go files as directory operations are little bit different.

Changing Directories

The pwd (print working directory) shows current directory we are in, and cd (change directory) is used switch your working directory from one to another. For more illustration look at the picture bellow. As per pwd command shows we are at first home directory "ubuntu", now we can use: cd pathname to move to different directory. Here we use cd test to move to directory test. We can use cd .. command to move out of the directory. After using this command, we can we have again come to "ubuntu" directory. Actually, the "." (dot) symbol refers to the working directory itself and ".." the (dot dot) symbol refers to the parent directory of the working directory

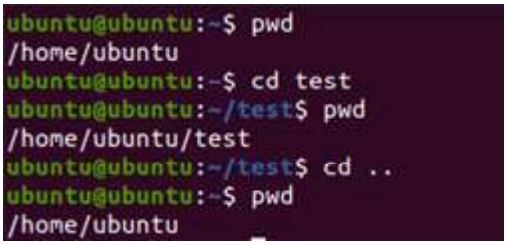


Fig 6.4: Snapshot of cd command

Creating and Deleting Directories

In order to create a directory, we use the mkdir (make directory) command, syntax for this command is mkdir directory_name. It will create a new directory in current directory. To remove a directory, we use the rmdir command. Remember that the rmdir command can only be used to remove empty directories. We can use rm (remove) command to delete directory, but it can also delete empty directories but with -rf option i.e.rm -rf directory_name We can force to delete the directory.



Fig 6.5: Snapshot of mkdir & rm -rf command

File Operations

we can create delete edit files using Linux GUI itself as like in windows. There is a general-purpose text editor for genome desktop called "gedit "with which we can create, edit and delete files. The general syntax is:

$gedit filename

For more details we can look into Linux manual i.e., $man gedit. Next, we have discussed some other means of file operations. All the operations have lots of different options, which can be used to do extra functionalities of the commands. But here we have only use straight forward utility of the commands only. We urged users to explore further possibilities with these commands

The touch command can be used to create a new file. It will create and open a new blank file if the file with a filename does not exist. And in

case the file already exists then the file will not be affected.

$touch filename

```
ubuntu@ubuntu:~$ ls
Desktop    Downloads  Pictures  Templates  test
Documents  Music      Public    Videos     test0.txt
ubuntu@ubuntu:~$ touch new_file
ubuntu@ubuntu:~$ ls
Desktop    Downloads  Pictures  Templates  new_file  test0.txt
Documents  Music      Public    Videos     test
```

Fig 6.6: Snapshot of touch command

In above picture with ls command, we can see there is no "new_file" in directory, but with touch command we have created the file.

cat command can be used to display the contents of a file. This command will display the contents of the 'filename' file. And if the output is very large then we could use more or less to fit the output on the terminal screen otherwise the content of the whole file is displayed at once.

Syntax for cat command is:

$cat [Option] filename

```
ubuntu@ubuntu:~$ cat new_file
this is the content of new_file .
```

Fig 6.7: Use of cat command

The cat is a very powerful tool user can look for its more uses. For example:

$cat > filename creates a new file

$cat filename1 filename2>filename3 joins two files (1 and 2) and stores the output of them in a new file (3)

Copying a File

cp command could be used to create the copy of a file. It will create the new file in destination with the same name and content as that of the file 'filename'. It remains in the same working directory. General syntax for cp is:

$cp [Option] source/filename destination/filename

First image is copy without path name and second one is copy with a pathname.

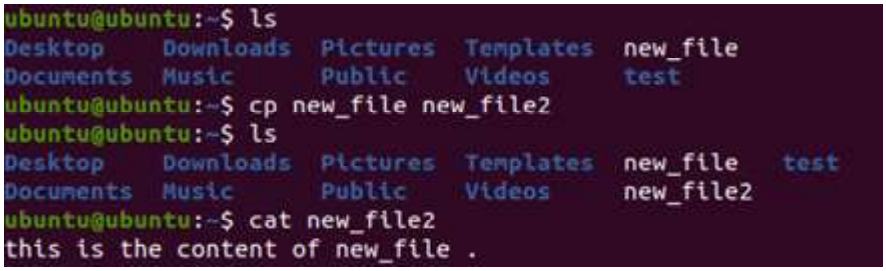


Fig 6.8: Snapshot of copy without path name

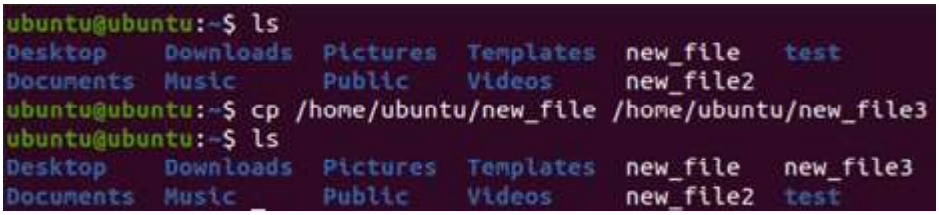


Fig 6.9: Snapshot of copy with path name

mv command could be used to move a file from source to destination. It will remove the file filename from the source folder and would be creating a file with the same name and content in the destination folder.

$mv [Option] source/filename destination/

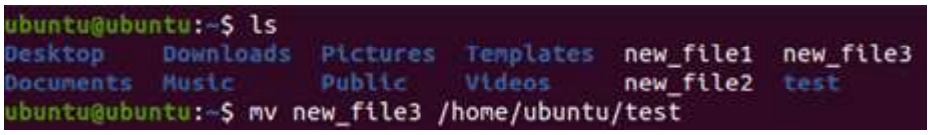


Fig 6.10: Snapshot of moving a file

In this screen shot we have move "new_file3" to "test" directory

rm command could be used to delete a file. It will remove the filename file from the directory. The general syntax for it is

$rm [Option] filename

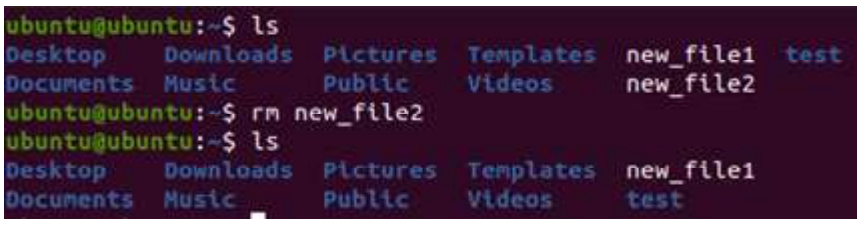


Fig 6.11: Snapshot of removing a file

The find command in UNIX is a command line utility for walking a file hierarchy. It can be used to find files and directories and perform subsequent operations on them. It supports searching by file, folder, name, creation date, modification date, owner and permissions. By using the '-

exec' other UNIX commands can be executed on files or folders found.

$ find /directory where to search [expression determines what to find] [-options] [what to find]



Fig 6.12: Snapshot of find command

Here we search new_file3 from the test directory using name option in find command.

Absolute Path vs Relative Path

A path is a unique location to a file or a folder in a file system of an OS. A path to a file is a combination of / and alpha-numeric characters. An absolute path is defined as the specifying the location of a file or directory from the root directory(/). In other words we can say absolute path is a complete path from start of actual filesystem from / directory. For example

/home/ubuntu/test

/home//boot/grub/grub.conf

all these paths started from / directory which is a root directory for every Linux/Unix machines

Relative path is defined as path related to the present working directory(pwd). Suppose at the moment we are at /home/ubuntu, we want to change it to /home/ubuntu/test, then we can directly change it t test using cd test command. If you observe there is no / before test which indicates it is a relative directory to present working directory. We can use an absolute path from any location where as if you want to use relative path we should be present in a directory where we are going to specify relative to that present working directory.

