Linux Administration workshop

Bridging the Mediterranean

Introduction

- Fundamental of Linux, including the file system structure, user and group management, and permission settings.
- Manage software packages, monitor system performance through logging, and securely access remote systems.
- Network configurations, including essential services and file-sharing protocols.
- Automation techniques using Bash scripting, Python programming, and configuration management with Ansible.
- Special Supportive Tools for Developments like Git, PXE server, terminal multiplexers and Readthedocs.
- Containerization, Orchestration and Kubernetes.

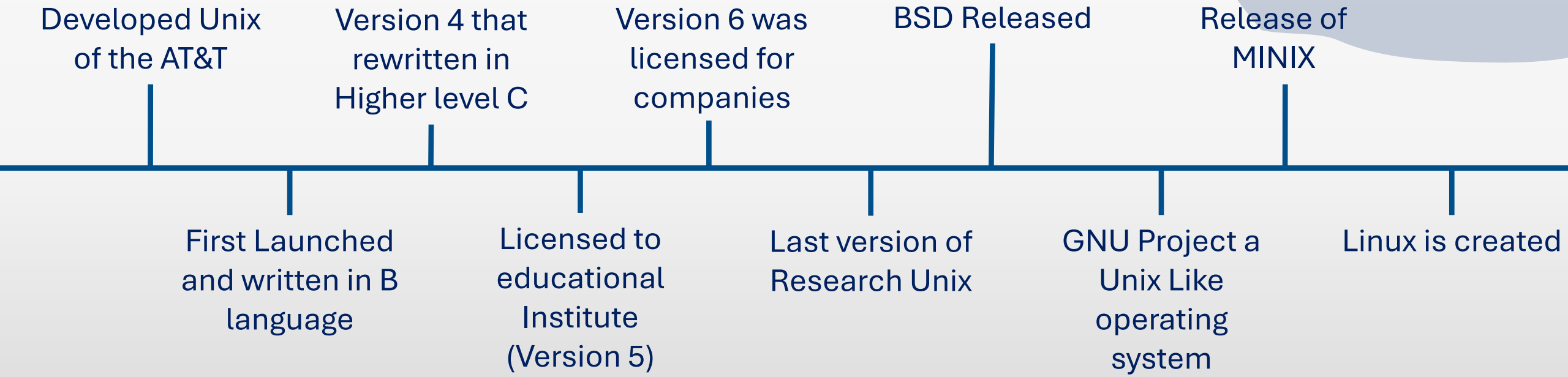
Outcomes

- Understand Linux architecture, including file systems, processes, networking, and system administration tasks, such as user management, permission settings, and software installation.
- Build, deploy, and manage containerized applications using Docker.
- Utilize Kubernetes for managing complex applications.
- Develop scripts to automate repetitive tasks and manage Linux systems efficiently.

Install the VMs

- Install VMware Workstation Player.
- Install the client and server VMs.
- Access the VM (the password is: sesame)

History of Linux



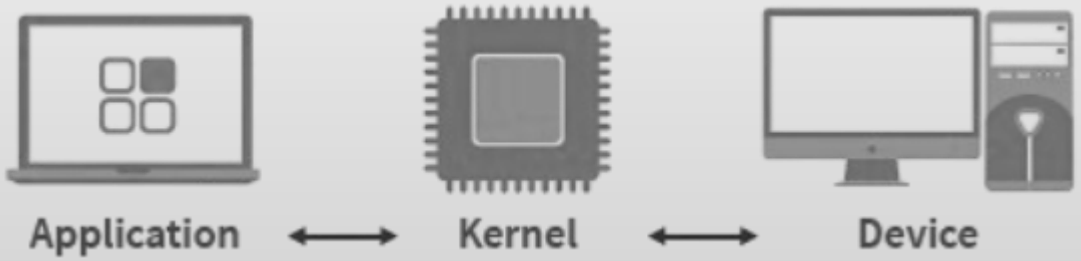
https://frontpagelinux.com/articles/guide-through-history-of-unix-linux-everything-you-need-to-know/

What is a Kernel?

It is a computer program that is the core of a computer's operating system, with complete control over everything in the system.

It manages the following :

- File management
- Process management
- I/O management
- Memory management
- Device management etc.



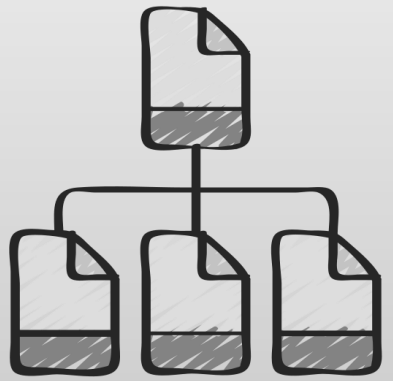
What is a Shell?

