LINUX - SHELL SCRIPTING (BASH SHELL)

WEEK 1	WEEK 2		WEEK 3	
Module 1 - Introduction	**Module 2 & Module 3**		**Module 4 & Module 5**	
Introduction to Linux	Unix Vs Linux	List Files & Folders	Disk Utilization	Types of Shell Scripting
Purpose of scripting	Features of Linux	Change directory	Vi/Vim Editors	Introduction with Bash Shell Script
Purpose of Automation	OS Installation	Copy Files & Folders	File Content Creation	Purpose of Notation
Types of Operating System	Administrator Users & Groups	Move Files & Folders	Insert command	Types of Variables
Linux OS Flavours	Help System	Resource Check	Echo command	Special Variables
System Architecture	File System Creation & Permission	CPU Utilization	File permissions	Basic Operators
Understanding File system hierarchy	Create a file	RAM/Memory Utilization	Chmod command	Decision Making
The Commands	Create a directory	Difference b/w Touch & Vim	History of commands	Loop Types

WEEK 4	WEEK 5	WEEK 6	WEEK 7	WEEK 8
Module 6 & Module 7		**Module 8 & Module 9**		**Module 10**
Arrays	Functions	User Management	Benefits of Logging	Remote System Management
Control Substitution	Types of functions	Group Management	Tools Integrations	Shell Scripting in DevOps
Quoting Mechanisms	Regular Expressions	Linux Administration	API Management using Script	Scripting in Orchestration Tools
Environmental Variables	SED/AWK/CURL command	System Performance	Database Management	Workflow Management Support
Input/Output Redirections	Installing & Managing Software	Process Management	Capacity Management	Scripting in Configuration Mgnt
Basic Scripting with Examples	Package Management	Log Management	Exception Handling in Script	Comparison with other scripting
Scheduling Linux tasks - Cron jobs				

WEEK 9		WEEK 10	WEEK 11	WEEK 12
Module 11 & Module 12		**Module 13 & Module 14**		**Module 15**
Introducing Ansible	Deploying Ansible	Ad-hoc commands	Exception Handling in Play	Ansible Vault
DevOps - Life Cycle	Configuration Files Overview	Help System	Conditional Statement	Ansible Roles
Introduction to Configuration Mgmt	Hosts & Inventory Management	Managing Variables & Facts	Looping Statement	Troubleshooting Ansible
Configuration Mgnt Tools	Ansible Modules	Implementing Playbooks	Notify & Handlers	Automation Use Cases
Ansible Automation Platform	Ansible Facts	Implementing Task Control	Ansible Tags	Ansible Tower Overview
AWX vs Ansible Tower	Connectivity Requirement	Deploying Files to Managed hosts	API Management in Play	Integration with Github
Ansible Architecture	Managing Hosts	Handling Multiple Tasks	File System Handling	Advanced Concepts & Interview Questions

DAY 01

Introduction

Unix Vs Linux

Types of Operating System

Linux OS Flavors

System Architecture

Purpose of Scripting

Purpose of Automation

Understanding File System Hierarchy

The Commands

INTRODUCTION

TYPES OF OPERATING SYSTEM

WHY LINUX IS BEST

“Linux is basically a simple operating system, but you have to be a genius to understand its simplicity”.

- Linux is a critical technology for IT professionals to understand.
- Linux is in widespread use, worldwide. Internet users interact with Linux applications and web server systems daily, by browsing the World Wide Web and using e-commerce sites to buy and sell products.
- Linux is in use for much more than the internet. Linux manages point-of-sale systems and the world's stock markets, powers smart TVs and in-flight entertainment systems, and runs most of the top 500 supercomputers in the world.
- Linux provides the core technologies that power the cloud revolution and the tools to build the latest generations of container-based microservices applications, software-based storage technologies, and big data solutions.
- In the modern data center, Linux and Microsoft Windows are the predominant operating systems. Linux use continues to expand in enterprise, cloud, and device spaces. Due to its widespread adoption, you have many reasons to learn Linux:
- A Windows user needs to interoperate with Linux systems and applications.
- In application development, Linux commonly hosts the application and its runtime.
- In cloud computing, both private and public cloud instances use Linux as the operating system.
- Mobile applications and Internet of Things (IoT) devices commonly run on Linux.
- When looking for new IT career opportunities, Linux skills are in high demand.

SUMMARY

Linux is open-source software.

Being open-source means that you can see all how a program or system works. You can also experiment with changes and share them freely for others to use. The open-source model means that improvements are easier to make, enabling faster innovation.

Linux provides a command-line interface (CLI) for easy access and powerful scripting.