6.5 FILE SYSTEM MOUNTING AND UNMOUNTING

All files in a Linux filesystem are arranged in form of a big tree rooted at '/'.We will discussed this organization mor thoroughly later in this chapter .These files can be spread out on various devices based on partition table, initially parent directory is mounted(i.e attached) to this tree at '/',

others can be mounted manually using GUI interface(if available) or using mount command. The mount command is used to mount the filesystem found on a device to big tree structure (Linux filesystem) rooted at '/'. Conversely, another command umount can be used to detach these devices from the Tree.

Syntax:

mount -t type device dir

Other forms:

mount [-l|-h|-V]

mount -a [-fFnrsvw] [-t fstype] [-O optlist]

mount [-fnrsvw] [-o options] device|dir

mount [-fnrsvw] [-t fstype] [-o options] device dir

All these commands tell the Kernel to attach the filesystem found at device to the dir.

Note:

- If you leave the dir part of syntax it looks for a mount point in /etc/fstab.
- You can use -source or -target to avoid ambivalent interpretation. mount --target /mountpoint
- /etc/fstab usually contains information about which device needs to be mounted where.
- Most of the devices are indicated by files like /dev/sda4, etc. But it can be different for certain filesystems. For more information explore man mount

Some Important Options:

- l : Lists all the file systems mounted yet.
- h : Displays options for command.
- V : Displays the version information.
- a : Mounts all devices described at /etc/fstab.
- t : Type of filesystem device uses.
- T : Describes an alternative fstab file.
- r : Read-only mode mounted.

Example: **mount dev/sd4 /home/ubuntu/test**

Here we have mounted device/sd4 file system to test directory.

6.6 ORGANIZATION OF THE FILE TREE

The Linux File Hierarchy Structure or the Filesystem Hierarchy Standard (FHS) defines the directory structure and directory contents in Unix-like operating systems. It is maintained by the Linux Foundation. In the FHS, all files and directories appear under the root directory /, even if they are stored on different physical or virtual devices. So it looks like a tree organization where root '/'is the top directory it contain first level subdirectories and each first level sub directories contain second level subdirectories, hence the tree structure propagate. The picture below depicts how directories are organized in tree structure

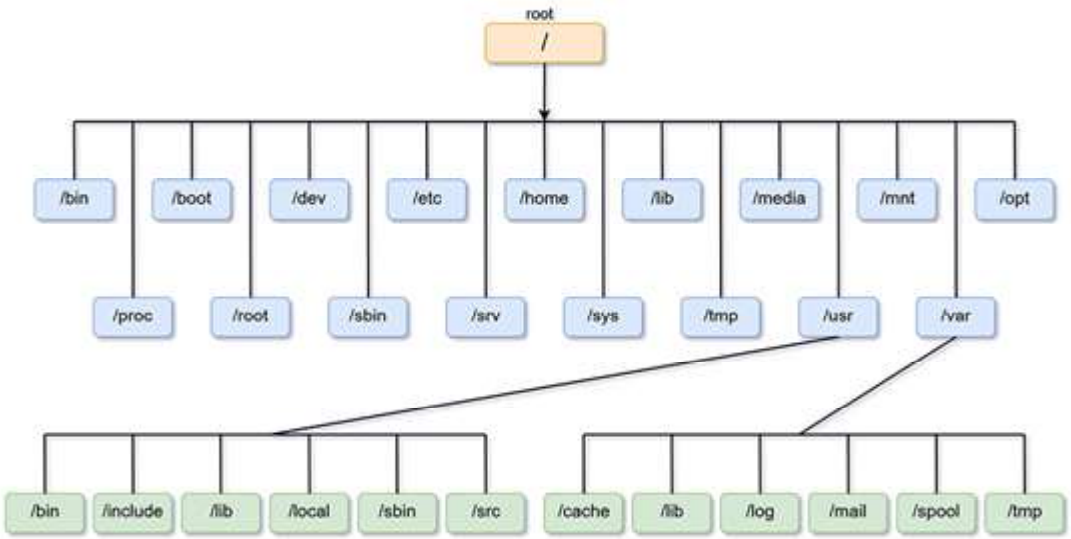


Fig 6.13: Linux File Structure Tree Organization

/ (Root): Primary hierarchy root and root directory of the entire file system hierarchy. Every single file and directory start from the root directory. The only root user has the right to write under this directory. the /root is the root user's home directory, which is not the same as /. Under root we have following subdirectories

- bin - This directory contains the binary or executable files that launch applications, all users have access to these applications. These represent the bare minimum set of programs required for a user to use the system.
- boot - This directory contains the files necessary to boot up your

computer including the boot loader and kernel activation files.

- dev - This directory contains the file pointers for character and block devices, which means the pointers to hard drives, flash drives, multimedia cards, external devices and system ports.
- etc - this directory contains the system configuration files and scripts.
- home - This directory contains the user profiles, desktops and user files, it functionality is similar to the "Documents and Settings" directory in MS-Windows.
- lib - This directory contains library files that are necessary for all programs to work.
- media - This directory is used by hal, the Linux auto-mounter, to load external devices for navigation such as floppy disks, cd-roms, dvd-roms, flash drives, etc…
- mnt - This directory is used to load external devices on system that do not use hal.
- opt - This directory contains optional software packages. Usually the contents of X11 and your window manger and loaded into opt, but you can add whatever programs you wish to this directory.
- proc - This directory is unique because it is not necessarily part of the file system, but a virtual file system that provides access to kernel or system information.
- root - This is the home directory for your root or system administrator. It is kept separate from the other users in case the partition that home is mounted on fails.
- sbin - This directory has a functionality similar to bin, which means it contains application executables, but rather than allowing all users access to these programs it restricts access only to the root or administrator.
- tmp - This is the temporary storage location. All users have read and write rights to the contents of this directory.
- usr - This is the largest directory on a Linux system. Pretty much everything that does not belong in the other directories is placed here. The contents usually include program, documentation and the kernel

source code.

- var - This directory contains most of the frequently changing files such as logging files, cache file and record locks.

If "tree" utility is installed is installed at your machine, we can check tree organization of file system inn UNIX/LINUX systems. It is a recursive directory listing program that produces a depth-indented listing of files. With no arguments, tree lists the files in the current directory. When directory arguments are given, tree lists all the files or directories found in the given directories each in turn. Upon completion of listing all files and directories found, tree returns the total number of files and directories listed. There are options to change the characters used in the output, and to use colour output. The most general way of using the tree command: $tree and the out will be like bellow.



Fig 6.14: Output of Tree Command



CHECK YOUR PROGRESS

Q 3: How to check tree structure of Linux?

.....

Q 4: What is the command for mounting file in Linux system

.....

6.7 DISK RELATED COMMANDS

There are several ways to list all the hard drives present in a system through Linux command lines. A hard drive could be physically connected, virtually connected, or even emulated. Here are some different commands which can list the hard drives, keep in mind there are others, but these are probably the most commonly used and easy to get the job done.

