

#linux

#os

#scheduling

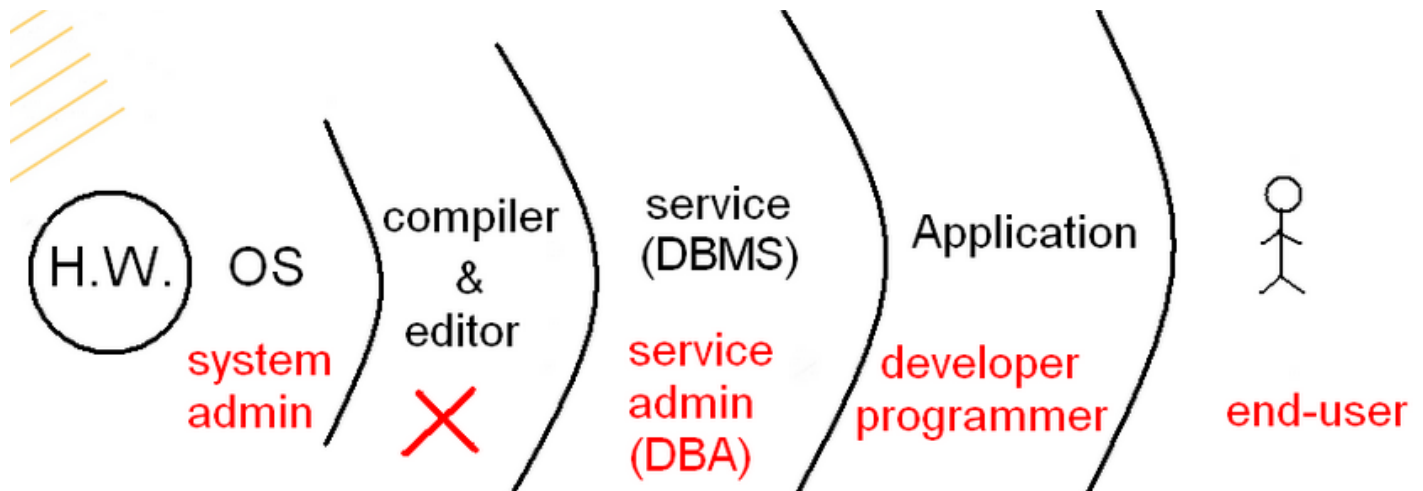
LPIC 1 class

Session 1

GNU/Linux → OS

Linux → Core or Kernel

40 years ago architecture → Multiuser Single server



OS = Kernel + tools

GNU/Linux = Linux + GNU

File system: store and recovery of files in system

- Format
- Block size

Multiuser Single server Architecture

Users with Terminal (IU and OU)

CPU		ALU		
Input Unit		CU		OU
		MU		

Client server Architecture

Every OS needs to work with special Filesystems

- Linux native file system: EXT4, btrfs
- Non-native: NTFS, XFS, APFS, ZFS, SWAP
- Cross-platform FS: FAT8, 16, 32, CDFS, ISO

mk

SHELL

GCC = GNU C Compiler

Enterprise Environments

- Big scale, scalable, multilocation, multiservice, complex
- high con-current users, 24/7
- zero down time = high availability
- separability of internal / external parts
- Income,

My ubuntu details:

```
user:hes          user:root
pass:123456       pass:123456
@10.0.2.15/24     ethernet 3 -> 172.17.17.17/24 root#192.168.160.1 #192.168.167.2
```

- share folder : D:\Hesam\Linux_shared

Anisa ftp
http://ftp.anisa.co.ir/OVA/RockyLinux_9_Minimal.ova
http://ftp.anisa.co.ir/ISO/Rocky-9.0-x86_64-dvd.iso
[Alireza's at share folder](#)

Client / Server old
→ 2-layer → 3-layer → N-tier

middle-tier

IPC - ICP ([Internal Communication Protocol](#))

Administration	DevOps	Developer
LPICs		
LPIC-3		
300 Mix Env. (SMB, LDAP)		
303 Security		

Administration	DevOps	Developer
305 Virtualization		

Linux

1. Debian based
 2. Redhat based
- LPIC 3
 - SMB protocol - CIFS protocol for connecting Windows - Samba (Windows mask on Linux)
 - LDAP protocol - LDAP servers implementation by providers
 - (Active directory Windows, Oracle Internet directory, Linux: several like OPEN LDAP)
 -
 - Security Enhanced Linux: for Application limitation (like firewall)

Anisa FTP with files for download

→

→

Session 2

? to review

part 2

1. `root` : as super user
2. `root` : `filesystem root` highest position in hierarchy - `/`
3. `root` : `root` a root's home directory - `/root/` دارکتوری هم نام یوزر
4. `root` : `root` a group - `/root/`
5. `root` : `root` as a access level

tilde ~ current user home directory

when login → go to home of user

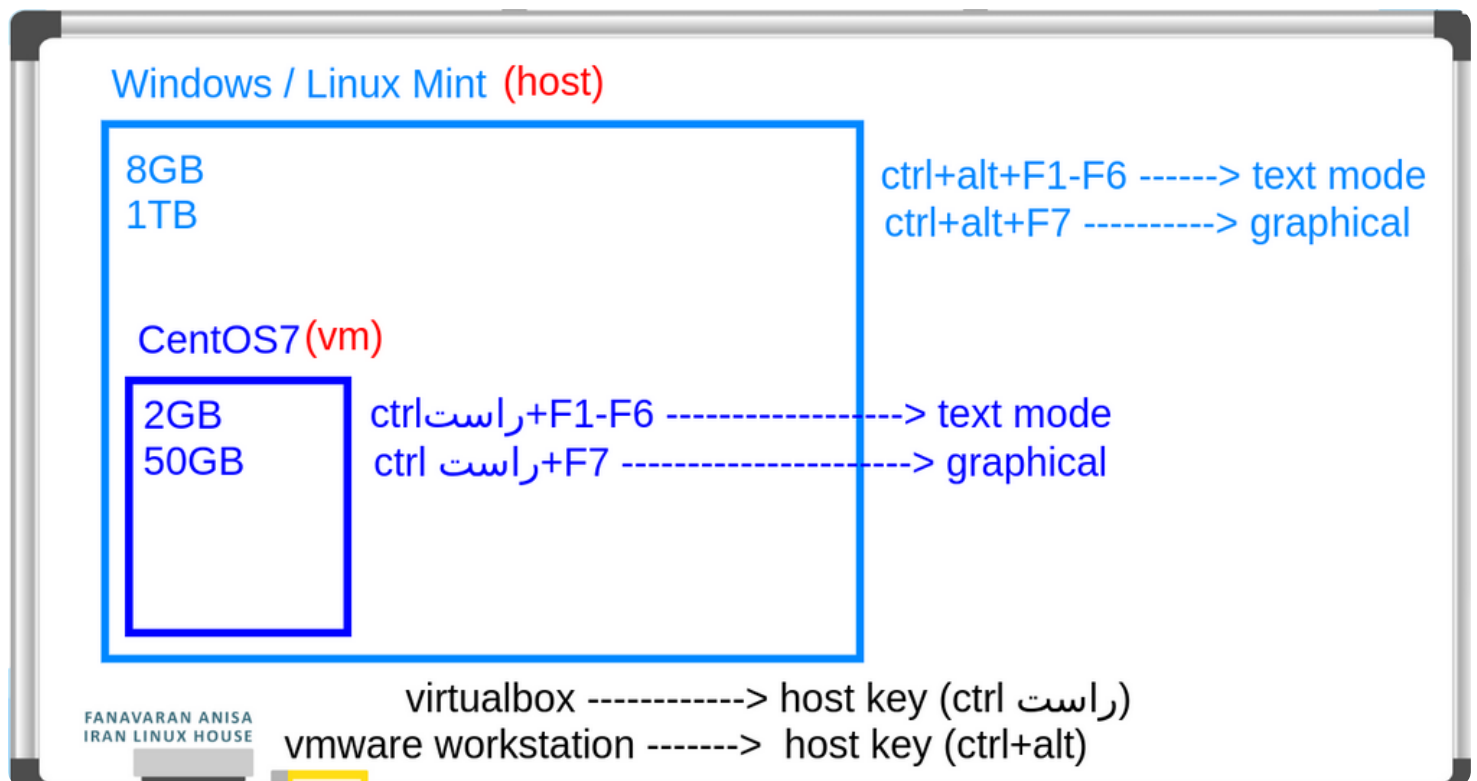
- `whoami`
- `pwd` - `cd`
- root account identifier at shell: `#`
- rest of user: `$`
- switch to `root` user
 1. Run `sudo <command>` and type in your login password, if prompted, to run only that instance of the command as root. Next time you run another or the same command without the `sudo` prefix, you will not have root access.
 2. Run `sudo -i`. This will give you an interactive root shell. Note that the `$` at the end of your prompt has changed to a `#`, indicating that you have root access. But you fall in the root

home directory (`/root/`). From here you can run any sequence of commands as root, or run the command `exit` to leave the root shell.