- It is a special user program that provides an interface for the user to use operating system services.
- It accepts human-readable commands from users and converts them into something which the kernel can understand.
- It is a command language interpreter that executes commands read from input devices such as keyboards or from files.

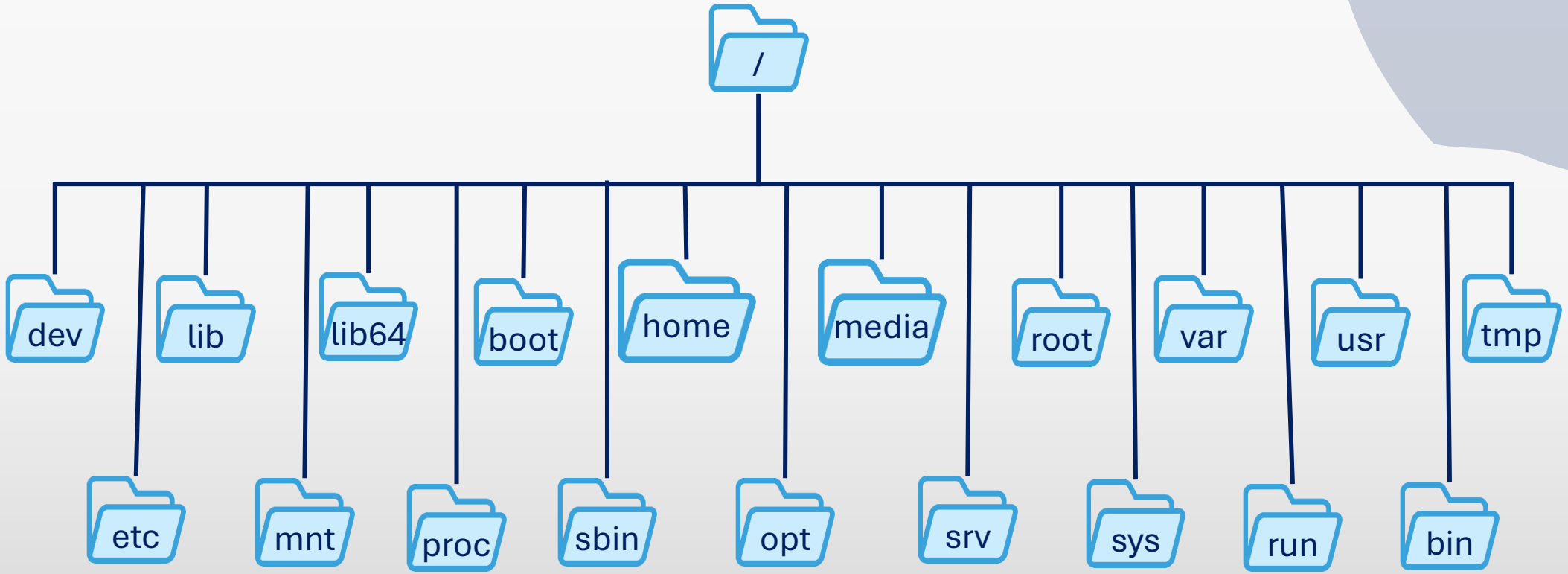
Linux Boot Process Explained

https://www.freecodecamp.org/news/linux-boot-process-in-rhel/#heading-grub-grand-unified-boot-loader

File Systems and Directory Structure



Linux File Hierarchy Structure



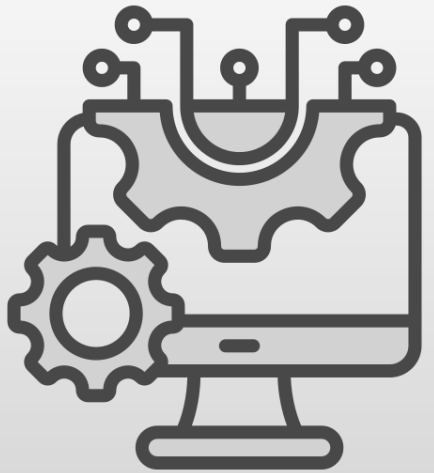
Linux File Hierarchy Structure

Directory	Purpose
/bin/	Essential User Command Binaries
/boot/	Static Files of the boot loader
/dev/	Device Files
/etc/	Host specific system configuration
/home/	User home Directories
/lib/	Shared Libraries
/media/	Removable Madia
/mnt/	Mounted Filesystem
/opt/	Add-on Application software package
/sbin/	System Binaries
/srv/	Data for service from system
/tmp/	Temporary Files
/usr/	User Utilities and Applications
/proc/	Process Information