Linux is built around a basic design philosophy that users can perform all administration tasks from the CLI. It enables easier automation, deployment, and provisioning, and simplifies both local and remote system administration. Unlike many other operating systems, these capabilities were in the architecture from the start, and result in ease of use and stability.

Linux is a modular operating system that is designed to easily replace or remove components.

System components can be upgraded and updated when needed. A Linux system can be a general-purpose development workstation or a purposefully minimized software appliance.

Differences	Linux	Unix
Origins	Linux was developed in the 1990s by Linus Torvalds as a free and open-source alternative to Unix.	Unix was developed in the 1970s at Bell Labs
Introduction	Linux is Open Source, and many programmers work together online and contribute to its development.	Unix was developed by AT&T Labs, different commercial vendors, and non-profit organizations.
Licensing	Linux, on the other hand, is open-source software and can be used freely without any licensing fees.	Unix is a proprietary operating system, meaning that it requires a license to use.
Kernels	both have a similar design but are less complex than the Unix hold-up that kernel.	both have a similar design but larger and more complex than the Linux kernel.
Availability	On the other hand, Linux is widely used on both enterprise and personal computers.	Unix is typically found on enterprise-level servers and workstations and is less commonly used on personal computers.
Community Support	Linux has a large and active community of developers and users who contribute to its development and provide support.	While Unix also has a community, it is generally smaller and more focused on enterprise-level users.
Accessibility	It is an open-source operating system which is freely accessible to everyone.	It is an operating system which can only be utilized by its copywriters.

CONT.,

Bug fixing time	Threat recognition and solution is very fast because Linux is mainly community driven. So, if any Linux client poses any sort of threat, a team of qualified developers starts working to resolve this threat.	Unix clients require longer hold up time, to get the best possible bug-fixing, and a patch.
Graphical User Interface	Linux provides two GUIs, KDE and GNOME. But there are many other options. For example, LXDE, Xfce, Unity, Mate, and so on.	Initially, Unix was a command-based OS, however later a GUI was created called Common Desktop Environment. Most distributions now ship with Gnome.
Use Cases	It is used everywhere from servers, PCs, smartphones, tablets to mainframes.	It is used on servers, workstations, and PCs.
Shell Compatibility	The default interface is **BASH** (Bourne Again Shell).	It initially used Bourne shell.
Source Code Availability	The source is accessible to the general public.	The source is not accessible
Operating System Versions	Some Linux versions are **Ubuntu, Debian GNU, Arch Linux**, etc.	Some Unix versions are **Sun OS, Solaris, SCO UNIX, AIX, HP/UX, ULTRIX**, etc.