3. Use the `su` (substitute user) command to get a root shell. This is effectively the same as using `sudo -i` . Note that when you use this command it will ask for the root password and not your login password. These are not the same. You may have to set or change the root password by running `sudo passwd root` first.
4. Run `sudo -s` . This gives you root access, but maintains your current SHELL. Shell specific settings, including your current directory, are preserved. For instance if you use `bash` (Ubuntu's default shell), aliases (and any other settings from `~/.bashrc`) are kept when you switch to the root user. To leave the root access, type `exit` as in the cases above.

Session 3

have 6 text mode and one graphical mode



right Ctrl +

Ctrl+alt+ F + {1,...,7}

- there exist `system` and `service/application` related users
 - like `ftp service` , `mail server service` , etc
 - We can check them in `config` file [later](in course)
- check whole users → check config file
- shutdown → `init 0`

< i see till min 41:03> exercises at <-48:00>

shutdown	init 0
reboot	init 6

whoami

pwd

dir or ls ls <address> ls /Var>

useradd with privileges of root

control + F2

useradd → passwd to set password

hostname <computername>.<domain name>

```
echo "server1.lpic.org" > /etc/hostname
init 6
```

SHELL

- reboot to be possible to see the file

ps ?

- Zoom in or out in Ctrl+Shift Ctrl
- clear screen clear
- R Ctrl+l hostkey
- L Ctrl+ C
- ls -l long list with details
- cat

```
ls /boot
```

```
boot loader# grub2
initrd #
# kernel linux file which contains version
```

SHELL

- -l (long listing), -a (show hidden dot files), and -t options (list by time)
- Sometimes, an argument is associated with an option .
- argument must immediately follow the option.
 - single-letter options, the argument typically follows after a space.
 - full-word options, the argument often follows an equal sign (=).

```
ls ?
```

```
ls --hide=Desktop
```

SHELL

1. دسترسی ها permission access

2. hard link

5. size (Byte)
6. modify time (mtime)
7. filename

- `ls -l -h <switch> <option>` > option does not work stand alone
- `ls -a` all provide all which add things start with `.`
- `.` and `..`
 - absolute address vs relative addressing
 - `ls .` folder im am in
 - `ls ./..` = `ls ..`
- 1993 - 1994 FHS Filesystem Hierarchy Standard → LSB (Linux Standard Base)
- Long Term Support LTS
- [check distro differences](#)
- Linux features LSB, LTS

• \ : ریشه فایل سیستم و شروع آدرس دهی از این دایرکتوری

- `/boot` : برخی فایل های اصلی در این قرار دارند از جمله
- `/:`
- `/:`
- `/bin` : فایل اجرایی دستورات عمومی سیستم که توسط همه کاربران قابل اجرا هستند در این مسیر قرار می گیرد. اصطلاح این : `user commands` یا `general commands` دستورات می گویند
- `/sbin` : فایل اجرایی دستورات عمومی سیستم که توسط همه کاربران قابل اجرا هستند در این مسیر قرار می گیرد. اصطلاح این : `administration commands` یا `system commands` دستورات می گویند
- `/lib` : فایل اجرایی دستورات عمومی سیستم که توسط همه کاربران قابل اجرا هستند در این مسیر قرار می گیرد. اصطلاح این : `user commands` یا `general commands` دستورات می گویند
- `/opt` : `<optional>` third-party applications

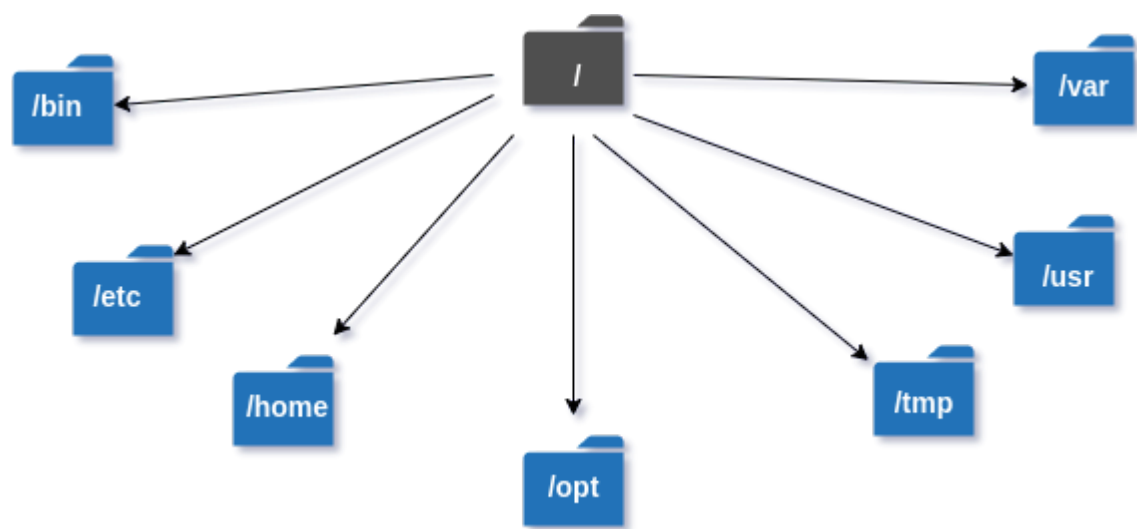
تکالیف

- Window
 - client
 - server
- Linux
 - client: N/A
 - server:
 1. Server (core) → ubuntu server, RHEL, OEL,
 2. desktop → ubuntu server, RHEL, OEL,
- `ls /bin/ | more` : only go done, `q` for quit
- `ls /bin/ | less` : ↑↓ work
 - less or more pagination
- `|` : pipe operator : to side of operator should be command
- `cat` :
- `cat <filename> | ws -l` or `ws -l <filename>` :

- `less <filename>`
- ``nl`
- `cat file | nl | more`
- `4700 PB` = $4700 \times 1000^5 \text{Byte} = 4700 \times 1000^4 \times 1000 \text{Byte} = 4.7\text{e}+15 \text{ KB}$
- `4700 PB` = $4700 \times 1000^5 \text{Byte} = 4700 \times 1000^7 \times \frac{1}{1000^2} \text{ZB} = 0.0047 \text{ ZB}$
- `4700 TB` = $4700 \times 1000^4 \text{Byte} = 4700 \times 1000^3 \times 1000 \text{Byte} = 4.7\text{e}+12 \text{ KB}$
- `4700 TB` = $4700 \times 1000^4 \text{Byte} = 4700 \times 1000^7 \times \frac{1}{1000^3} \text{ZB} = 4.7\text{e}-6 \text{ ZB}$

linux-directory-structure

These are the common top-level directories associated with the root directory:



Directories	Description
/bin	binary or executable programs.
/etc	system configuration files.
/home	home directory. It is the default current directory.
/opt	optional or third-party software.
/tmp	temporary space, typically cleared on reboot.
/usr	User related programs.
/var	log files.

Some other directories in the Linux system:

Directories	Description
/	

Directories	Description
/boot	It contains all the boot-related information files and folders such as conf, grub, etc.
/dev	It is the location of the device files such as dev/sda1, dev/sda2, etc.
/lib	It contains kernel modules and a shared library.
/lost+found	It is used to find recovered bits of corrupted files.
/media	It contains subdirectories where removal media devices are inserted.
/mnt	It contains temporary mount directories for mounting the file system.
/proc	It is a virtual and pseudo-file system to contains info about the running processes with a specific process ID or PID.
/run	It stores volatile runtime data.
/sbin	binary executable programs for an administrator.
/srv	It contains server-specific and server-related files.
/sys	It is a virtual file system for modern Linux distributions to store and allows modification of the devices connected to the system.