The du (Disk Usage) command shows the usage of disk. The du command will show you the disk usage for specific directories in Linux. The syntax for this command is :

$ du [optios] FILESYSTEM

When we use this command with no parameter, then this command will show you the information related to all the mounted file systems

$du



Fig 6.15: Snapshot of du command

Some other Functionality of du command

- **du -h:** - This will show you the disk usage in a human-readable format for each directory and subdirectory
- **du -a:** - This will show you the disk usage for all the files.
- **du -s:** - This will show you the total disk space used by a specific directory or file.

The df (Disk free) command will show you used and available disk spaces for in Linux system. The syntax for this command is:

$ df [options] FILESYSTEM

When we use this command with no parameter, then this command will show you the information related to all the mounted file systems:

$df

```
ubuntu@ubuntu:~$ df
Filesystem     1K-blocks    Used Available Use% Mounted on
udev              449040       0    449040   0% /dev
tmpfs             100232    1380     98852   2% /run
/dev/sr0         2999936 2999936         0 100% /cdrom
/dev/loop0       2109440 2109440         0 100% /rofs
/cow              501144  137112    364032  28% /
tmpfs             501144       0    501144   0% /dev/shm
tmpfs               5120       8      5112   1% /run/lock
tmpfs             501144       0    501144   0% /sys/fs/cgroup
tmpfs             501144       0    501144   0% /tmp
/dev/loop1         66688   66688         0 100% /snap/gtk-common-themes/1515
/dev/loop2         56832   56832         0 100% /snap/core18/2128
/dev/loop3         52224   52224         0 100% /snap/snap-store/547
/dev/loop4        224256  224256         0 100% /snap/gnome-3-34-1804/72
/dev/loop5         33152   33152         0 100% /snap/snapd/12704
tmpfs             100228      56    100172   1% /run/user/999
```

Fig 6.16: Snapshot output of df command

The following columns are contained in each line:

"Filesystem" - The filesystem's name.

"1K-blocks"- The filesystem's size in 1K blocks.

"Used"- Used means the used space in 1K blocks.

"Available" - Available means space available in 1K blocks.

"Use%"- The percentage of the used space.

"Mounted on"- The directory on which the filesystem is mounted/

Some options with df command

- 'df -h' option if we want to show information of all file system statistics in Gigabyte (GB).
- df-T option to tell df to display the file system types
- df -h If we want to show the details of disk drives in a human-readable format such as gigabytes, megabytes, kilobytes etc

fdisk can be used to manage disk partitions, you can create a new paritition, delete an existing partition…

the below let us see how to use fdisk command:

$ fdisk [options] FILESYSTEM

$fdisk -l

```
ubuntu@ubuntu:~$ sudo fdisk -l
Disk /dev/loop0: 2.1 GiB, 2160009216 bytes, 4218768 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/loop1: 65.1 MiB, 68259840 bytes, 133320 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

Fig 6.17: Snapshot of fdisk -l command



6.8 LET US SUM UP

- In Linux everything is treated as file and the file system is a hierarchical tree structure where root in the first node
- Command associated to work with file is ls, cd, rm, cp, mv
- Absolute path names are full path name to the file whereas relative pathnames are relative to the working directory
- For mounting a file system in Linux system, we use mount command and for unmounting we use unmount command
- df and du are two commands associated with Linux disk utility
- Leaner must go through to Linux manual to grasp the total functionality of these commands



6.9 FURTHER READINGS

1) Das, S. (2000).UNIX: Concepts and Applications. Tata McGraw-Hill.
2) Das, S. (2012). Your UNIX/Linux: The Ultimate Guide. (Third Edition). (Science/Engineering/Math). McGraw-Hill.
3) Christopher Negus (2012). Linux Bible. (8th Edition). Wiley.



6.10 ANSWERS TO CHECK YOUR PROGRESS

Ans to Q No 1: 3 types of file: Ordinary file, Directory File , Device File

Ans to Q No 2: For listing of files.

Ans to Q No 3: $tree command

Ans to Q No 4: $ mount command



6.11 MODEL QUESTIONS

Q 1: Describe tree organization of Linux file system.

Q 2: Create a test file in your home directory write something it display it on monitor then move the file to another directory.

Q 3: Explain how commands for directory is different than command used in case of file system.

Q 4: Explain the output of ls-l command in details.

Q 5: Why we need to mount a filesystem in Linux?

Q 6: Use df and du command to check disk utility.

*** ***** ***

UNIT 7: LINUX FILE HANDLING COMMANDS

UNIT STRUCTURE

7.1 Learning Objectives

7.2 Introduction

7.3 File and Directory Handling Commands

- 7.3.1 ls
- 7.3.2 cd
- 7.3.3 cp
- 7.3.4 mv
- 7.3.5 rm
- 7.3.6 mkdir
- 7.3.7 rmdir
- 7.3.8 pwd

7.4 Commands for Creating and Viewing Ordinary Files

- 7.4.1 cat
- 7.4.2 more
- 7.4.3 less

7.5 Let Us Sum Up

7.6 Further Readings

7.7 Answer to Check Your Progress

7.8 Model Questions

7.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- learn various file and directory handling commands
- understand commands for creating and viewing ordinary files

7.2 INTRODUCTION

In the previous unit, we have discussed the organisation of the Linux file structure. We have also discussed various disk related commands. In this unit, we will discuss various File and Directory Handling Commands

with examples. We will also discuss the commands for creating and viewing ordinary files in the subsequent sections.

7.3 FILE AND DIRECTORY HANDLING COMMANDS

As you get started out in Linux, you'll spend a great deal of time interacting with files and directories. In the following sections, we will discuss various file and directory management commands that are used mostly by a system administrator or a normal Linux user.

7.3.1 ls

The ls is the list command in Linux. It shows the full list or content of your directory.

```
localhost:~# ls
bench.py    hello.c     hello.js    readme.txt
localhost:~#
```

Various options can be used along with the ls command. Let us discuss the options below:

- **ls -a command**

In Linux, hidden files start with a dot (.) and can't be seen in the regular directory. The ls -a command displays the whole list of a directory including the hidden files also.

```
localhost:~# ls -a
.             .ash_history  .mozilla     bench.py     hello.js
..            .cache        .wine        hello.c      readme.txt
localhost:~#
```

- ls -l command

It displays the list in a long list format.

```
localhost:~# ls -l
total 16
-rw-r--r--    1 root     root           114 Jul  5  2020 bench.py
-rw-r--r--    1 root     root            76 Jul  3  2020 hello.c
-rw-r--r--    1 root     root            22 Jun 26  2020 hello.js
-rw-r--r--    1 root     root           151 Jul  5  2020 readme.txt
localhost:~#
```

- ls -lh command

Size of the file is very difficult to read when displayed in terms of bytes. This command displays the data in terms of Mb, Gb, Tb, etc.

```
localhost:~# ls -lh
total 16K
-rw-r--r--    1 root     root         114 Jul  5  2020 bench.py
-rw-r--r--    1 root     root          76 Jul  3  2020 hello.c
-rw-r--r--    1 root     root          22 Jun 26  2020 hello.js
-rw-r--r--    1 root     root         151 Jul  5  2020 readme.txt
localhost:~#
```

- ls -lhS command

This command displays the files in descending order (highest at the top) according to their size.

```
localhost:~# ls -lhS
total 16K
-rw-r--r--    1 root     root         151 Jul  5  2020 readme.txt
-rw-r--r--    1 root     root         114 Jul  5  2020 bench.py
-rw-r--r--    1 root     root          76 Jul  3  2020 hello.c
-rw-r--r--    1 root     root          22 Jun 26  2020 hello.js
localhost:~#
```

- ls -r

This command displays the list in reverse order.

