Linux -

Introduction -

Linus Torvalds created linux as a project in his university to create a new free open source OS kernel that will be availabe to everybody Because linux is free, a lot of distros developed to specific uses and applications.

Some of them are:

Desktop PC's with distros like ubuntu, mint, popOs! for new users in the platform

Web servers and Mainframe

Data networking and Security, linux is used widley on penetration testing

with kali linux because of it's preinstalled tools like nmap and ettercap

Embedded systems and IoT devices, which are smaller devices with low resources that are used for smart houses and other smart devices that need a small OS.

Android for phones

Most server use linux for some reasons

Versatility -

Linux is available for every type of application, from web servers to cloud Due to it's open source nature linux is highly customizable

Security -

Because linux is open source and very popular, a lot of eyes are looking at the code of it and fix it, In addition a lot of big companies like IBM and Intel are constantly contributing to this project with their security experts to keep it very secure.

Flexability -

If the needs of the project changes, and new distros are becoming better for the project, it's very easy to switch it and to install tools on linux with it's package managers, and many open-source code bases are developed for linux and can be used easly and freely

Most people use windows for some reason:

1. Winodws most of the time comes pre installed on a new computer and regular users won't learn to put a new OS on their computers.

2. Windows is very user friendly in terms of GUI, and most of the system is built on

3. the intuitive graphical use to click on things insted of the black console on linux

4. A lot of daily used software is only supported on windows like computer games, editing software like adobe, and more.

linux directory

In linux every thing is a file, thus a directory is also a special kind of a file that is formatted in a special way. a directory includes the ownership on it, the permission on it, the type which isdirectory, and the data blocks of the actucal files in the directory A regular file is just a block of data that is stored on a disk and not formatted in a special way

The Red Hat Enterprise Linux is a linux distro that is widely used in IT platforms. which helps with automation, and management, and is very focused on security, compliance, and consisntent performance.

Permissions and users -

Bashrc

a file that gets executed every time a bash session starts

Bash profile

a file that gets executed every time a user is logged in

useful files

/etc/passwd - files with all the users

/etc/shadow - file with all the hash of the password of the users

/etc/group - file with all the groups and the users in the groups

Regular file permissions

The main file permission is linux are Read Write and Execute. Every file has this permissions for the owner of the file, the main group of the file and other users. We can change the file permission with the chmod command And check the permissions of a file with ls -l

another less common permissions are SUID GUID and STICKY BIT

SUID and GUID to run the file with the main group or the owner user, and sticky bit on a directory so users on the same file can edit only the files they own.

Umask

A umask is the default permissions of a file, which are the permissions that are granted to the file when it's created We can check and change the umask with with the umask command

UID

user ID is a number that is given to a user or a group and with that number the OS recognizes the user, if two users with a different name have the same UID the OS treats them as the same user, for example if a user is granted the special UID of 0 which is given to the root user by default will be counted as root and will have all the permissions root has.

GID

same as user id but for groups, each group has it's own GID

SSH

secure shell protocl is the protocol used to connect to a linux console remotly. with this protocol we can run commands on a remote machine and manage it securely. ssh encrypts all the data and commands and because of that is secure, in addtion, ssh allows to log in with keys and thus is removing the vulnerability of passwords and brute forces.

File System Stracture -

Individual user bin folder

If the files are only use by my user, I will save them in the $HOME/bin or $HOME/.local/bin

/bin vs /lib

The /bin inclueds that bineries that run when we execute a command, or a some code, but to keep our bineries small, and make our code more practical we use libraries that groups similar functionality together also in a binery but not in an executeable one. thease libraries are saved under /lib

/mnt with wsl

The /mnt directory is most of the time used for temporary mount points that are not the system, for example another disk or even another operating system with it's own files. When we use wsl our /mnt will have our C disk in a folder name "C" and

The super user -

su command

The su command allows us to open a shell as a different user and thus execute commands, and do the stuff as that user, it is useful to switch for example to the root user because it has all the permissions.

Root without sudo or su

Yes, a weak user can execute command as root without the sudo, a command is just an executable file so if we give that file SUID permissions and the owner of the file is root

setuid vs setgid

both are flags on the file or the directory but flag that sets the SUID permission is the setuid and the flag that sets the SGID permission is setgid. if the flags are on the permission is granted.