Log Files:

Log Files	Descriptions
/var/log/lastlog	It stores user's last login info.
/var/log/messages	It has all the global system messages
/var/log/wtmp	It keeps a history of login and logout information.

System Configuration Files:

Configuration Files	Description
/etc/bashrc	It is used by bash shell that contains system defaults and aliases.
/etc/crontab	A shell script to run specified commands on a predefined time interval.
/etc/exports	It contains information on the file system available on the network.
/etc/fstab	Information of the Disk Drive and their mount point.
/etc/group	It is a text file to define Information of Security Group.
/etc/grub.conf	It is the grub bootloader configuration file.
/etc/init.d	Service startup Script.
/etc/lilo.conf	It contains lilo bootloader configuration file.
/etc/hosts	Information of IP and corresponding hostnames
/etc/hosts.allow	It contains a list of hosts allowed accessing services on the local machine.

Configuration Files	Description
/etc/host.deny	List of hosts denied accessing services on the local machine.
/etc/inittab	INIT process and their interaction at the various run levels.
/etc/issue	Allows editing the pre-login message.
/etc/modules.conf	It contains the configuration files for the system modules.
/etc/motd	It contains the message of the day.
/etc/mtab	Currently mounted blocks information.
/etc/passwd	It contains username, password of the system, users in a shadow file.
/etc/printcap	It contains printer Information.
/etc/profile	Bash shell defaults.
/etc/profile.d	It contains other scripts like application scripts, executed after login.
/etc/rc.d	It avoids script duplication.
/etc/rc.d/init.d	Run Level Initialisation Script.
/etc/resolv.conf	DNS being used by System.
/etc/security	It contains the name of terminals where root login is possible.
/etc/skel	Script that initiates new user home directory.
/etc/termcap	An ASCII file that defines the behavior of different types of the terminal.
/etc/X11	Directory tree contains all the conf files for the X-window System.

Virtual and Pseudo Process Related Files:

Virtual and Pseudo Process Related Files	Descriptions
/proc/cpuinfo	CPU Information
/proc/filesystems	It keeps useful info about the processes that are currently running.
/proc/interrupts	it keeps the information about the number of interrupts per IRQ.
/proc/ioprots	Contains all the Input and Output addresses used by devices on the server
/proc/meminfo	It reports the memory usage information.
/proc/modules	Currently using kernel module.
/proc/mount	Mounted File-system Information.
/proc/stat	It displays the detailed statistics of the current system.
/proc/swaps	It contains swap file information.

Session 4

- Programs
 - compiler - binary file
 - system related `/sbin`
 - person related `/bin`
 - interpreter
 - python, shell, perl - text file (script) - `~` ?
- executable file
 - windows
 - Linux
 - executable file
 - use permission to use `rwX`
 - whatever get `x` permission
- `wc -l` line count
- `cat localrepo | nl`
- `ls /boo -a -l -h` arguments
 - 0 referencing
 - `ls -l -h -a /boot` more Linux
 - similar arguments can be mixed `ls -alh /boot`
 - switches without values can be merged
 - `command -a <value> -b <value>`
 - `ls -lhtr` ? ?????
- unix style: `ls -a` , `rpm -i ...` , `ps -a -u -x` or `ps -aux`
- GNU style `ls --all` , `rpm --install ...`
- BSD style `ps aux` , `rpm --install ...`
- `user manual` or `manual`
 - `man ls` : `man <name>`
 - 1. user command - default user - `/bin`
 - 2. system command - `/sbin`

NAME `ls` - list directory contents

SYNOPSIS
`ls [OPTION]... [FILE]...`

DESCRIPTION
 List information about the FILES (the current directory by default). Sort entries alphabetically if none of `-cftuvSUX` nor `--sort` is specified.

Mandatory arguments to long options are mandatory for short options too.

`-a, --all`
 do not ignore entries starting with `.`

`-A, --almost-all`
 do not list implied `.` and `..`

`--author`
 with `-l`, print the author of each file

`-b, --escape`
 print C-style escapes for nongraphic characters

`--block-size=SIZE`
 scale sizes by SIZE before printing them; e.g., `'--block-size=M'` prints sizes in units of 1,048,576 bytes; see SIZE format below

Manual page ls(1) line 1 (press h for help or q to quit)

- search `/`
 - `n` : next match
 - `N` : previous match
 - `q` : quit
- `echo $?`
- `touch`
 - `touch .f4 touch ../.f5 touchf4`
 - `~` home directory of root user
 - `rm -i <filename>` : `i` = `interactive`
 - in `redhat` and `centos` and??? : as these are for servers to prevent deleting files
 - `rm -f` - `force`
 - `cd -` : last address
 - `history`
 - `!number`
- `cd ~` or `cd ../../../../root` or `cd ../../../../root` `cd $HOME` `cd/root` or simply `cd`
- absolute vs relative address
 - absolute: not important where we are: only address `destination` start with `/`
 - `.` currnet directory
 - `..` currnet directory
 - relative

```
cd /var
```

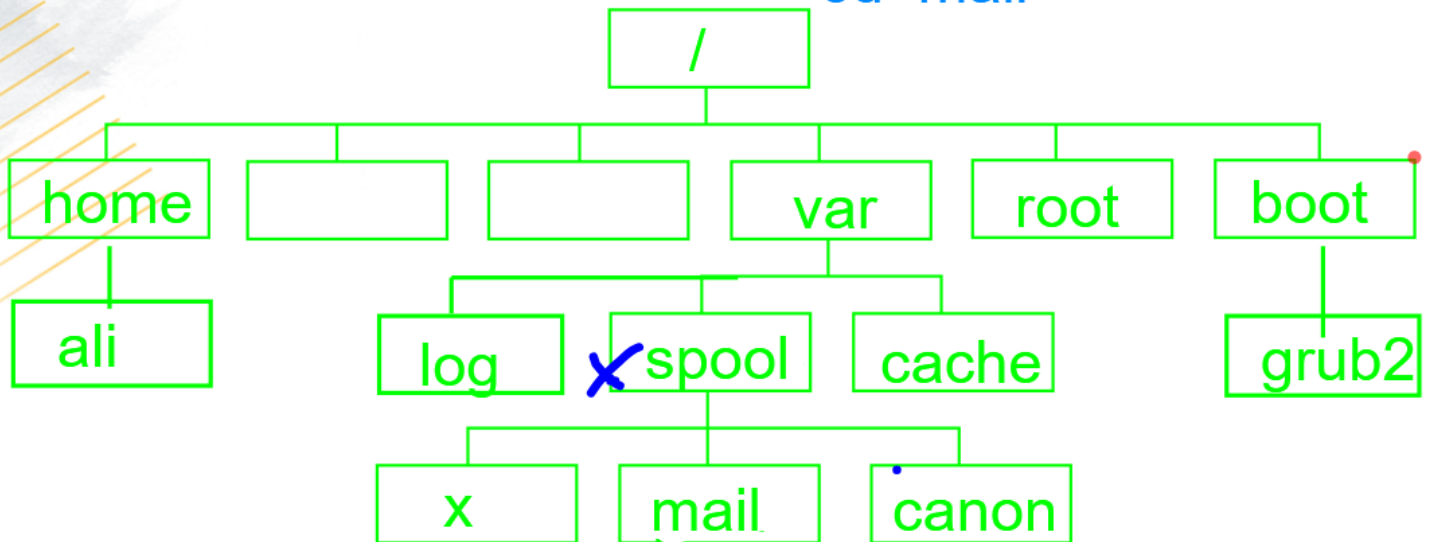
```
cd ../
```

```
cd ..
```

```
cd /var/spool/mail
```

```
cd ./mail
```

```
cd mail
```



- `mkdir /root/1403/mehr/11/linux/class/`
 - `mkdir/root/1403` `cd ~
 - `mkdir/root/1403/mehr` `mkdir ./1403 cd ~
 - ...
 - missed for few minutes
 - `rm f?`
 - `mkdir -p`
- `auto completion` : to character
- `echo`
 1. print argument:
 - `echo salam baxs`
 - `echo "salam baxs"`
 2. see `variable` value
 - `V1=500` : in RAM -= memory
 - `echo V1` `echo $V1`
 - `echo $?` : `?` exit code = 0 correct else not correct
- `echo $PATH`
- `which mkdir` the one is nearest
- `which -a mkdir`
- `whereis mkdir`