```
localhost:~# ls -r
readme.txt  hello.js    hello.c     bench.py
localhost:~#
```

- ls -R

This command displays the subdirectories also.

```
localhost:~# ls -R
.:
bench.py    hello.c     hello.js    readme.txt
localhost:~#
```

- ls -lX

This command groups the files with same extensions together in the list.

```
localhost:~# ls -lX
total 16
-rw-r--r--    1 root     root          76 Jul  3  2020 hello.c
-rw-r--r--    1 root     root          22 Jun 26  2020 hello.js
-rw-r--r--    1 root     root         114 Jul  5  2020 bench.py
-rw-r--r--    1 root     root         151 Jul  5  2020 readme.txt
localhost:~#
```

- ls - lt

This command sort the list by displaying recently modified filed at

top.

```
localhost:~# ls -lt
total 16
-rw-r--r--    1 root     root           151 Jul  5  2020 readme.txt
-rw-r--r--    1 root     root           114 Jul  5  2020 bench.py
-rw-r--r--    1 root     root            76 Jul  3  2020 hello.c
-rw-r--r--    1 root     root            22 Jun 26  2020 hello.js
localhost:~#
```

- ls -~

It gives the contents of home directory.

```
localhost:~# ls ~
bench.py    hello.c     hello.js    readme.txt
localhost:~#
```

- ls ../

It displays the contents of parent directory.

```
localhost:~# ls ../
bin    etc    lib    mnt    proc   run    srv    tmp    var
dev    home   media  opt    root   sbin   sys    usr
localhost:~#
```

7.3.2 cd

This command is used to change the directory. For example, if we want to change the current directory to the directory KKHSOU2, we have to type cd KKHSOU2

- To navigate into the root directory, use "cd /"
- To navigate to your home directory, use "cd" or "cd ~"
- To navigate up one directory level, use "cd .."
- To navigate to the previous directory (or back), use "cd -"

7.3.3 cp

This command is used to copy a file or a directory.

For example, to copy the file hello.c to hello_new.c in the same directory, we have to type cp hello.c hello_new.c

```
localhost:~# cp hello.c hello_new.c
localhost:~# ls
KKHSOU2       KKHSOU_NEW    hello.c       hello_new.c
KKHSOU3       bench.py      hello.js      readme.txt
localhost:~#
```

To copy a file or a directory to a different location, we have to mention the path of the destination directory.

- To copy multiple file or directories in a directory, we have to type cp file1 file 2 directory name

  For example, to copy the files hello.c and hello_new.c to the directory KKHSOU_NEW, the following command can be used.

```
localhost:~# cp hello.c hello_new.c KKHSOU_NEW
```

- cp -r command is used to copy a directory along with its sub directories.

  For example, the directory KKHSOU_NEW now contains two files named hello.c and hello_new.c. Let us copy the directory KKHSOU_NEW with all the files to the directory KKHSOU2 which is at present empty.

```
localhost:~# cp -r KKHSOU_NEW KKHSOU2
```

Now, to check the content of the directory KKHSOU2, we have to change the directory using cd command.

```
localhost:~# cp -r KKHSOU_NEW KKHSOU2
localhost:~# cd KKHSOU2
localhost:~/KKHSOU2# ls
KKHSOU_NEW
localhost:~/KKHSOU2#
```

7.3.5 rm

rm stands for remove and this command is used to remove a file. We have to be very careful while using this command as it permanently removes a file.

rm filename

```
localhost:~# ls
bench.py          file.tar.gz       hello.js          student_marks.txt
file.tar          hello.c           readme.txt        test.txt
localhost:~# rm file.tar.gz
localhost:~# ls
bench.py          hello.c           readme.txt        test.txt
file.tar          hello.js          student_marks.txt
localhost:~#
```

In the above example, you can see that we have removed/ deleted the file 'file.tar.gz' using the rm command.

- rm -r

With rm '-r' option, you can delete a directory having sub directories inside it. So you don't need to delete sub-directories manually.
The syntax is: rm -r directoryname

- rm -i

The option 'i' stands for 'interactivity'. Before deleting a file it will ask you whether you want to delete it or not. The syntax is:

rm -i filename

Similarly, to delete a directory 'interactively', the following command is used:

rm -ri directoryname

7.3.6 mkdir

To create a new directory, mkdir command is used. The mkdir stands for 'make directory'. For example, to create a new directory named KKHSOU, you have to type mkdir KKHSOU

```
localhost:~# mkdir KKHSOU
localhost:~#
```

The directory KKHSOU is created. To check whether the directory is created or not, type the ls command.

```
localhost:~# ls
KKHSOU      bench.py    hello.c     hello.js    readme.txt
localhost:~#
```

We can create multiple directories with a single command. Suppose, we want to create directories named KKHSOU1, KKHSOU2 and KKHSOU3

So, we have to type mkdir KKHSOU1 KKHSOU2 KKHSOU3

```
localhost:~# mkdir KKHSOU1 KKHSOU2 KKHSOU3
localhost:~# ls
KKHSOU      KKHSOU2     bench.py    hello.js
KKHSOU1     KKHSOU3     hello.c     readme.txt
localhost:~#
```

Note: By default, your directory will be created in your current

directory. If you want to create your directory somewhere else, then provide the path of your destination directory and your file will be created there.

7.3.7 rmdir

To delete a directory, rmdir command is used. But it can delete an empty directory only. If the directory contains sub-directories, it will not be able to delete the directory.

For example, to delete the directory KKHSOU, we have to type rmdir KKHSOU

```
localhost:~# rmdir KKHSOU
localhost:~# ls
KKHSOU1     KKHSOU3     hello.c     readme.txt
KKHSOU2     bench.py    hello.js
localhost:~#
```

7.3.8 pwd

This command displays the current working directory. It displays the whole path starting from the root ending to the directory.

```
localhost:~/KKHSOU2# pwd
/root/KKHSOU2
localhost:~/KKHSOU2#
```



CHECK YOUR PROGRESS

Q 1: _____ command displays the whole list of a directory including the hidden files.

Q 2: To display the whole list of content of a directory including the sub-directories also _____ command is used.

Q 3: To display the content of the parent directory, _____ command is used.

Q 4: _____ command is used to copy a directory along with its sub directories.

Q 5: _____ command displays the current working directory.

7.4 COMMANDS FOR CREATING AND VIEWING ORDINARY FILES

7.4.1 cat

The 'cat' command is one of the most frequently used commands in Linux. Let us discuss the various uses of the cat command.

- The 'cat' command can be used to display the content of a file. For example, to display the content of the file hello.c, we have to type

  cat hello.c

```
localhost:~# cat hello.c
#include <stdio.h>
int main(void)
{
        printf("hello world\n");
        return 0;
}

localhost:~#
```

To display the content of multiple files at once, type file names in one single line like "cat file1 file2 file3... filen".

For example, if we want to display the content of the files named bench.py and hello.c, we have to type the command as

cat bench.py hello.c

```
localhost:~# cat bench.py hello.c
def fib(n):
    if n < 2:
        return n
    return fib(n-2) + fib(n-1)

n = 25
print("fib(", n, ")= ", fib(n))
#include <stdio.h>
int main(void)
{
        printf("hello world\n");
        return 0;
}

localhost:~#
```

- The 'cat' command can be used to create a new file with greater

than sign (>).

For example, to create a new file named "file1.txt", we have to type the command

cat > file1.txt and press 'enter'. You will be directed to the next line. Press 'enter' after every line and you will be directed to the next line. To save your file, go to the next line, press 'ctrl+d' and your file will be saved.