Basic Terminal Commands -

pwd - Prints the current working directory

cd - changes the directory to the give path

ls - lists the files and directories in the current dir

clear - clears the screen

cp - copies a file from a path to a path

mv - moves a file from a path to a path

rm - removes a path

cat - prints the content of a file

less - shows a file in an easy form with an easy scrolling one page at a time whereis - locate source or binary files of a command

watch - executes a command and shows the output of the command on full screen

whoami - prints the username of the effective uid

date - print or set the system date and time

echo - print the text into the screen

vim - text editor ment for programmers

reboot - powers the machine on and off

tar - used to pack a files and comperss (optionally) into an archive and also unarchive an archive into the files

gzip - attemp to compress the given file name and reduce it's size

find - search for a specific file in a given directory

grep - extracts lines that match a pattern from the input text and prints them

uniq - removes repeating lines from a text that come one after another

cut - removes sections of text from each line

wc - used to count words or lines in a text file

sort - sort lines of text by given parameters

tail - prints the last 10 lines of the text

head - prints the first 10 lines of the text

awk - is a language that is used to scan and process language text

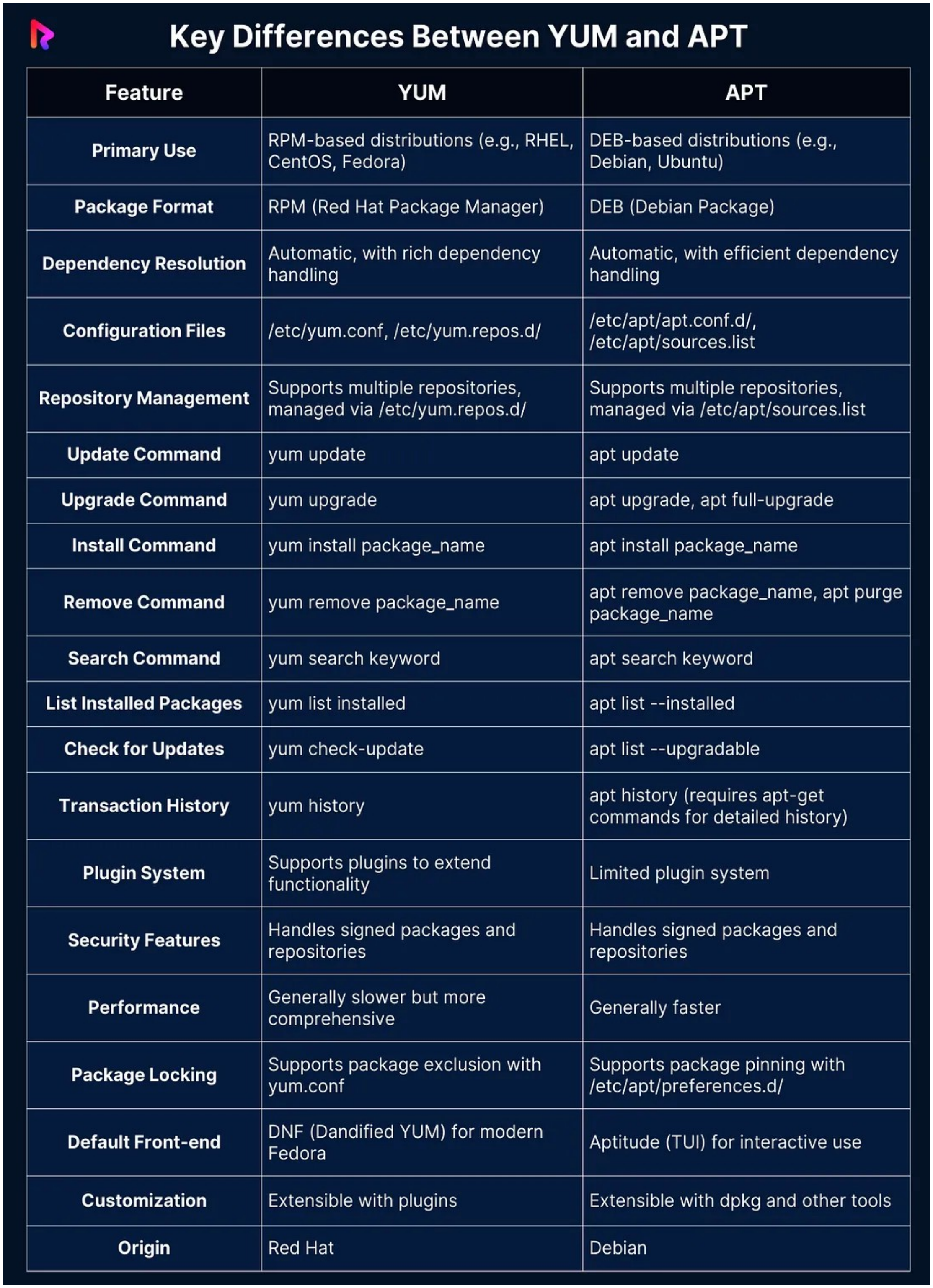
sed - strem editor to perform transformation on text, for example switch word one with the word two