File Types

- `-` regular files (data → txt, jpg, pdf, mp3)
- `d` directory
- `l` symbolic link (shortcut)
- `c` special device, character
- `b` special device, block
- `p` pipe: interfacing 2 file
- `s` socket: 2 applications
- `file <filename>`
- `ls -l zero`
- `ls -l cdrom` → `ls -sr0`
- `ls -l sd*`
- `touch f4.mp4`
- `reset`
- `cd /dev` and `cd /etc` and check file types
- `pwd > f1`
- command `>` file: to write to file (making file method 2)
- command `>>` file : to append (making file method 3)

exercise 1

```
cd
echo Hesam > myinfo.mp3
echo MH >> myinfo.mp3
echo 09350000 >> myinfo.mp3

cd /root/1043/mehr/11/Linux/calss

echo "Hesam
MH
09350000" > myinfo.mp3

echo Hesam > myinfo.mp3
echo MH >> myinfo.mp3
echo 09350000 >> myinfo.mp3
```

SHELL

Find file python - check version

- python - check version

```
ls /usr/bin/python*
which systemctl

python3 --version
```

SHELL

Session 5

exercises of session 4: i missed

- copy file: `cp <source file> <destination>`
 1. make new copy in same place with new name: `cp myinfo m1`
 2. make new copy in other place with same name: `cp ./myinfo ./root/hes/`
 3. make new copy in other place with new name: ``cp ./myinfo ./root/moz.txt`
 1. ``cp ./myinfo ../../hes_info.txt`
- move file: `mv <source file> <destination>`
 1. rename file in same place : `mv myinfo m1`
 2. move = cut and past : `mv ./myinfo ./root/hes/`
 3. move + rename: ``mv ./myinfo ./root/moz.txt`
 1. ``mv ./myinfo ../../moz.txt`
-

```
echo "1403/07/18" >>moz.txt
```

SHELL

```
date >> moz.txt
```

Exercise:

1. make following path ``/root/1403/mehr/18/linux/class/mydir`
 - since does not have `/` at end, it is file and should be made by touch

```
mkdir /root/1403/mehr/18/linux/class/mydir
```

SHELL

```
mkdir /root/1403
```

```
cd ./1403/
```

```
mkdir /root/1403
```

```
cd ./1403/
```

```
## or
```

```
mkdir -p /root/1403/mehr/18/linux/class
```

```
cd /root/1403/mehr/18/linux/class
```

```
touch mydir
```

```
# or
```

```
touch /root/1403/mehr/18/linux/class/mydir
```

2. remove file `mydir`
3. move file `myinfor` `11/class` to `18/class`

```
mv ./root/1403/mehr/11/myinfo ./root/1403/mehr/18/myinfo
```

SHELL

4. `myinfo.mp3`

```
date >> myinfo.mp3
cp ./myinfo.mp3 ./opt/myinfo.mp3
```

SHELL

5. myinfo.mp3

```
cd ./opt
cp ./myinfo.mp3 hajiz.pdf
```

SHELL

6. myinfo.txt

```
cp ./root/hes/myinfo.txt ./root/hes/1403/
mv ./root/hes/myinfo.txt ./root/hes/test47
```

SHELL

Exercise:

```
dmesg >
```

SHELL

- dmesg : command for getting ????
- copy directory: ?
 - read man cp -r --recursively = with all details within
 - cp -r 1403 1404
- tree command
- rmdir 1404 # not empty folders
- rm 1404
- rm -r 1404
 - rm -rf 1404 : force so that not to ask for each cleaning
- ; : : pwd; date; ls/booooot; hostname
 - mkdir anisa; cd anisa
- define alias : `alias pk='cp /var/log/messages /root/1403/mehr/18/m1.txt`
 - alias
 - unalias : `unalias ll
 - alias are session based unless we write it in related file to be permeant`
 - alias hmh='pwd; ls'

```
ll = ls -l
```

- alias has priority over \$PATH
- /bin/rm fl
- timedatectl
- ping
- ifconfig
- systemctl restart network
- yum install -y <package name>

- `sudo ????????`

session 5 - part 2

- `ls -l <names>`
- `ls -l <wildcard on names>`
 - `ls -l m* : *` nothing or something with any length

wildcards

- on file names
- meta characters:
 - `*`

```
cd /dev
ls -l sd*
ls -l sd?          # 3 chars
ls -l sd??         # 4 chars
ls -l sd[abcdek]    #
ls -l sd[a-ek]      #
ls -l sd[!aekrt]    #
ls -l sd{a,b,c,d,e,k} #
ls -l sd{a,b1,cdefg123} #
```

Exercise: on `anisa sample`

```
cd /anisa
ls -l
ls -l man*
ls -l man?
ls -l ????
ls -l man??

ls -l *i
ls -l [s-S]*
rm -f {s,S}*; rm -f [sS]* ; rm -f s* S*
touch majid, karim; touch /root/anisa/majid /root/anisa/karim;
ls -l *i*
ls -l ??i*
clear; pwd; cd ./var/log/
mkdir /opt/test
mv /root/anisa/??[0-9]* ./opt/test/
```

- 4th way to make file: use text editor

- nano, emacs
- gedit, kedit, nedit, xed,
- vi, vitiny

vi

- mode of operations
 - command mode \longleftrightarrow unten i for Insertoben Esc - insert (edit) mode
- `vi /opt/f1, /opt/f1`
- `vi f1 f2, f3`

```
cd ~
ls
vi batman.rrr
h j k l
i
Hesam
Esc.
:w \n = Enter
:q \n = Enter

:wq

vi batman.rrr
Mohammad Hosseini
099352002331
:wq
```

SHELL

- command
- `i h j k l`
- `:`

```
cd ~
dmesg > f1
vi f1

:set nu # set number
:185 # go to line 185
: # movment in
# <--h j k^ l-->
: # movment in <--h 4j 2k^ 8l-->
#w word forward 6w
#b word backward 2b

#H head line screan
#L Last line screan
```

SHELL

```
#gg = :1 last line of file

#G last line of file
ctrl+f 1 page forward (down)
ctrl+b 1 page backward (up)
:55
i 8 space
Esc.
---> ^ caret start of line
      0 start of line
      $ end of line

:      # movment in <--h j k^ l-->
::
```

```
ls - lhtr #?
```

SHELL

regular expression

- More capable for working on contents file

Session 6

- on using `vi`

```
demsg >f1
vi f1

:99
----> go to middle of line
### deletion from text
x 8x # delete from right of cursur
X 9X # delete from left of cursur

s Xi # delete 1 character from right and go to Insert mode
6s =
S 3S + insert mode # delete line and go to insert mode
### cut from text
d6w # d + number + dirction
d9l
d7w
d5j
d$ --> D
dG

...
dd # cut one line
```

```

5dd
...
cd6w  # c + number + direction + go to Insert mode
c9l
c7w
c5j
c$ --> C
cG

....
cc  # cut one line + go to Insert mode
5cc # cut one 5 line + go to Insert mode
### copy from text
Yank = copy
y6w  # y + number + direction
y9l
y7w
y5j
y$ --> Y
yG

....
yy # copy one line
5yy


p # paste right of  - 6p
P # paste left of


u # (undo)
ctrl+r # (redo)
. # redo the last command


#### searching in manual
/word + enter # go from location downward
?word + enter # go from location upward and from start to location
n # next match in direction
N # next match in anti direction


i # switch to Insert mode + start add from location to the right
a # switch to Insert mode + start add from location to the left
I # switch to Insert mode + cursor go to start of line
A # switch to Insert mode + cursor go to end of line
o # switch to Insert mode + **open** (add) new line after cursor
O # switch to Insert mode + **open** (add) new line before cursor
r # switch to Insert mode + replace 1 character + switch to command mode
R # switch to Insert mode + replace Mode
Insert or Ins. # Switch between Insert mode and replace mode
#### : File / host command mode
:w + Enter # save file if name is given
:w + name and address file # save file
vi f1 f2 f3 f4
:n # go to next file
:e + name and address file # open new file
:r + name and address file # copy other file inside what you are in

```

```

:q # quit vi if you saved
:wq # to save and exit
:q! # to force ignoring changes

:n! #
:e!
:w! # save for file you are not owner

:wq! # not correct # not needed to force
:wq = :x = ZZ # save and exit
:w!q # correct

#### work on sample @ 39:00
:set nu # line number
:180
y31
6p

/pci # search downward
      # n next, N the one before

?PCI # search in last direction we worked
      # n , N switch direction

vi numbers myinfo

:w # save file
# save as continue in old file in Linux # unlike Windows

```

if we close `vi` and open new `vi`, what we copy or cut will not be accessible any more.