```
localhost:~# cat > file1.txt
```

- The 'cat' command with double greater than sign (>>) append (add something in the last of a file) something in your already existing file.

```
localhost:~# cat >> file1.txt
This is line no1.
Welcome to KKHSOU.
```

Here, we have added two lines in the already existing file file1.txt. In this case, since the file1.txt is an empty file, the two lines will be appended. After typing the lines, press ctrl + d to save the file.

- The 'cat' command can be used to copy the content of a file into another file.

  For example, to copy the file readme.txt to another file named readme_copy.txt, we have to write the following command:

  cat readme.txt>readme_copy.txt

```
localhost:~# cat readme.txt > readme_copy.txt
localhost:~# cat readme_copy.txt
Some tests:

- Compile hello.c with gcc (or tcc):

  gcc hello.c -o hello
  ./hello

- Run QuickJS:

  qjs hello.js

- Run python:

  python3 bench.py
localhost:~#
```

- The 'cat' command can be used to concatenate the contents of multiple files in a single new file.

```
localhost:~# cat file1.txt hello.c > new_file.txt
```

In the above example, we have concatenate the files file1.txt and hello.c to a new file new_file.txt

Now, after concatenation, to view the content of the file new_file.txt, we have to write

cat > new_file.txt

```
localhost:~# cat new_file.txt
This is line no1.
Welcome to KKHSOU.

#include <stdio.h>
int main(void)
{
        printf("hello world\n");
        return 0;
}

localhost:~#
```

- The 'cat -n' option displays line numbers in front of each line in a file.

```
localhost:~# cat -n hello.c
     1  #include <stdio.h>
     2  int main(void)
     3  {
     4          printf("hello world\n");
     5          return 0;
     6  }
     7
localhost:~#
```

- The 'cat -b' option removes the empty lines.

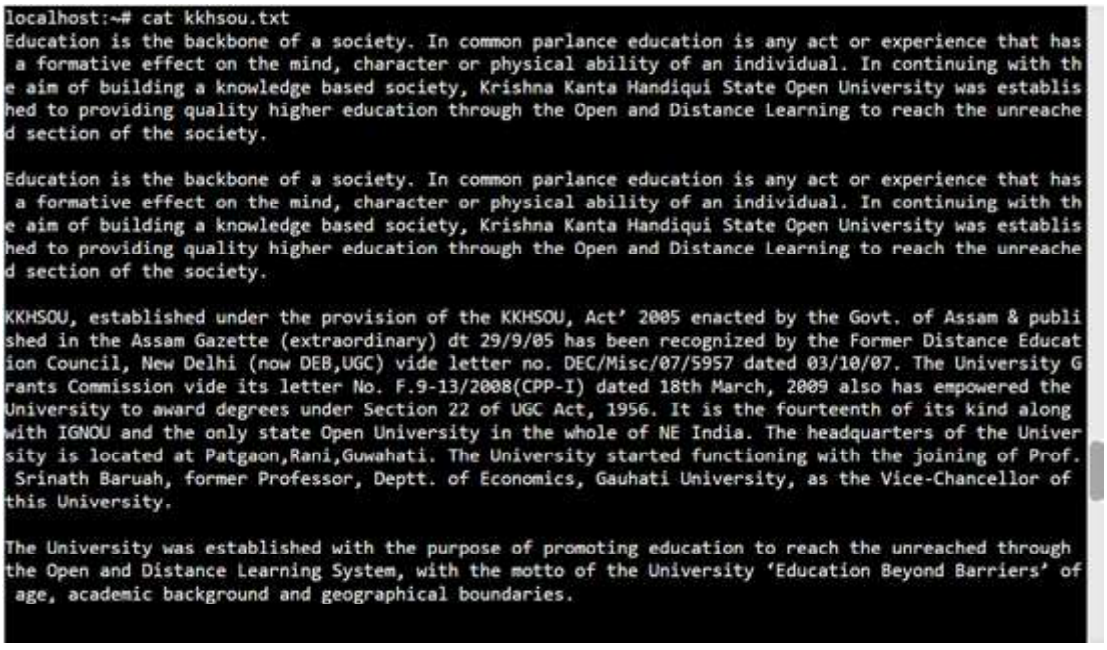
```
localhost:~# cat -b hello.c
     1  #include <stdio.h>
     2  int main(void)
     3  {
     4          printf("hello world\n");
     5          return 0;
     6  }

localhost:~#
```

7.4.2 more

We have already learnt that 'cat' command can be used to display the content of a file. Similarly we can also use the 'more' command to display the content of a file. Now, the question may arise that what are the differences between the 'cat' and 'more' command while displaying the content of a file? The answer is that, in case of larger files, 'cat' command output will scroll off your screen while 'more' command displays output one screenful at a time.

For example, suppose we have a file named 'kkhsou.txt' which contains some long texts. Now, to display the content of the file 'kkhsou.txt', we have to type the command cat kkhsou.txt as shown below:



You have seen that, we have to scroll to read the content of the file. Let us use the 'more' command to display the content of the same file 'kkhsou.txt' and see the differences.

- **more -num:** Limits the line displayed per page.(num is the total number of lines to be displayed)

```
localhost:~# more -10 kkhsou.txt
Education is the backbone of a society. In common parlance education is any act or experience that has
 a formative effect on the mind, character or physical ability of an individual. In continuing with th
e aim of building a knowledge based society, Krishna Kanta Handiqui State Open University was establis
hed to providing quality higher education through the Open and Distance Learning to reach the unreache
d section of the society.

Education is the backbone of a society. In common parlance education is any act or experience that has
 a formative effect on the mind, character or physical ability of an individual. In continuing with th
e aim of building a knowledge based society, Krishna Kanta Handiqui State Open University was establis
hed to providing quality higher education through the Open and Distance Learning to reach the unreache
localhost:~#
```

It displays the first 10 lines of the file 'kkhsou.txt'.

7.4.3 less

The 'less' command is same as 'more' command but include some more features.

It automatically adjust with the width and height of the teminal window, while 'more' command cuts the content as the width of the terminal window get shorter.

For example, less kkhsou.txt

```
KKHSOU, established under the provision of the KKHSOU, Act' 2005 enacted by the Govt. of Assam & pub
lished in the Assam Gazette (extraordinary) dt 29/9/05 has been recognized by the Former Distance Educ
ation Council, New Delhi (now DEB,UGC) vide letter no. DEC/Misc/07/5957 dated 03/10/07. The University
 Grants Commission vide its letter No. F.9-13/2008(CPP-I) dated 18th March, 2009 also has empowered th
e University to award degrees under Section 22 of UGC Act, 1956. It is the fourteenth of its kind alon
g with IGNOU and the only state Open University in the whole of NE India. The headquarters of the Univ
ersity is located at Patgaon,Rani,Guwahati. The University started functioning with the joining of Pro
f. Srinath Baruah, former Professor, Deptt. of Economics, Gauhati University, as the Vice-Chancellor o
f this University.

The University was established with the purpose of promoting education to reach the unreached through
the Open and Distance Learning System, with the motto of the University 'Education Beyond Barriers
 of age, academic background and geographical boundaries.