man - shows the reference manual of a command

Difference between yum and apt

Package managers are software pieces to easly manage installation, deletion, upgrade and manage software. Yum and Apt are both package managers.

The package managers are very similar in terms of the installation, upadate and deletion of software, and in the overall management, and the main big difference is the apt is mostly ment for debian distrebution like ubuntu and yum is ment for Red Hat distributions such as centos



Key Differences Between YUM and APT

Feature	YUM	APT
Primary Use	RPM-based distributions (e.g., RHEL, CentOS, Fedora)	DEB-based distributions (e.g., Debian, Ubuntu)
Package Format	RPM (Red Hat Package Manager)	DEB (Debian Package)
Dependency Resolution	Automatic, with rich dependency handling	Automatic, with efficient dependency handling
Configuration Files	/etc/yum.conf, /etc/yum.repos.d/	/etc/apt/apt.conf.d/, /etc/apt/sources.list
Repository Management	Supports multiple repositories, managed via /etc/yum.repos.d/	Supports multiple repositories, managed via /etc/apt/sources.list
Update Command	yum update	apt update
Upgrade Command	yum upgrade	apt upgrade, apt full-upgrade
Install Command	yum install package_name	apt install package_name
Remove Command	yum remove package_name	apt remove package_name, apt purge package_name
Search Command	yum search keyword	apt search keyword
List Installed Packages	yum list installed	apt list --installed
Check for Updates	yum check-update	apt list --upgradable
Transaction History	yum history	apt history (requires apt-get commands for detailed history)
Plugin System	Supports plugins to extend functionality	Limited plugin system
Security Features	Handles signed packages and repositories	Handles signed packages and repositories
Performance	Generally slower but more comprehensive	Generally faster
Package Locking	Supports package exclusion with yum.conf	Supports package pinning with /etc/apt/preferences.d/
Default Front-end	DNF (Dandified YUM) for modern Fedora	Aptitude (TUI) for interactive use
Customization	Extensible with plugins	Extensible with dpkg and other tools
Origin	Red Hat	Debian

Linux files, IO and Piping

Regular files

Regular files most of the time contain text, executable text, images, data of a program etc..

Directories

A directory is a file that is ment to oragnanize and locate files and directories, and the binary format is used so that directories containing large numbers of filenames can be search quickly.

Device Files (special)

Thease files help us to manage the input and output, and appear in the system just like a regular file.

Character files - are used for input and output of one character at a time, this type is called raw device access.

Block files - are also used for input and output of blocks of data of a fixed size, this type is called block device access.

Every device has only one type of file to access to it (character OR block)

Links

Tool for having multiple filenames for the same data in disk.

Hard link - points to the same memory location of the original files, and acts as it's own file with the same memory location with a copy of the file attributes (location on disk, permissions etc), if the original file is deleted it can still be accessed with the hard link. A regular file is also a hardlink

Soft link - soft link is a pointer to the original file, so if we delete the original file, the soft link won't point on nothing so it will become invalid, this type of link is called 'stale link'.