- old process of `vi`
- new process of `vi`
- open 2 files if we know in advance
- `:e` to open other files and do what we need like past
- decide on action on newly opened like paste and move on
- `:r`
 - `ubuntu` and `Mint` - copied in `common RAM` part
 - open temporary `subshell`

```

:!command # to do bash command in vi

      # Enter to go back

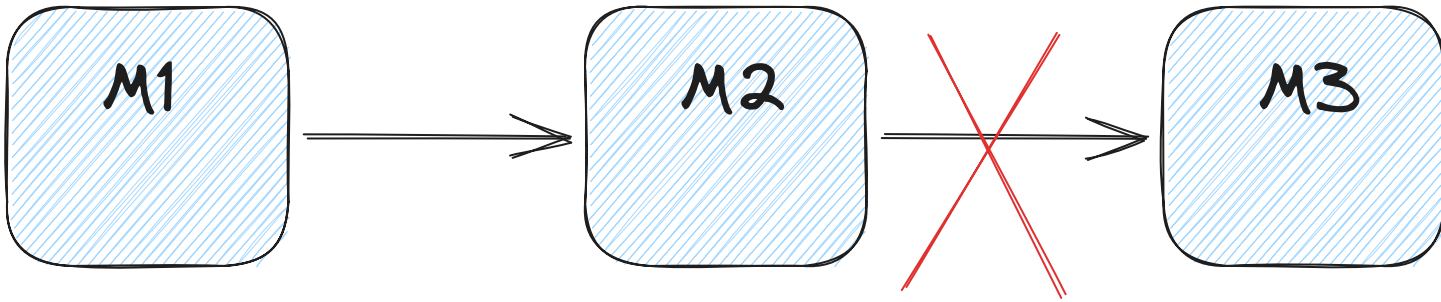
```

processes and Subshell

Working with few machines in a queue

SSH or TELNET connecting to next Machines

- what happens if we exit from last?
- what happens if network connection somewhere disconnect? Machine before that



```
bash
--- ls YES
    --- cp ? NO
        # while copy in progress: no place to do this
        # wait until task done and shell allow us
        --- pwd ?

ps # show processes
```

- is it possible to run `bash` inside `bash` ? Yes

```
bash
|___ bash # subshell of above
    --- pwd
    --- zsh
        --- csh
            --- ksh
                --- pwd
                --- bash
                # if exit, go 1 step back
                # if KILLED, to pattern of where it was
```

```
ps # show
```

- `process ID` = `PID` of `ps` changes every time as `ps` task is 2 print exiting processes and terminate
- `PID` of `bash` is same. Interactive processes are like this = endless life time processes.

```

File Edit View Search Terminal Help
[root@kashani3 ~]# ps
  PID TTY          TIME CMD
 2404 pts/0        00:00:00 bash
 9526 pts/0        00:00:00 ps
[root@kashani3 ~]#
[root@kashani3 ~]#
[root@kashani3 ~]# ps
  PID TTY          TIME CMD
 2404 pts/0        00:00:00 bash
 9536 pts/0        00:00:00 ps
[root@kashani3 ~]#
[root@kashani3 ~]#
[root@kashani3 ~]# ps
  PID TTY          TIME CMD
 2404 pts/0        00:00:00 bash
 9554 pts/0        00:00:00 ps
[root@kashani3 ~]# █

```

- `exit` or `Ctrl+D` (left Ctrl)

```

bash
|__ vi
    --- :! pwd # temporary subshell is opened
    --- :! --- bash # if more than 1 line bash command is needed
                  # line of commands

```

7 notes

LPIC 1 concepts

1. `Linux` assign a number to each `file` , called `inode#`
2. `Linux` assign a number to each `user` , called `UID = User Identity`
3. `Linux` assign a number to each `Group` , called `GID = Group Identity`
4. `Linux` assign a number to each `Process` , called `PID = Process Identity`

LPIC 2 concepts

1. `mail server` by default at any `Linux` . For internal email between users of this `Linux`
2. `router` by default (software) . routing table . Off by default
 - command to start
3. `firewall` : Kernel itself handle this functionality.
 - Interfaces for configuring firewall = to write rules : `firewalld` , `ufw` , `ipchains` , `iptables` , etc

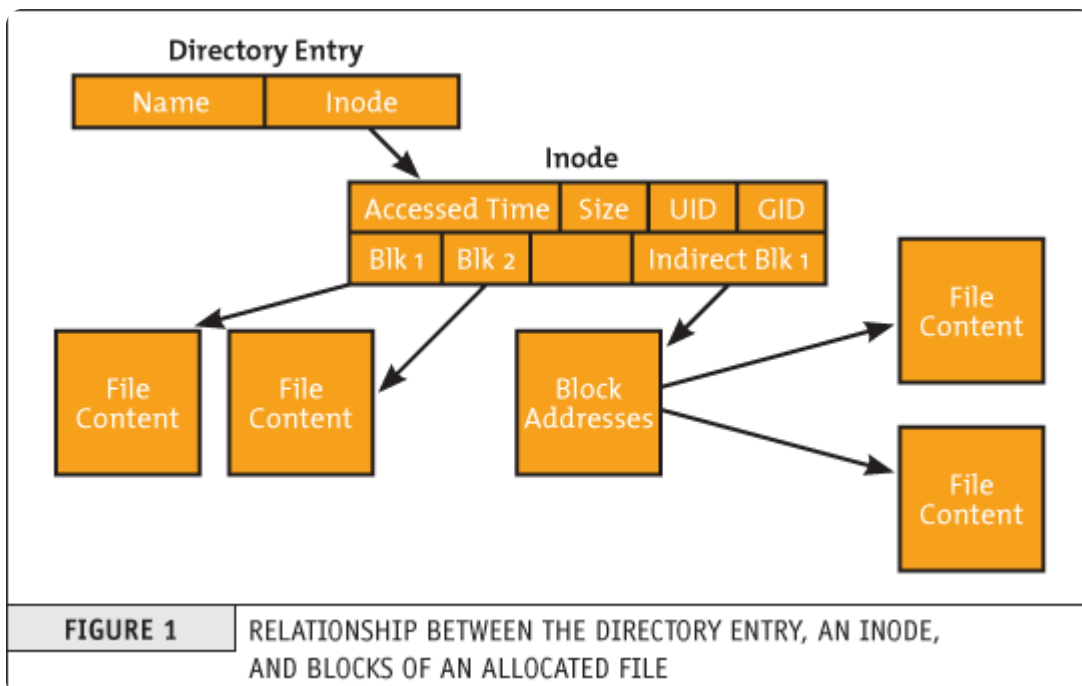
LPIC 1 concepts details

1. `inode#`

```
ls -li # to watch inode,
      # random number allocation to same file
ls -li
```

- **inode table** when disk/partition is formatted
 - contains more columns more on inode