Basic Linux Commands

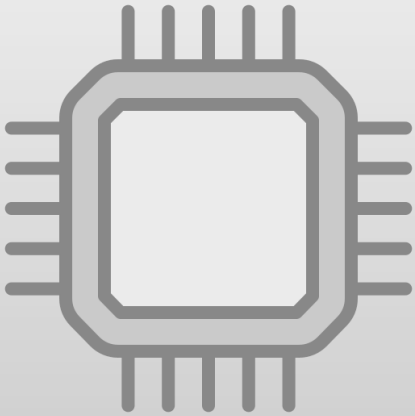
System Information Commands

Command	Description
hostname	Show system host name
last reboot	Show system reboot history
w	Display who is online
whoami	Who you are logged in as
date	Show the current date and time
cal	Show this month's calendar
uname -a	Display Linux system information



Hardware Information Commands

Command	Description
cat /proc/cpuinfo	Display CPU information
cat /proc/meminfo	Display memory information
free -h	Display free and used memory (-h for human readable)
lspci -tv	Display PCI devices
lsusb -tv	Display USB devices
hdparm -i /dev/sda	Show info about disk sda



Performance Monitoring and Statistics Commands

Command	Description
top	Display and manage the top processes
htop	Interactive process viewer
mpstat 1	Display processor related statistics
vmstat 1	Display virtual memory statistics
iostat 1	Display I/O statistics
tail –number file	Display the last lines in the file.
tcpdump -i interface	Capture all packets on the interface eth0
lsof	List all open files on the system

File and Directory Commands

Command	Description
ls	List the directory contents
pwd	Display the current working directory
touch filename	Create a new empty file, or update the last modified timestamp
vim filename	Create a new empty file
cat filename	Display the content of the file
cp source_file Destination_file	Copying the file
rm filename	Remove file

File and Directory Commands

Command	Description
mkdir directory	Create a directory
rmdir directory	Remove a directory
mv source destination	Move or rename files and directories
more filename	Browse through a text file
less filename	Browse through a text file
head filename	Output the beginning (or top) portion of file
tail filename	Output the ending (or bottom) portion of file

Directory Navigation Commands

Command	Description
cd ..	Go up one level of the directory tree. (Change into the parent directory.)
cd	Go to the home directory
cd path/directory	Change to the path/directory

User and Group Management

User and Group Management

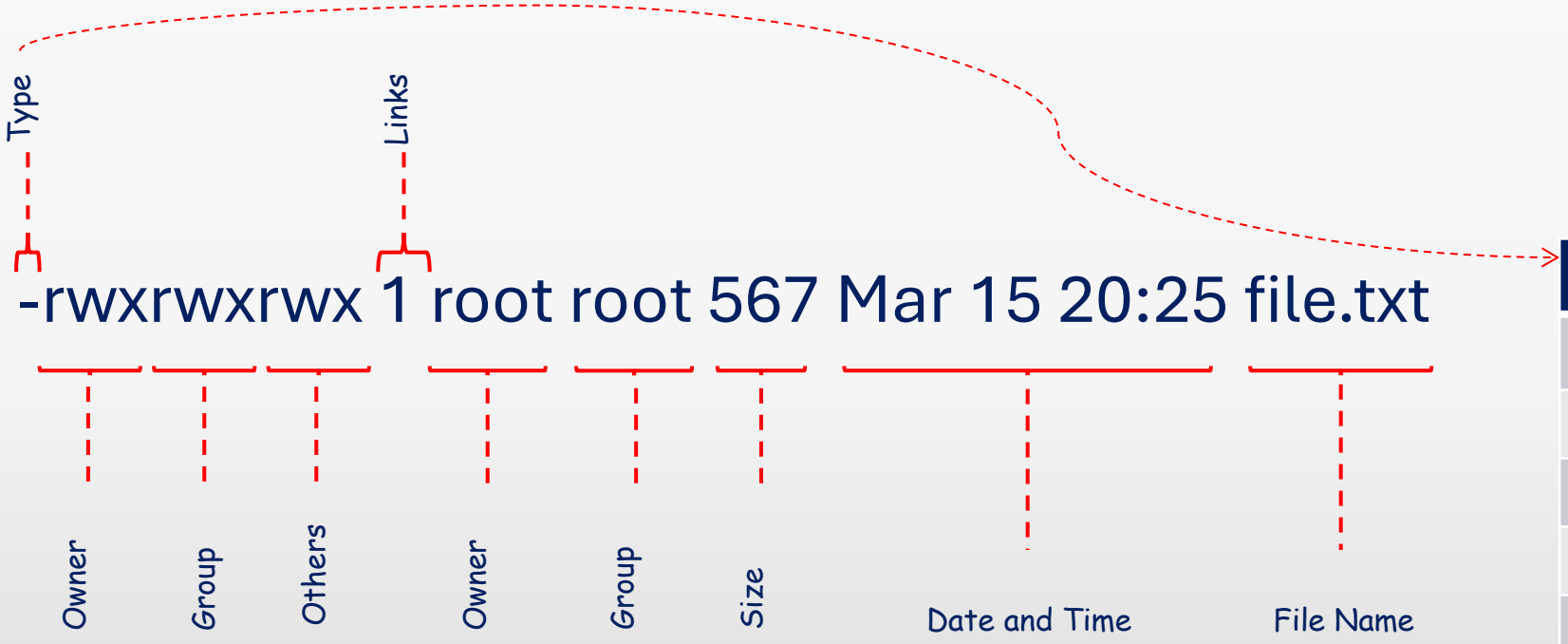
- A user can belong to several groups
- A file can belong to only one user and one group at a time
- The superuser “root” has extra privileges
- The password information in **/etc/shadow**, the group information in **/etc/group** and the user information in **/etc/passwd**.
- The user who own the file can change the ownership of it.