The main aim of the university is to develop and provide easily accessible modes of quality higher edu
cation and training with the use of latest educational inputs and technology. Because of the inherent
flexibility in terms of pace and place of learning, methods of evaluation, etc., The University holds
the promise of providing equality of opportunities for higher education and bringing into its fold the
 deprived and denied sections along with the fresh learners.

The programmes of the University are so designed that it can spread education to all the learners of t
he region.The course structures of the university have been designed at par with the national curricul
um. Along with the traditional programmes, the University offers various professional Certificate and
Diploma programmes to enable the already enrolled students in various colleges and educational institu
tions of the state to pursue add-on studies simultaneously with their regular courses.

:
```



7.5 LET US SUM UP

- ls command shows the full list or content of your directory.
- ls -l command displays the list in reverse order.
- ls -lX command groups the files with same extensions together in the

list.

- cp command is used to copy a file or a directory.
- cp -r command is used to copy a directory along with its sub directories.
- To rename a directory mv command is used.
- rm command permanently removes a file.
- You can delete a directory having sub directories inside it with the rm -r command.
- To create a new directory, mkdir command is used.



7.6 FURTHER READINGS

1) Das, S. (2000).UNIX: Concepts and Applications. Tata McGraw-Hill.
2) Das, S. (2012). Your UNIX/Linux: The Ultimate Guide. (Third Edition). (Science/Engineering/Math). McGraw-Hill.
3) Christopher Negus (2012). Linux Bible. (8th Edition). Wiley.
4) https://www.javatpoint.com/linux-tutorial



7.7 ANSWERS TO CHECK YOUR PROGRESS

Ans to Q No 1: ls -a

Ans to Q No 2: ls -R

Ans to Q No 3: ls ../

Ans to Q No 4: cp -r

Ans to Q No 5: pwd



7.8 MODEL QUESTIONS

Q 1: Create a directory named EMPLOYEE and DEPARTMENT.

Q 2: Copy a file to the directory EMPLOYEE.

Q 3: Rename the directory DEPARTMENT to DEPT.

Q 4: Change your current directory to a different directory and then display your current working directory.

*** ***** ***

UNIT 8: LINUX FILTER COMMANDS

UNIT STRUCTURE

8.1 Learning Objectives

8.2 Introduction

8.3 Filter Commands

- 8.3.1 wc
- 8.3.2 head
- 8.3.3 tail
- 8.3.4 cut
- 8.3.5 tr
- 8.3.6 top
- 8.3.7 grep

8.4 File Compression and System Monitoring Commands

- 8.4.1 tar
- 8.4.2 unzip
- 8.4.3 kill
- 8.4.4 netstat

8.5 File Access Permission Commands

- 8.5.1 chmod
- 8.5.2 chown
- 8.5.3 chgrp

8.6 Let Us Sum Up

8.7 Further Readings

8.8 Answer to Check Your Progress

8.9 Model Questions

8.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- understand the use of wc command with various options
- learn head, tail, cut, tr and top commands
- search words in a file using the grep command

- understand to compress and decompress files
- work with various file access permission commands like chmod, chown and chgrp

8.2 INTRODUCTION

In the previous unit, we have discussed various file handling commands with examples. We have also discussed various commands for Creating and Viewing ordinary files. In this unit, we will discuss various filter commands and commands for creating and viewing ordinary files. In the last part of this unit, we will discuss various file access permission commands also.

8.3 FILTER COMMANDS

When a program takes its input from another program, it performs some operation on that input, and writes the result to the standard output. It is referred to as a filter. A large numbers of filter commands are available in Linux. In the next sections, we will discuss these commands with examples.

8.3.1 wc Command

The wc command displays the number of lines, number of characters, and the number of words in a file.

Example: To display the number of lines, words, and bytes from a file.

wc <filename>

```
localhost:~# wc kkhsou.txt
      31      933     6227 kkhsou.txt
localhost:~#
```

Example: Display the complete count information of multiple files.

wc <file1><file2>

```
localhost:~# wc hello.c kkhsou.txt
       7       10       76 hello.c
      31      933     6227 kkhsou.txt
      38      943     6303 total
localhost:~#
```

The above command displays the number of lines, words, and the number of the bytes from the files 'hello.c' and 'kkhsou.txt'. It also displays the total number of lines, words, and the number of the bytes of the two files.

Example: Display the number of lines in a file.

wc - l filename

```
localhost:~# wc -l kkhsou.txt
31 kkhsou.txt
localhost:~#
```

Example: Display the number of characters in a file

wc -m kkhsou.txt

```
localhost:~# wc -m kkhsou.txt
6215 kkhsou.txt
localhost:~#
```

Example: Display the number of bytes in a file

```
localhost:~# wc -c kkhsou.txt
6227 kkhsou.txt
localhost:~#
```

Example: Display the number of words in a file

wc -w kkhsou.txt

```
localhost:~# wc -w kkhsou.txt
933 kkhsou.txt
localhost:~#
```

Example: Display the total number of files and folders in a directory

ls | wc -l

```
localhost:~# ls | wc -l
8
localhost:~#
```

Example: Display the length of the longest line

wc -L kkhsou.txt

```
localhost:~# wc -L kkhsou.txt
941 kkhsou.txt
localhost:~#
```

8.3.2 head Command

The 'head' command displays the starting content of a file. By default, it displays starting 10 lines of any file.

head <filename>

```
localhost:~# head hello.c
#include <stdio.h>
int main(void)
{
        printf("hello world\n");
        return 0;
}

localhost:~#
```

Example: Display the first 10 lines of multiple files

head <file1><file2>

```
localhost:~# head hello.c readme.txt
==> hello.c <==
#include <stdio.h>
int main(void)
{
        printf("hello world\n");
        return 0;
}


==> readme.txt <==
Some tests:

- Compile hello.c with gcc (or tcc):

  gcc hello.c -o hello
  ./hello

- Run QuickJS:

  qjs hello.js
localhost:~#
```

In the above screenshot, we have seen that the lines are displayed from two files named 'hello.c' and 'readme.txt' with separate headings. (Since, the two files contains less than 10 lines, it displays the lines less than 10)

Example: Display specific number of lines from a file.

head -n filename

```
localhost:~# head -3 hello.c
#include <stdio.h>
int main(void)
{
localhost:~#
```

Example: Display specific number of bytes of a file.

head -c <number><filename>

```
localhost:~# head -c 220 kkhsou.txt
Education is the backbone of a society. In common parlance education is any act or experience that has
 a formative effect on the mind, character or physical ability of an individual. In continuing with th
e aim of buildinlocalhost:~#
```

In the above example, we have displayed the 220 bytes of content from the file 'kkhsou.txt'.

8.3.3 tail Command

Linux tail command is used to display the last ten lines of one or more files.

tail <filename>

```
localhost:~# tail test.txt
11
12
13
14
15
16
17
18
19
20
localhost:~#
```

Example: Display specific number of lines from last.

tail -n <number><file name>

```
localhost:~# tail -5 test.txt
16
17
18
19
20
localhost:~#
```

Example: Display specified number of bytes from the last.

tail -c <number><file name>

```
localhost:~# tail -10 test.txt
11
12
13
14
15
16
17
18
19
20
localhost:~#
```

From the above output, the last six bytes of the file 'test.txt' are displayed.

Example: Display the first 10 lines of multiple files

tail <file1><file2>

```
localhost:~# tail hello.c test.txt
==> hello.c <==
#include <stdio.h>
int main(void)
{
        printf("hello world\n");
        return 0;
}


==> test.txt <==
11
12
13
14
15
16
17
18
19
20
localhost:~#
```

8.3.4 cut Command

It is used to cut a specific section by byte position, character, and field and writes them to standard output. To cut a specific section, it is necessary to specify the delimiter and it can be a space (' '), a

hyphen (-), a slash (/), or anything else. After '-f' option, the column number is to be mentioned.

- To cut by using the hyphen (-) as the delimiter, execute the below command:

  cut -d- -f(columnNumber) <fileName>

  Let us explain with an example.