inode#	filename
1	
2	write related file name
200_000_000	

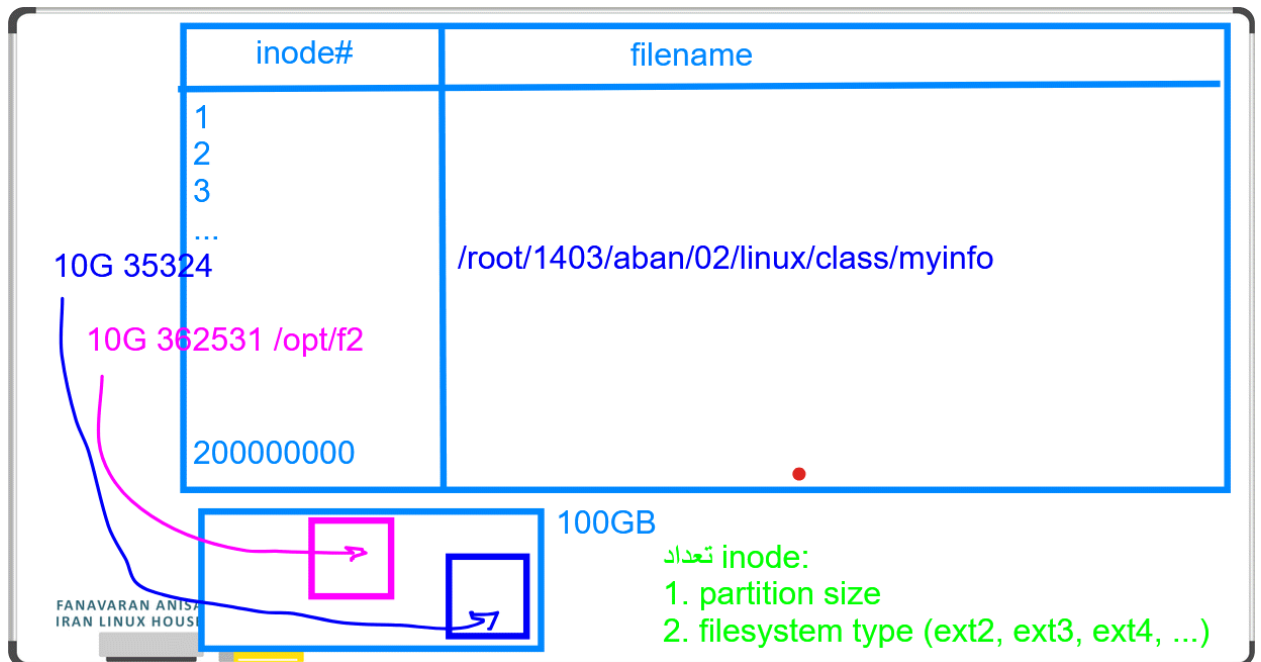


- number of **inode** is function of
 1. **partition size**
 2. type of **file system**
- What is the maximum number of files? = number of **inode** s
 - if small size files and **inode** finished, no other file can be made even if half of hard is empty. Like Car plate پلاک
 - block size is something else
 - delete file → make **inode** free

```
inode    size    address
35324    10G    /root/1403/aba/02/linux/class/myinfo
53297547 2K      /opt/f1
```

- how commands work in backend
 - **cp** - make a new file at same folder or other folder

- 2 different file at the end of copy process
- `rm` remove file -
 - in graphical interface - only pointer to file is deleted
 - pointer to file deleted
 - filename in table will be deleted
 - summary of changes write in temp [recyclebin??]
 - Hard is not cleaned and space is not freed - `restore` in `trash`



- `rm` in `command mode`
 - pointer to file deleted
 - filename in table related row will be deleted
 - no information be stored
 - invalid marker on file in Harddisk → space free up
 - recovery from backup - depends on your actions on partition
 - recovery tool capabilities: `testdisk`
- `mv`
 - filename in table be updated
 - no physically change on file
 - `mv` to `/boot` → `inode` will be changed, why?
 - since partition is changed: *each partition has its own inode table*

```
cd /dev
ls -l sd* # try to understand

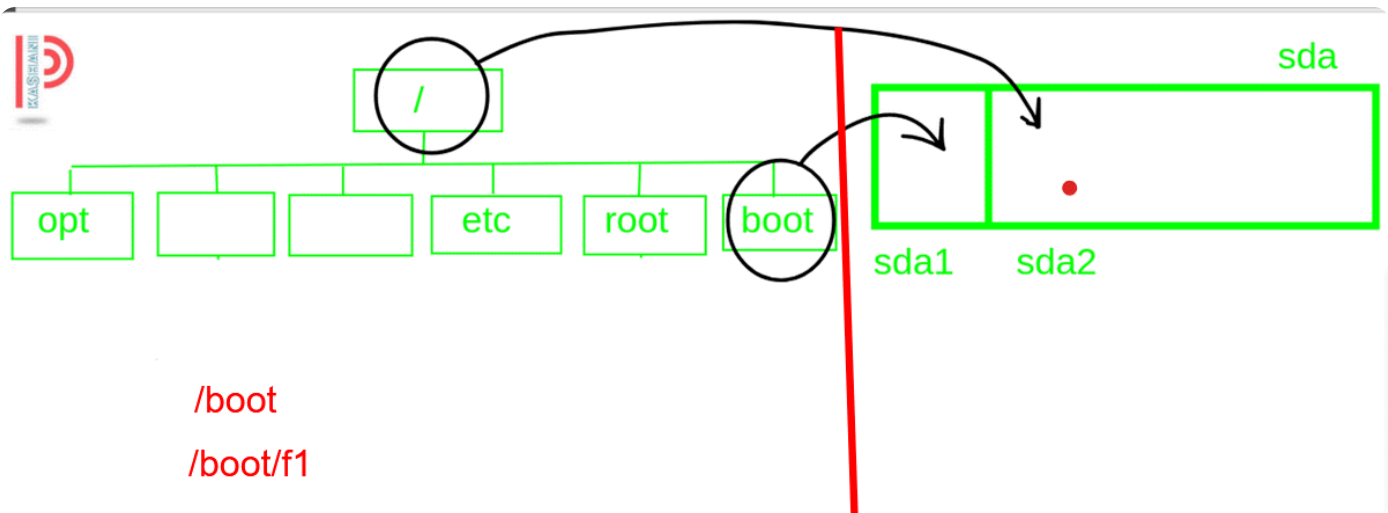
ls -l sda*

df -h
```

- disk partition


```
[root@kashani3 dev]# ls -l sda*
brw-rw---- 1 root disk 8, 0 Oct 21 16:52 sda
brw-rw---- 1 root disk 8, 1 Oct 21 16:52 sda1
brw-rw---- 1 root disk 8, 2 Oct 21 16:52 sda2
[root@kashani3 dev]#
[root@kashani3 dev]# df -h
Filesystem                Size      Used Avail Use% Mounted on
/dev/mapper/centos-root    18G        6.2G    12G   36% /
devtmpfs                   992M          0   992M    0% /dev
tmpfs                      1001M      140K    1001M    1% /dev/shm
tmpfs                      1001M       8.7M    992M    1% /run
tmpfs                      1001M          0   1001M    0% /sys/fs/cgroup
/dev/sda1                  497M       106M    391M   22% /boot
[root@kashani3 dev]#
```

- logical vs physical partition understanding



- mount point** - 2:20:00
 - a directory in logical domain (addressing domain) which is connected to partition in physical domain is called a **mount point**.
 - connecting is called mounting. disconnecting = unmounting
 - hard with **n** partition needs **n** **mount point**
 - mount** command - make connection
 - umount** first and then do action like
 - only these could be moved **/boot, /root, /home, /opt, /temp, /var**
 - /bin, /etc, /sbin, /lib** should not be
 - if **/var** moved, services might be affected

On Hard-disk and partitioning

- magnetic disk → track → sector → bit (magnetic bipolar stuff which can store 1 bit of data)
- how to use a new hard? stranded partitioning = fixed size
 - cable connection (data) + power
 - partitioning → **fdisk**
 - choose filesystem
 - format → **mkfs**

- (example: ext4 → Block size (4KB))
 - at the end of format we have `inode table`
5. mount: make a directory and point it to partition → `mount != umount`
6. to view → `df -hT`
- hard devices type
 - IDE (PATA =Parallel ATA) - 2 cable each with 2 free port = 4 port.
IDE cables - connectors (primary, secondary) -
power cable
jumpers to mention which hard to be - or based on sequence of connecting to slots
 - `/dev/hda`
`hdb`
`hdc`
`hdd`
 - SCSI → card on motherboard - port # 8-1=7 or 16 -1= 15
 - `/dev/sda`
`sdb` , `sdc` , `sdd` , ...
 - SATA = Serial ATA
 - `/dev/sda`
`sdb` , `sdc` , `sdd` , ...
 - cool disk = cool memory = pen memory = memory stick = flash memory
USB connected, plug&play
 - `/dev/sda`
`sdb` , `sdc` , `sdd` , ...
 - If system has SCSI, SATA and cool disk, first SCSI labeled and then SATA and finally cool disk.
 - SSD hard
1. SATA - technologically is SATA
- `/dev/sda`
`sdb`
2. NVMe
- `/dev/nvme0n1`
`nvme1n1`
`nvme2n1`
- SD cards: mini/micro SD
 - `/dev/mmcblk0`
`mmcblk1` , ...

Why we partition?