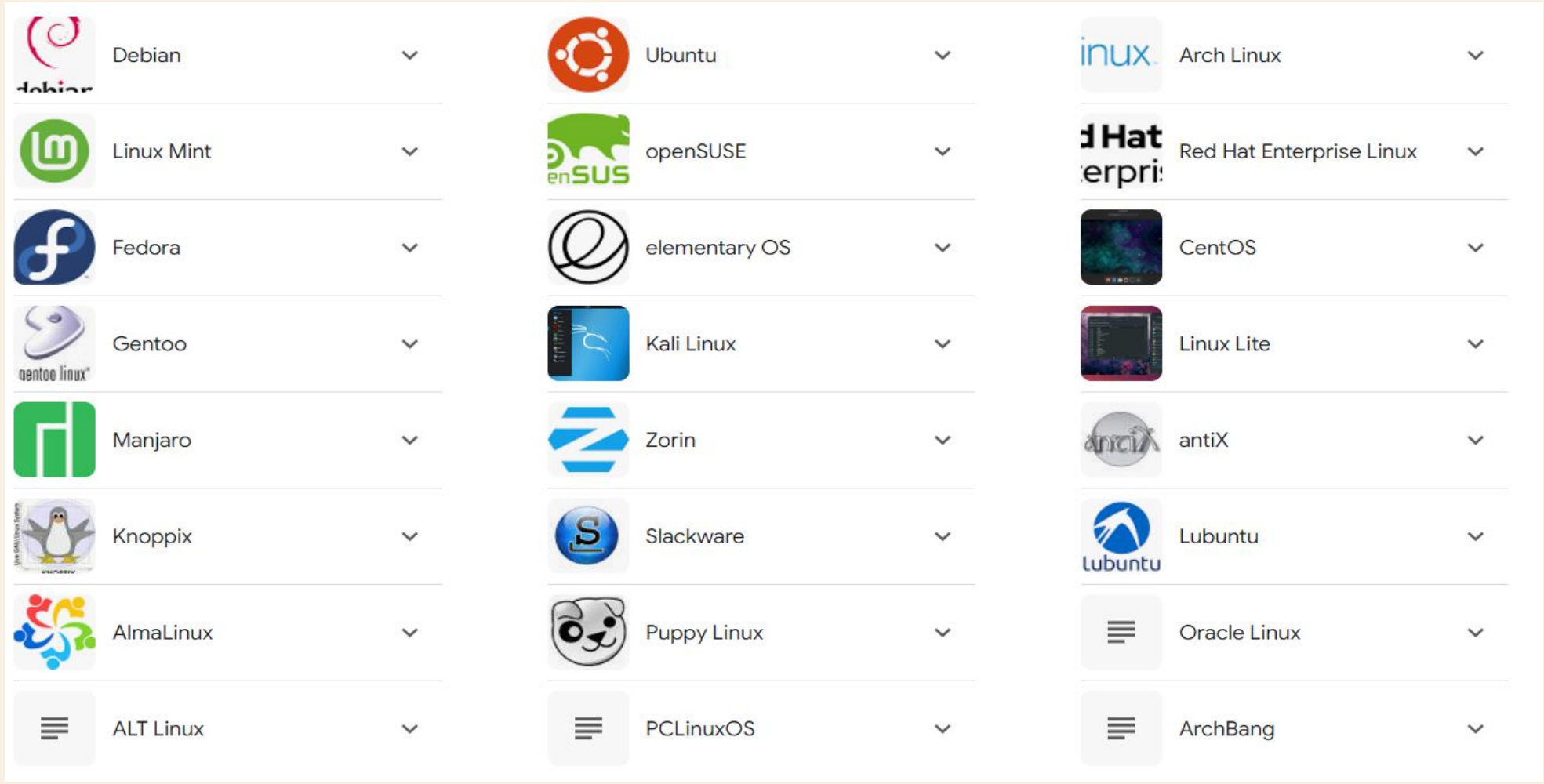
LINUX OS FLAVORS

Unix:

Linux
HP UX
IBM AIX
MacOS
Solaris

Linux:

Redhat
CentOS
Ubuntu
Fedora
SUSE



FEATURES OF LINUX

Linux offers both CLI (Command Line Interface) and GUI (Graphical User Interface) to its users. Unix has certain standard features that makes it unique and special. Some of them are listed below:

Multi-user: Multiple users can login and use the system and share the resources in a Unix machine without compromising the security of one user from the other at the same time.

Multi-tasking: It can handle multiple tasks at once while sharing the system resources fairly among them. Along with managing the active process, it can also effectively handle multiple background processes as well.

Pipes: It allows you to connect multiple small programs and create a complex program using pipes

Hierarchical File System: It stores information in a hierarchical file system. This makes it easier to maintain files and folders and other information in the system effectively. We will learn more about the Unix file system in the upcoming resources.

Security: A valid username and password is needed to access a Unix/Linux system.

Utilities: Users can perform various functions in a Unix machines instantly by using the utilities provided by Linux. Create a file using ‘touch’, print a statement using ‘echo’ and what not.

Portability: Linux is written in a high-level machine language which makes it easily portable.

Redirections: Using the redirection features in Linux, one can modify the standard input, output, and error streams of Unix/Linux as per the need.

Software development tools: It supports almost all software development tools.

SYSTEM ARCHITECTURE

- An operating system is the underlying software that acts as an interface between the user and the computer resources.
- An operating system has its own hardware and software components and a well-designed architecture.
- The architecture of an operating system is the overall design of its hardware and software components. Unix OS has a layered architecture.

- The components of the architecture are:

Kernel: Kernel is the resource manager of the operating system. It is responsible for sharing system resources between multiple processes that are running in the system.

Shell: Shell is the interface between the user and the kernel. Unix offers many different types of shells like bourne shell, bourne again shell, C shell, T shell, K Shell, etc.,

Utilities: Users communicate to the kernel through the shell using the utilities. Unix provides many in-built utilities, and it is easier for users to create a utility by combining the existing ones.

INTRODUCTION TO THE BASH SHELL

A command line is a text-based interface that is used to input instructions to a computer system.

The Linux command line is provided by a program called the shell. Many shell program variants have been developed over the years.

The default user shell in Red Hat Enterprise Linux (RHEL) is the GNU Bourne-Again Shell (bash). The bash shell is an improved version of the original Bourne Shell (sh) on UNIX systems.