Named pipes

Tools that allow two or more system process to communicate with each other using a file that acts as a pipe between them this time of communication is known as inter process communication (IPC)

Sockets

sockets are also files that help with inter process communication, the only difference between sockets and pipes is that sockets will facilitate the communication between processes running on a different systems or over the network.

files types in ls -l

"-" regular file

"d" directory

"c" character special file

"b" block special file

"l" symbolic link

"p" named pipe

"s" socket

use of symbolic links

the use of symbolic links is to give multiple access points to the same file or directory. for example shortcuts, that are used to make it easy to access a file from the desktop or home directory without actually putting the file there. and all the symbolic links update automaticl if the file changes or moved

Transfer value of one command to the other

we can use the output value of a command as the input of another command if we use the pipeline character | for example, we can print a file and then sort it line with this command 'cat file_name | sort'

Services

Three main services are

httpd, bluetooth, networkd

what is a service

a service is a program that is run by the system to perfrom a task for the system services can be one timed at the system start for example fsck (file system check), that runs or boot or for a long time such as sshd which is called a daemon

What is a daemon

a daemon is a service that runs on the background and as long as the system is runnin. For example sshd or networkd that always run and handle ssh or other connections

called like that because it's like a demons in the background that do some jobs for us.

Process

process is a set of instructions that is runs on the system, it can be a regular program service or daemon

Parent process

process are created with the fork syscall, to copy the original process and then exec to switch the running instructions and resource usage. The parent process is the process that is being forked and for most of the start processes is the init process that is started on boot

Orphan process

this is process that it’s parent finished running before he finished, so he has no parent, this kind of process are “adopted” by the init process which have he pid of 1

Zombie process

zombie process is a process that finished running but he still has an entry in the process table because it’s parent process didn’t sent him SIGCHLD to get the exit status and resource usage

Administration commands

hostname - show's the system hostname and DNS name

systemctl - command to manage services on the system and to interact with systemd

who - display information on the currenly logged in users in the system

free - show's the amount of free and used memory in the system

top - show's an ordered and updating list of processes in the system

vmstat - gives a report on the virtual machine resources

ps - show's a snapshot of the current processes in the system

Detect problems in the system

This depends on the problem of the system.

Most of the time I think the problem would be on a certain service or in a ceratin process. With the systemctl command we can easly manage services, determine thier state, and start them, stop them or restart them if needed and see thier status and logs. and for further diagnostic we can use the journalctl command.

If the problem is from a certain process, we can check it's status with the ps command with the flag aux or -ef for more information on the process. we can also use the top command for and updating list that is easier to see.

if this is a storage problem we can use the free command or the vmstat command if we are on a vm to see how much storage left in the system.

What is a process

a process is an exection of a program the is loaded into memory. each process contains additional information to help the system to identify it, this info includes, PID (process identity), state, parent process PID, children process PID, siblings, processor register and information about it's status and more.

linux processes are located at the /proc directory and each process has it's own directory with all the information associated with it.

what is #!/bin/bash

The #! part is called a shabang, which tells the operating system with which executeable to run this command. the shabang must be on the first line of the script otherwise it will count as a comment.

the /bin/bash is the full path of the interpreter to run the script with, in this case bash.

shell processes

no, the shell is a command, like all others and it runs on the process the system gives it. different shells run from a different commands, so they have different processes

File system utility

how the system knows where to save each path

The system knows in which path to save each path, because when we mount a partition we give it a base path on /, for example we can mount the /dev/sdb1 partition on /mnt/partition, and every file that will be created or transfered or edited in this base path will be saved on this partition.

also, the default partitions are defined in /etc/fstab

/etc/exports

the /etc/exports is a folder that contains all the permanent shares of nfs files that we created, when the system boots, restarts etc.. it will by default export all the files in /etc/exports

User mode / Kernel Mode

user mode is the unprivlaged mode of the system where users can't access the hardware an they can't do much. this is good because this makes them to be able to make mistakes without crashing the entire computer. kernel mode has access to anything it want's which makes it very strong, and mistakes and crashes in kernel mode are very dangerous

Interaction with hardware and syscalls

When a user wants to interact with the hardware, he can't do it in the user mode, because only the kernel mode has access to that, but because we still wants our users to be able to interact with the hardware for creating files, printing etc we have systemcalls, which are very carefuly tested functions that are allowed to be executed in kernel mode by users. with thease systemcalls we can interact with the hardware to create files, print and do all the other stuff.

when we create a file, we don't switch into kernel mode, we use systemcalls to do the hardware stuff for us.