- segregate `system data` from `user/services/projects data`
 - encrypt on partition
 - make read-only for backup
- speed in recovery

- access management - DB admin to only access his related partition (part)
- differentiation between services
- different file system on partition per service needs
- back up
 - data
 - block → whole partition blocks to be copied
- Low-level backup = [image](#)
- LVM = Logical volume ?????????????? [3:27:00](#)
- dual-boot capability

Session 7

```

root@hes:~# df -h
Filesystem                Size      Used Avail Use% Mounted on
tmpfs                    197M        1.2M  196M   1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv 8.1G       7.2G   448M  95% /
tmpfs                    985M          0  985M   0% /dev/shm
tmpfs                    5.0M          0   5.0M   0% /run/lock
/dev/sda2                1.7G       94M   1.5G   6% /boot
tmpfs                    197M       12K   197M   1% /run/user/0
Linux_shared             327G      312G   15G  96% /media/sf_Linux_shared

root@hes:~# ls -l sd
ls: cannot access 'sd': No such file or directory
root@hes:~# ls -l
total 8
-rw----- 1 root root 391 Oct 28 15:24 50-cloud-initbackup.yaml
drwx----- 3 root root 4096 Oct 30 06:48 snap
root@hes:~# find / -name "sd*"

root@hes:~# cd /dev
root@hes:/dev# ls -l s*
brw-rw---- 1 root disk 8, 0 Oct 30 06:48 sda
brw-rw---- 1 root disk 8, 1 Oct 30 06:48 sda1
brw-rw---- 1 root disk 8, 2 Oct 30 06:48 sda2
brw-rw---- 1 root disk 8, 3 Oct 30 06:48 sda3
crw-rw---- 1 root cdrom 21, 0 Oct 30 06:48 sg0
crw-rw---- 1 root disk 21, 1 Oct 30 06:48 sg1
crw----- 1 root root 10, 231 Oct 30 06:48 snapshot
brw-rw---- 1 root cdrom 11, 0 Oct 30 06:48 sr0
lrwxrwxrwx 1 root root 15 Oct 30 06:04 stderr -> /proc/self/fd/2
lrwxrwxrwx 1 root root 15 Oct 30 06:04 stdin -> /proc/self/fd/0
lrwxrwxrwx 1 root root 15 Oct 30 06:04 stdout -> /proc/self/fd/1

shm:
total 0

```

```

snd:
total 0
drwxr-xr-x 2 root root      60 Oct 30 06:05 by-path
crw-rw---- 1 root audio 116,  5 Oct 30 06:48 controlC0
crw-rw---- 1 root audio 116,  3 Oct 30 06:48 pcmC0D0c
crw-rw---- 1 root audio 116,  2 Oct 30 06:48 pcmC0D0p
crw-rw---- 1 root audio 116,  4 Oct 30 06:48 pcmC0D1c
crw-rw---- 1 root audio 116,  1 Oct 30 06:05 seq
crw-rw---- 1 root audio 116, 33 Oct 30 06:48 timer

```

Methods of partitioning

- **MBR scheme** : master boot record = first 512 byte of Disk, (other names **MSDOS style**, **MSDOS**, **DOS**)
 - only 4 primary is possible
 - extended to 16 then 64
 - size limited for each to 2 TB
- **GPT scheme** : GUID partition Table
 - 128 partition
 - larger size till 9.5 ZB

Both limited, but differences:

1. 4 partition vs 128 primary partition
change 1 extended to logical **min12:00**
2. ????

MBR scheme

- Extended Partition will be used to make Logical partitions

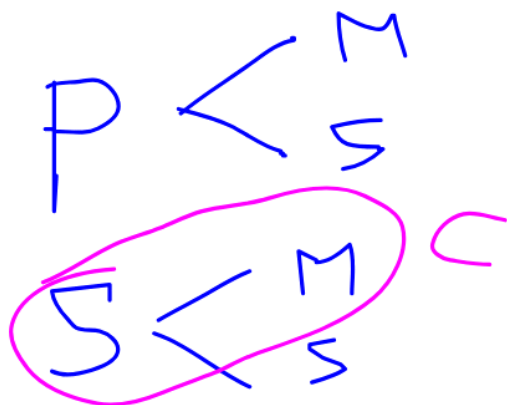
Primary Partition	Extended Partition	Logical Partition
Max 4	n/a	n/a
Max 3	Max 1	Max 60

- example: What is partition name connected to Secondary Master → IDE
 - **hdc**

hd.

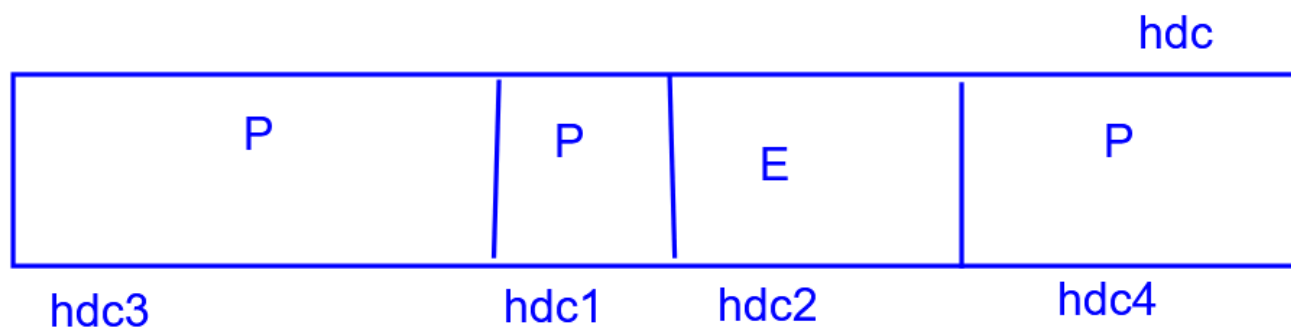


sdb



FANAVARAN ANISA
IRAN LINUX HOUSE

- partition name on this hard
 - add number to partition name - `hdc1` , `hdc2` ,

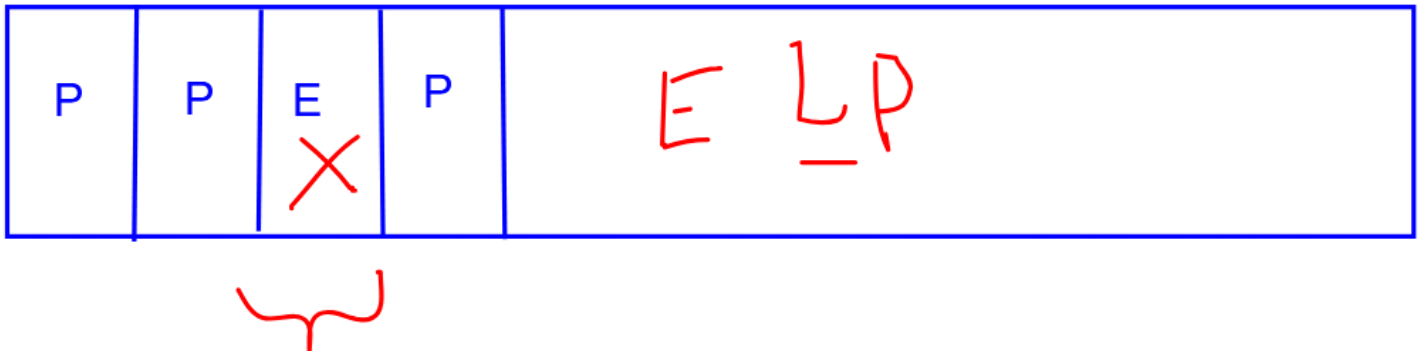


- example: we have 3 IDE, 2 SCSI, 1 SATA and 1 flash. What is flash name?
SCSI, SATA and flash: `sd#`
 - we have 4 then name is `sdd`
- 1 P → `sdd1`
- can we make Disk as 1 primary partition? Yes
- can we make Disk as 1 extended partition? Yes. One primary was needed to be possible to be booted from 30 years ago. if it was 2nd hard.
- partition identified by sector or cylinder number. start or end is not important
- SAS hard on enterprise servers.
 - SATA with 10000 rpm ??
- is this possible? Yes
 - what happened to end of Disk? wasted - unallocated = free space
 - maximum count limit reached (one dimension of limitation) - max Disk space not met