The shell displays a string when it is waiting for user input, called the shell prompt. When a regular user starts a shell, the prompt includes an ending dollar ($) character:

[user@host ~]$
A hash (#) character replaces the dollar ($) character when the shell is running as the superuser, root. This character indicates that it is a superuser shell, which helps to avoid mistakes that can affect the whole system.

[root@host ~]#
Using bash to execute commands can be powerful. The bash shell provides a scripting language that can support task automation. The shell has capabilities that can enable or simplify operations that are hard to accomplish at scale with graphical tools.

TAKEAWAY

The bash shell is conceptually similar to the Microsoft Windows cmd.exe command-line interpreter. However, bash has a sophisticated scripting language, and is more similar to Windows PowerShell.

On macOS, the bash shell was the default shell before macOS 10.15 Catalina. Starting from macOS 10.15 Catalina, Apple changed the default shell to the zsh shell.

SHELL BASICS

Commands that are entered at the shell prompt have three basic parts:

Command to run.

Options to adjust the behavior of the command.

Arguments, which are typically targets of the command.

The command is the name of the program to run. It might be followed by one or more options, which adjust the behavior of the command or what it does. Options normally start with one or two dashes (-a or --all, for example) to distinguish them from arguments. Commands might also be followed by one or more arguments, which often indicate a target that the command should operate on.

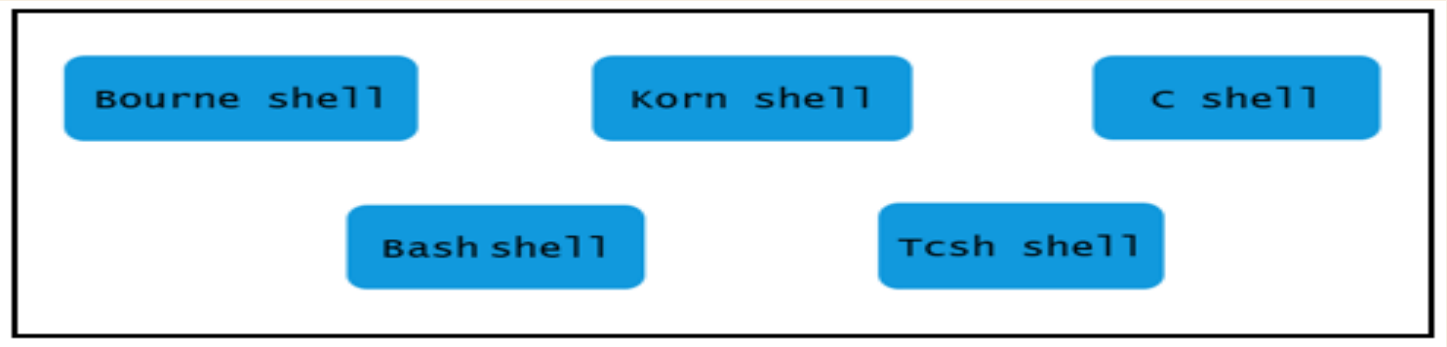
For example, in the **usermod -L user01**

usermod is the command, **-L** is the option, and **user01** is the argument.

This command locks the password of the user01 user account.

PURPOSE OF SCRIPTING

- A **shell** acts as an interface between the user and the operating system to perform the commands typed by the user. It is a command language interpreter that executes the command read from file or keyboard.
- Below are some of the different types of shells that can be used in a Linux environment. Note that the default shell for a Linux user is bash.



- A **shell script** is a computer program which is run by one of the Unix/Linux shells. These programs are group of commands, constructs and various other concepts involved in shell scripting. Usually, these computer programs are written to automate the daily mundane tasks carried out by a System Administrator to tell about the more basic use case

PURPOSE OF AUTOMATION

- **IT infrastructure** are the components required to operate and manage enterprise IT environments. These components include hardware, software, networking components, operating systems (OSs), and data storage—all of which are used to deliver IT services and solutions. IT infrastructure products are available as downloadable software applications that run on top of existing IT resources—like software-defined storage—or as online solutions offered by service providers—like Infrastructure-as-a-Service (IaaS).
- The benefits of automated operations are **higher productivity, reliability, availability, increased performance, and reduced operating costs**
- IT automation is the process of creating software and systems to replace repeatable processes and reduce manual intervention.
- It accelerates the delivery of IT infrastructure and applications by automating manual processes that previously required a human touch.
- **Automation** is the use of technology to perform tasks with reduced human assistance. Automation helps you accelerate processes and scale environments, as well as build continuous integration, continuous delivery, and continuous deployment (CI/CD) workflows. There are many kinds of automation, including IT automation, business automation, robotic process automation, industrial automation, artificial intelligence, machine learning, and deep learning.

DAY 1 - END