Users and Groups Management Commands

Command	Description
id username	Know more about the user
adduser username	Adding a new user to the system
su – username	Access the system with the new user
usermod –aG groupname username	Add username to a new group
userdel username	Remove username
groupadd groupname	Add a new group
groupadd –g groupid groupname	Add a new group with specific GID
groupdel groupname	Remove a group
passwd	Change the current user's password.

Permissions and Access Control

File Permissions



r: read, w: write, x: execute

Type	
Symbol	Meaning
d	Directory
-	Regular file
l	Symbolic link
s	Unix domain socket
p	Named pipe
c	Character device file
b	Block device file

File Permissions

Octal	Binary	String	Description
0	0	---	No permissions
1	1	--x	Execute only
2	10	-w-	Write only
3	11	-wx	Write and execute (2+1)
4	100	r--	Read only
5	101	r-x	Read and execute (4+1)
6	110	rw-	Read and write (4+2)
7	111	rwx	Read, write, and execute (4+2+1)

Changing Permissions

Command	Description
chmod mode filename	Change the mode of the file
chown owner:group filename	Change the owner of the file
chgrp groupname filename	Change the group for a file
chgrp -R groupname folderame	Change the group for the folder recursively.

Managing Software and Devices

YUM (Yellowdog Updater Modified)

- It package management tool that simplifies the process of installing, updating, and removing software. It resolves dependencies automatically, which means it can install required packages without requiring user intervention.
- YUM configurations are stored in **/etc/yum.conf** and repositories are defined in the **/etc/yum.repos.d/** directory.

YUM Commands

Command	Description
yum install package_name	Install a Package
yum remove package_name	Remove a Package
yum update	Update All Packages
yum list installed	List Installed Packages
yum search package_name	Search for a Package

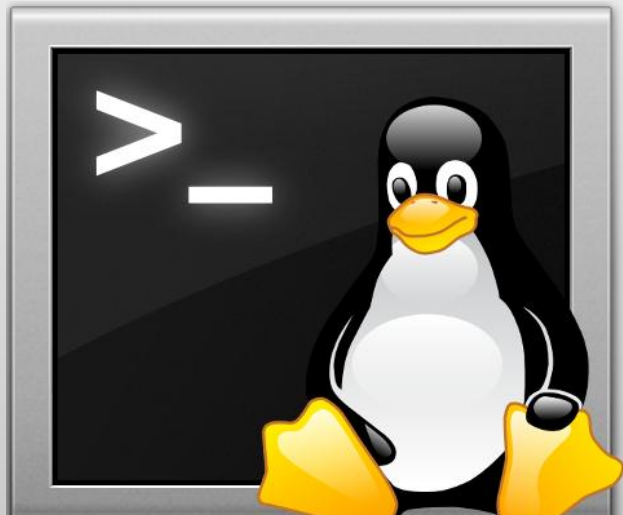
DNF (Dandified YUM)

- It is the next-generation version of YUM and is designed to address some of the limitations and inefficiencies of YUM. It allows for faster package management operations.
- DNF configurations are stored in **/etc/dnf/dnf.conf** and repositories are defined in the **/etc/yum.repos.d/** directory.

DNF Commands

Command	Description
dnf install package_name	Install a Package
dnf remove package_name	Remove a Package
dnf update	Update All Packages
dnf list installed	List Installed Packages
dnf search package_name	Search for a Package

Basic Shell Commands and Utilities



Variable Types

- A shell variable is a special variable that is set the shell and is required by the shell in order to function correctly.
- **Variable Types:**
  - **Local Variable**: is a special type of variable which has its scope only within a specific function or block of code.
  - **Environment Variable:** is a variable with global scope. It is accessible throughout the program.
  - **Shell variable:** is special type of variable created and maintained by Linux Shell itself.

Function in Shell Scripting

- A function is a collection of statements that execute a specified task.
- Advantages of using the Functions:
  - allow you to write code once and use it multiple times throughout your script.
  - help organize code into logical blocks, making scripts easier to read and understand.
  - simplify debugging and updating code since changes can be made in one place rather than throughout the script.