The linux boot process

BIOS vs PXE

while regular bios does the POST process and then boots up the operating system from the disk, the PXE process downloads the operating system image directly to RAM and then boots the system up. this is useful in a diskless stations, or for booting a machine without a floppy disk of some kind.

What is the initrd

when the system is booting, most of the time a generic image of the linux kernel is used, this kernel has a lot of modules in it, and some of them are not needed and can crush, for example module fails to activate wifi because the nic don’t support it, to prevent this from happening and create a more generic, fast and resilient to error booting process the initrd or initramfs is used.

The initrd stands for the initial RAM disk, because the regular root partition might not be localy on the disk, and it can be mounted on an nfs file system, or in an LVM which requires more cases to handle by the boot process, the initrd is mounted temporarily as the root file system, this has the minimal tools and resources to get the real file system, for example from nfs.

What is the boot process on linux

<u>BIOS / UEFI:</u>

BIOS (Basic input output system)

The BIOS is the process that is done in the hardware on powerup, and is ment to start the system kernel

The first stage is the POST (Power On Self Test), this process is ment to check and verify that the computer hardware is configured correctly and in a working postion, and the kernel can be booted without a problem.

Then the BIOS searches a boot loader and executes it.

UEFI (Unified Extensible Firmware Interface)

The UEFI is a much smarter booting process that most modern computers use. This is a "mini OS" that allows for booting with

PXE, and manage configuration even without OS, its loading the bootloader automaticly and is able to boot from both MBR and GUID

MBR (Master Boot Record):

The MBR is the first sector of the disk and it holds the information on how the disk is partitioned, the MBR also is charge of loading the boot loader or the GRUB, The MBR can only utilize disks the are smaller them 2.1TB because of the address limitations. 2^32 * 512B sectors ~= 2.1TB

GPT (GUID (Globaly Unique IDentifier) Partition Table):

Improved version of the MBR. this is used when booting with UEFI instead of bios, GPT can be used with bios if the first partition has the configuration like the MBR that the bios expects. The GPT also support much larger disks because it has 2 ^ 64 address space, so with 512B sectors it can support about 2 ^ 64 * 512B ~= 8ZB

GRUB (Grand Unified Boot Loader):

Grub is a second level boot loader, this is a software that run before the operating system and it can. the bios when starting to boot the system boots the GRUB, and the grub can help us to choose which partition we want to boot, so we can have multiple operating systems.

Kernel :

In this stage to operating system is mounted on the disk, and the main process is executed (/sbin/init) which most of the times points into systemd

initrd may be mounted on the RAM temporarily, before the operating system mounts the root directory because the file system may not be on the disk for example nfs or it may be on a multiple disks for exmple lvm, so the initrd functions as the root directory temporarily and load all the necesery tools to load the real root file system

Init:

Init is the first process that is being run by the kernel, and thats why it has the PID of 1, the init process has a run level, with mostly used 3 which stands for multi user with cli interface and 5 which stands for multiuser with graphical interface, this runs all the services with the already initialized systemd process

RunLevel Programs:

after the system has initialized systemd and all the services that are used for noraml functioning, the runtime programs start to be initialized, and have the text on the screen of "starting sendmail ….. OK", after all thease startup programs have been initialized, the user can use the system

Booting process

POST – power of self test, system checks that all the devices are ok

Boot Device Search (MBR or GUID) – the bios searches for a bootable device, guid partition is marked as bootable and mbr has the first sector for booting

Boot Loader – the boot loader is most of the time loaded and it is responsible to load the system from the storage

Kernel initialization and systemd process – the kernel is ran by the boot loader and it initializes systemd

Advantages of systemd over sysV-init

1. parallelization and faster boot time

2. dependency resolution - auto dependency resolution, when a service depends on other service, systemd ensures correct order startup

3. Monitoring and service restart

4. cgroups and process tracking - cgroups allows for better resource management

5. service management with systemctl - this tool allows for easy service management (start, stop, reload, enable, disable) instead

of bash scripts

6. snapshotting and state restoration

Login and Logout process

important files in the logon process

the files that are very important for the user log in is the ,/etc/password, /etc/shadow, and also /etc/profile for initialization

unique files for every user

.bash_profile to configure login stuff, and .bashrc to configure each bash session

what is getty and agetty

getty is the first program used for login that was written for the unix operating system. agetty is an alternative program written for linux usage.