```
localhost:~# cat >> student_marks.txt
Rajib-90
Pranjit-67
Pradip-45
Pritam-78
Prakash-89
localhost:~# cut -d- -f2 student_marks.txt
90
67
45
78
89
localhost:~# cut -d- -f1 student_marks.txt
Rajib
Pranjit
Pradip
Pritam
Prakash
localhost:~#
```

Here, we have created a new file 'student_marks' and added the content of the marks obtained by students using 'cat' command. Then, we have displayed only the marks using the 'cut' command. In the next section, we have displayed the name of the students only.

- The '-b' option is used to cut a section of line by byte. To cut a file by its byte position, execute the command as follows:

  cut -b <byte number> <file name>

```
localhost:~# cat student_marks.txt
Rajib-90
Pranjit-67
Pradip-45
Pritam-78
Prakash-89
localhost:~# cut -b 5 student_marks.txt
b
j
i
a
a
localhost:~#
```

8.3.5 tr Command

The command 'tr' stands for 'translate'. It is used to translate, like from lowercase to uppercase and vice versa.

command | tr <'old'> <'new'>

```
localhost:~# cat student_marks.txt
Rajib-90
Pranjit-67
Pradip-45
Pritam-78
Prakash-89
localhost:~# cat student_marks.txt | tr 'rai' 'RAI'
RAjIb-90
PRAnjIt-67
PRAdIp-45
PRItAm-78
PRAkAsh-89
localhost:~#
```

8.3.6 top Command

It is used to display all the running process within the environment of a system. It helps in monitoring system usage and performances. It is mainly used to detect load on the server by system administrators.

Type the following command:

top

```
Mem: 5868K used, 181376K free, 8K shrd, 0K buff, 872K cached
CPU:   0% usr   0% sys   0% nic  96% idle   0% io   0% irq   1% sirq
Load average: 0.00 0.00 0.00 1/32 151
  PID  PPID USER     STAT   VSZ %VSZ CPU %CPU COMMAND
  151    61 root     R     1516   1%   0   1% top
   61     1 root     S     1552   1%   0   0% sh -l
   97    61 root     T     1524   1%   0   0% cat
    1     0 root     S     1512   1%   0   0% {init} /bin/sh /sbin/init
   56     1 root     S     1260   1%   0   0% dhcpcd -q
   55     1 root     S      744   0%   0   0% settime -d /
    7     2 root     SW       0   0%   0   0% [ksoftirqd/0]
   15     2 root     SW       0   0%   0   0% [kworker/0:1]
   62     2 root     SW       0   0%   0   0% [kworker/u2:1]
   10     2 root     SW<      0   0%   0   0% [writeback]
   11     2 root     SW       0   0%   0   0% [kcompactd0]
   12     2 root     SW<      0   0%   0   0% [crypto]
   13     2 root     SW<      0   0%   0   0% [bioset]
   14     2 root     SW<      0   0%   0   0% [kblockd]
    6     2 root     SW<      0   0%   0   0% [mm_percpu_wq]
   16     2 root     SW       0   0%   0   0% [kswapd0]
   17     2 root     SW<      0   0%   0   0% [bioset]
   34     2 root     SW       0   0%   0   0% [khvcd]
   35     2 root     SW<      0   0%   0   0% [bioset]
   36     2 root     SW<      0   0%   0   0% [bioset]
   37     2 root     SW<      0   0%   0   0% [bioset]
   38     2 root     SW<      0   0%   0   0% [bioset]
   39     2 root     SW<      0   0%   0   0% [bioset]
   40     2 root     SW<      0   0%   0   0% [bioset]
    8     2 root     SW       0   0%   0   0% [kdevtmpfs]
    4     2 root     SW<      0   0%   0   0% [kworker/0:0H]
```

8.3.7 grep Command

The 'grep' command stands for "global regular expression print". It filters the content of a file which makes our search easy.

- The 'grep' command is generally used with pipe (|).

Syntax: command | grep <searchWord>

Example: cat student_marks.txt | grep 9

```
localhost:~# cat student_marks.txt
Rajib-90
Pranjit-67
Pradip-45
Pritam-78
Prakash-89
localhost:~# cat student_marks.txt | grep 9
Rajib-90
Prakash-89
localhost:~#
```

- The 'grep' command can also be used without the pipe (|).

Syntax: grep <searchWord> <file name>

Example: grep 9 student_marks.txt

```
localhost:~# cat student_marks.txt
Rajib-90
Pranjit-67
Pradip-45
Pritam-78
Prakash-89
localhost:~# grep 9 student_marks.txt
Rajib-90
Prakash-89
localhost:~#
```



CHECK YOUR PROGRESS

Q 1: The syntax to display the complete count information of multiple files is _____

Q 2: _____ command displays the number of lines in a file.

Q 3: wc -w kkhsou.txt command displays the number of _____ of the file 'kkhsou.txt'

Q 4: _____ command can be used to display the starting 10 lines of any file.

Q 5: _____ command is used to translate from lowercase to uppercase and vice versa.

Q 6: To search word in a file, _____ command is used.

8.4 FILE COMPRESSION AND SYSTEM MONITORING COMMANDS

We often need to compress files and directories to save space without compromising the quality. In Linux, few handy commands are available to achieve the task. In the next sections, we will discuss those commands. We will also discuss how to work with some system related commands like kill and netstat in the next sections.

8.4.1 tar Command

The tar command stands for tape archive. This command is used for creating Archive and extracting the archive files.

- Making an uncompressed tar archive with -cvf option

  The command is:

  tar cvf file.tar *.txt

```
localhost:~# tar cvf file.tar *.txt
readme.txt
student_marks.txt
test.txt
localhost:~#
```

The above example makes a tar file known as file.tar. It is the archive of every .txt file inside the current directory.

- Extracting files through the archive with -xvf option

  The command is:

  tar xvf file.tar

```
localhost:~# tar xvf file.tar
readme.txt
student_marks.txt
test.txt
localhost:~#
```

- gzip compression over tar archive with -z option

This option makes a tar file known as file.tar.gz. It is the archive of every .txt file.

The command is as follows:

tar cvzf file.tar.gz *.txt

```
localhost:~# tar cvzf file.tar.gz *.txt
readme.txt
student_marks.txt
test.txt
localhost:~# ls
bench.py          file.tar.gz       hello.js          student_marks.txt
file.tar          hello.c           readme.txt        test.txt
localhost:~#
```

- Extracting the gzip tar archive with -xvzf option

This option can extract the files through file.tar.gz tar archived files.

The command is as follows:

tar xvzf file.tar.gz

```
localhost:~# tar xvzf file.tar.gz
readme.txt
student_marks.txt
test.txt
localhost:~# ls
bench.py          file.tar.gz       hello.js          student_marks.txt
file.tar          hello.c           readme.txt        test.txt
localhost:~#
```

8.4.2 unzip Command

To compress a file, the zip command can be used as follows:

zip myfile.zip filename.txt

To unzip the file, the following command is used:

unzip myfile.zip

8.4.3 kill Command

It is used for manually terminating the processes. The command is:

type -a kill

The kill command transfers a signal towards a process in which further terminates the process. We can use the kill command along with the -l option for getting the list of every available signal:

kill -l



8.4.4 netstat Command

netstat command stands for Network statistics. It displays information about different interface statistics, including open sockets, routing tables, and connection information.

The basic syntax of the netstat command is as follows:

netstat

- Installation of the netstat command

If the netstat command is not installed on your machine, it will display the traditional Linux installation error message "Command 'netstat' not found."

To install it, execute the below command:

sudo apt install net-tools

The above command will ask for the administration password to install the command.

netstat -a

The above command will list all the existing connections.