Nsswitch

This file is used to tell the system from where to take certain services files such as /etc/passwd /etc/shdow etc...

Services in the nsswitch.conf

passwd, group, shadow, gshadow, hosts, networks, protocols, services, ethers, rpc, netgroup.

Switching options with nsswitch

files (local files), systemd, dns, db

Advanced commands for administartors

jq - command line tool for parsing and working with json file

lsof - lists all the open files on the system

lspci - lists all the devices connected with pci, for example ram, nic

lsblk - lists all the block devices on the system, for example hard drives, ssds

starce - traces and records the systemcalls and signals generated by a command

fdisk - command used to create and manage disk partitions

mkfs - command that is used to make file systems on formated partitions

tune2fs - tool for managing and fine tuning the ext2-4 file systems

fsck - used to check and repair file systems

crontab - manage crontab and scheduled tasks for users

nmcli - network manager command line, used to configure and control the network manager

nmtui - network manager text user interface, used for text ui for the nm

tcpdump - dump the network traffic, prints the content of passing packets

nmap - useful tool for mapping the network, with port scanning, ip scanning and more

telent - interactive communication using the telnet protocol with another host

netcat - easly establish tcp and udp connections in the terminalman

What is the largest directory

The largest directory in the / (root) directory, and it is because it is the main file system, that holds everything

What is the root directory, why is it the largest

The root directory is the starting directory of every path in linux, and it contains all the files in the system in it

Add another network interface

Linux does not allow two default routes on the same routing table. so we can team both interfaces together to one logical nic

LVM

problem with shrinking

when there are files on the filesystem inside the lvm volume, it is very important that we shrink the filesystem also, if there is a problem with shrinking the file system volume, it is not recommended to shrink the lvm volume because the file system will think it has more storage then it really has.

why to use lvm

it is very useful to use lvm because we can combine the storage capacity of multiple storage devices and thus we can storge bigger files and manage our memory more easly

File system concepts

what is an inodes

inode is a metadata that is stored on a file, this includes it's file type (directory, regular file, link etc..), it's permissions, the owner user and group, creation time and more, most importantly it holds the data blocks of the file on the disk

each inode is identified by a number that is unique per file system, the same number can be used in different file systems

The inodes are located in the inode table that is allocated when the file system created, and there is a limited number of inodes per file system

what is the ext4 file system

ext is the first file system in linux, and it's ment to solve the problem of the file system that was on the MINIX operating system that was used in the early versions of linux, this file system had corrently 4 generations, and ext4 is the one used today, the main changes are reducing in fragementations, larger address space for files, journaling of changes, better saving of data so there are less corruption during power loss because changes are first written into the journal and then being transferred to the original file

what is the xfs file system

the xfs file system is ment for large files and data bases it is very scaleable and performes much better on large files then ext4, it supports jounaling and snapshots and is ideal for enterprise environments and large databases

connection between inodes and lvm

lvm itself don't use inodes, because it's a smart way to combine physical volumes, while inodes are the metadata the is stored on files.

The connection is that if we decide to use lvm and put our disks on a volume group, the inodes that will be created on an lvm partition.

Advenced linux and security

the use of cgroups

connection of namespaces and containers

containers are the idea to give an isolated environment to a program so it can be shiped with all of it's dependencies. namespaces is a linux feature that helps to isolate processes, users, file systems and more, so we can create the isolated environment for our process

what does selinux solves

selinux seeks to make the linux system much more secure by defining rules and policies that can give users only the permissions they need instead of root privlage

selinux vs AppArmor

app armor is the main competitor of selinux which is avilable on suse linux, app armor is built on the DAC, making it more secure but also mandatory, SElinux acts just like a 3rd party software, and don't use the existing file permissions on linux

what are capabilities

capabilities are the abilities of the root user, insted executing commands as root, so they are with the highest privlages everywhere, we can give our environment certain capabilities so the env has only eleveted privlages in that section, for example CAP_SYS_TIME to change the time of the system

groups of capabilities

yes we can group them by logic to several groups

Administrative capabilities, process management, system management, network management, file system management, user management