```
localhost:~# netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
netstat: /proc/net/tcp6: No such file or directory
udp        0      0 0.0.0.0:bootpc          0.0.0.0:*
netstat: /proc/net/udp6: No such file or directory
netstat: /proc/net/raw6: No such file or directory
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node Path
unix  2      [ ACC ]     STREAM     LISTENING       2930 /run/dhcpcd.sock
unix  2      [ ACC ]     STREAM     LISTENING       2932 /run/dhcpcd.unpriv.sock
unix  2      [ ]         DGRAM                      2961
localhost:~#
```

8.5 FILE ACCESS PERMISSION COMMANDS

In linux, three popular file access permission commands are used. They are (a) chmod (b) chown (c) chgrp. In the next sections, we will discuss how they can be used.

8.5.1 chmod Command

Linux chmod command is used to change the access permissions of files and directories. chmod stands for change mode. In the Linux, each file is associated with a particular owner and have permission access for different users. The user classes may be:

- owner
- group member
- Others (Everybody else)

The file permissions in Linux are the following three types:

- read (r)
- write (w)
- execute (x)

Let's see how to change the file permission using the chmod command.

The basic syntax of chmod command is as follows:

chmod <options> <permissions> <file name>

To view the permission of all text file from the current working directory, execute the below command:

```
localhost:~# ls -l *.txt
-rw-r--r--    1 root     root          151 Jul  5  2020 readme.txt
-rw-r--r--    1 root     root           51 Oct  8 13:54 student_marks.txt
-rw-r--r--    1 root     root           51 Oct  8 13:09 test.txt
localhost:~#
```

To set the read and write permission for other users, execute the below command:

chmod o+w *.txt

It will set the read and write permission for other users of the text files. Consider the below output:

```
localhost:~# ls -l *.txt
-rw-r--r--    1 root     root          151 Jul  5  2020 readme.txt
-rw-r--r--    1 root     root           51 Oct  8 13:54 student_marks.txt
-rw-r--r--    1 root     root           51 Oct  8 13:09 test.txt
localhost:~# chmod o+w *.txt
localhost:~# ls -l *.txt
-rw-r--rw-    1 root     root          151 Jul  5  2020 readme.txt
-rw-r--rw-    1 root     root           51 Oct  8 13:54 student_marks.txt
-rw-r--rw-    1 root     root           51 Oct  8 13:09 test.txt
```

Numerical Shorthand

We can use the numeric values instead of letters to specify the permissions. A three-digit value is used to specify the permission. The leftmost digit represents the owner (u), and the middle digit represents the group members (g). The rightmost digit represents the others (o).

The following table represents the digits and their permissions:

Digits	**Permissions**
000	No permission
001	Execute permission
010	Write permission
011	Write and execute permissions
100	Read permission
101	Read and execute permissions
110	Read and write permissions
111	Read, write, and execute permissions

8.5.2 chown Command

The chown stands for change owner. chown command is used to change a file's ownership, directory, or symbolic link for a user or group.

- Display the UID, GID, and Groups

To display all the groups, execute the groups command as follows:

groups

To display the UID and GID, execute the id command as follows:

id

```
localhost:~# id
uid=0(root) gid=0(root)
localhost:~#
```

- Display the User and Group Ownership of a File

To display the user and group owner of a file, execute the "ls -l" command with a specific file name. Consider the below command:

ls -l hello.c

The above command will display the user and group ownership of the 'hello.c' Consider the below output:

```
localhost:~# ls -l hello.c
-rw-r--r--    1 root     root           76 Jul  3  2020 hello.c
localhost:~#
```

- Change the owner of a File (Using user name)

To change the owner of a file, pass the user name (new owner) with the chown command as follows:

sudo chown <username> <File name>

- Change the Owner of a File (Using UID)

We can change the owner of a file or directory using the UID of a user. To change the ownership by using the UID, pass the UID with chown command as follows:

sudo chown <userid> <File name>

- Change the Group of a File

To change the group of a file, specify group name or GID

with the chown command. Execute the chown command specifying the group name separated by a colon as follows:

sudo chown :<groupname> <filename>

or

sudo chown :<GID> <filename>

8.5.3 chgrp Command

chgrp command is used to change the group ownership of a file or directory.

- Change the Group Ownership of a File

To change the group ownership of a file, execute the command as follows:

sudo chgrp groupname filename

- Change the group ownership of a folder

To change the group ownership of a folder, execute the 'chgrp' command with the folder name as follows:

sudo chgrp groupname directoryname



8.6 LET US SUM UP

- The wc command displays the number of lines, number of characters, and the number of words in a file.
- The 'head' command displays the starting content of a file. By default, it displays starting 10 lines of any file.
- tail command is used to display the last ten lines of one or more files.
- Cut command is used to cut a specific section by byte position, character, and field and writes them to standard output.
- tr command is used to translate, like from lowercase to uppercase and vice versa.
- Top command is used to display all the running process within the environment of a system.
- grep command is used to search a keyword in a file.
- tar command is used for creating Archive and extracting the archive

files.

- kill command is used for manually terminating the processes.
- Netstat command displays information about different interface statistics, including open sockets, routing tables, and connection information.
- chmod command is used to change the access permissions of files and directories.
- chown command is used to change a file's ownership, directory, or symbolic link for a user or group.
- chgrp command is used to change the group ownership of a file or directory.



8.7 FURTHER READINGS

1) Das, S. (2000).UNIX: Concepts and Applications. Tata McGraw-Hill.
2) Das, S. (2012). Your UNIX/Linux: The Ultimate Guide. (Third Edition). (Science/Engineering/Math). McGraw-Hill.
3) Christopher Negus (2012). Linux Bible. (8th Edition). Wiley.
4) https://www.javatpoint.com/linux-tutorial



8.8 ANSWERS TO CHECK YOUR PROGRESS

Ans to Q No 1: wc <file1><file2>

Ans to Q No 2: wc - l filename

Ans to Q No 3: word

Ans to Q No 4: head <filename>

Ans to Q No 5: tr

Ans to Q No 6: grep



8.9 MODEL QUESTIONS

Q 1: Explain the Linux wc command with various options.

Q 2: Differentiate between head and tail command with examples.

Q 3: What are the functions of cut command?

Q 4: Explain the function of (a) tr and (b) top commands .

Q 5: Explain the grep command with examples.

Q 6: How will you compress and un-compress files using tar and untar commands? Explain with examples.

Q 7: How will you compress and un-compress files using zip and unzip commands? Explain with examples.

Q 8: For what purpose kill command is used?

Q 9: Explain the netstat command.

Q 10: Explain the chmod command with examples.

Q 11: Explain the chown command with examples.

Q 12: Explain the chgrp command with examples.

*** ***** ***