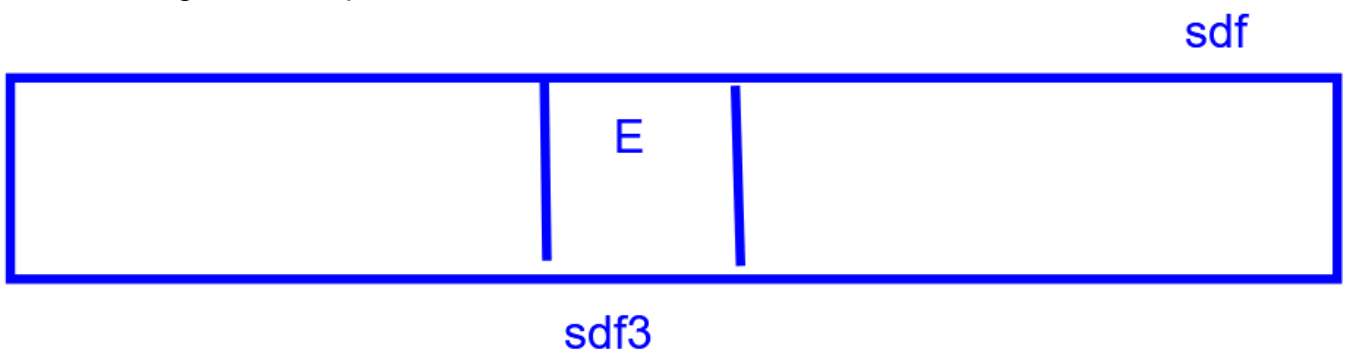


→

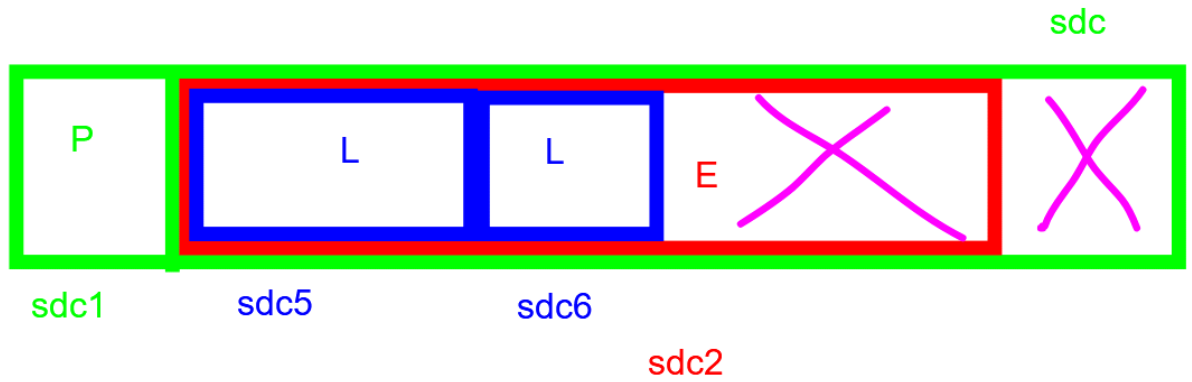
whole data will be lost ????



- example: 6th SCSI hard. what is name? `sdf`
 - can label not starting from 1.
 - Normal admin will not do such.
 - start from start of Disk
 - name starting from 1 as partition number



- example:
 - logical inside `extended`
 - the first logical should be `5` and should be name in order
 - last logical number = 64
 - first `Logical` on `Extended` labeled 5
 - then we can have till 64 - on `sdc` hard for example `sdc64`

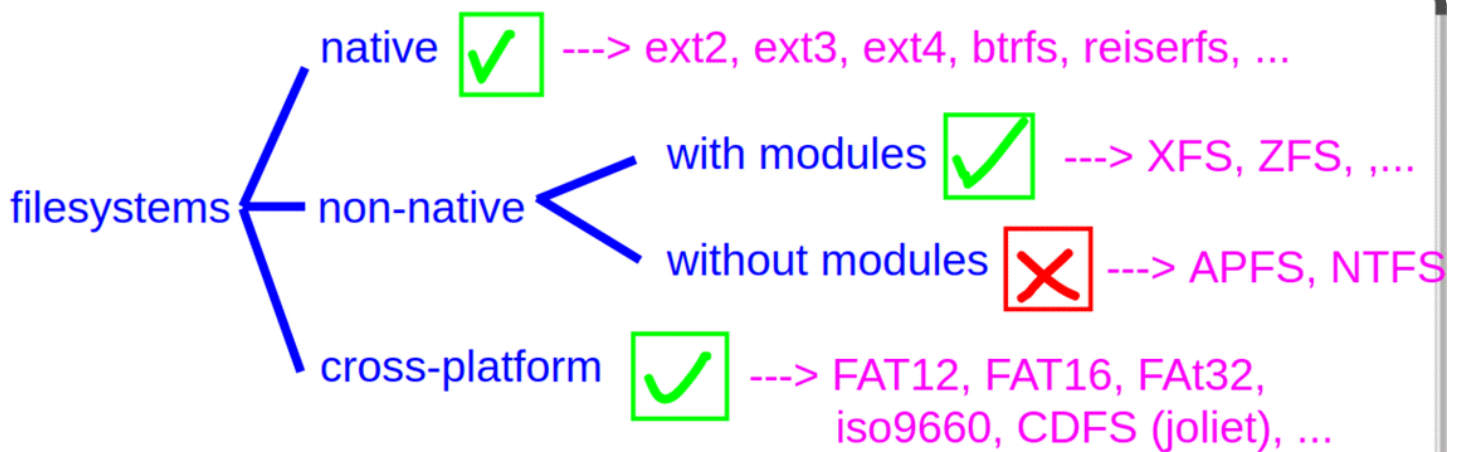


↺

```
root@hes:/dev# ls -l /dev/sd*
brw-rw---- 1 root disk 8, 0 Oct 30 06:48 /dev/sda
brw-rw---- 1 root disk 8, 1 Oct 30 06:48 /dev/sda1
brw-rw---- 1 root disk 8, 2 Oct 30 06:48 /dev/sda2
brw-rw---- 1 root disk 8, 3 Oct 30 06:48 /dev/sda3
root@hes:/dev# ^C
root@hes:/dev# ls -l cdrom
lrwxrwxrwx 1 root root 3 Oct 30 06:05 cdrom -> sr0
```

↺

filesystems



- small partition $\leq 2TB <$ large partition

the extended filesystem \rightarrow `extfs` \rightarrow `ext`

- `ext` - linux native - 1992
 - with coming of `ext2` in few months, in early 1993, deprecated.

- max partition size: 2GB - max single file size : 2GB
- `ext2`
 - linux native - 1993. similar to first generation with main distinction.
 - not have `journal` functionality. hence, if `crashed` , `recovery` is slow and mostly not successful.
 - max partition size: 32TB - max single file size : 2TB
- `ext3`
 - linux native - 1999. similar to `ext2` with only adding `journal` functionality.
 - hence, if `crashed` , `recovery` is slow and mostly not successful.
 - max partition size: 32TB - max single file size : 2TB
- what is `journal` ?
 - metadata gathered from not closed and files in use are called.
 - application of `journal` : is for `recovery` after `crash` . This metadata is store with different method in different filesystems.
 - save and decide / store unsaved things
 - `crash` : edited, open changed not saved, ... like electricity : clean vs dirty. consistent vs not consistent.
 - if `crashed` , `recovery` is slow and mostly is successful.
- `ext4`
 - Linux native - 2006. similar to older generation, but have diverse changes with enhancement. enhanced for huge size.
 - max partition size: 1 EB = 1,000,000 TB - max single file size : 16 TB
-
- Use `ext4` for personal and enterprise
- `btrfs`
 - advanced Linux native by Oracle- 2007. similar to XFS and ext4, good for huge size partitions on very huge Disk.
 - `time shifting` functionality
 - max partition size: 16 EB = 16,000 PB - max single file size : 16 EB
 - Oracle concept: `table space` : logical -> 1 or some file
 - big table space
- `reiserfs`
 - Linux native by German guy called Reiser- 2001. similar to XFS and ext4, good for huge size. journal functionality.
 - million small files of like 1 KB
 - max partition size: 16 TB - max single file size : 8 TB
 - versions
- `vFAT` virtual File Allocation Table (FAT)
 - Linux knows them as Virtual FAT, including FAT12, FAT16, FAT32, etc.
 - FAT12: max partition size: 32 MB- max single file size : 32 MB
 - FAT16: max partition size: 4 GB - max single file size : 2 GB
 - FAT32: max partition size: 8 TB - max single file size : 4 GB
- rest of cross-platform like `iso9660`, `CDFS (joliet)`

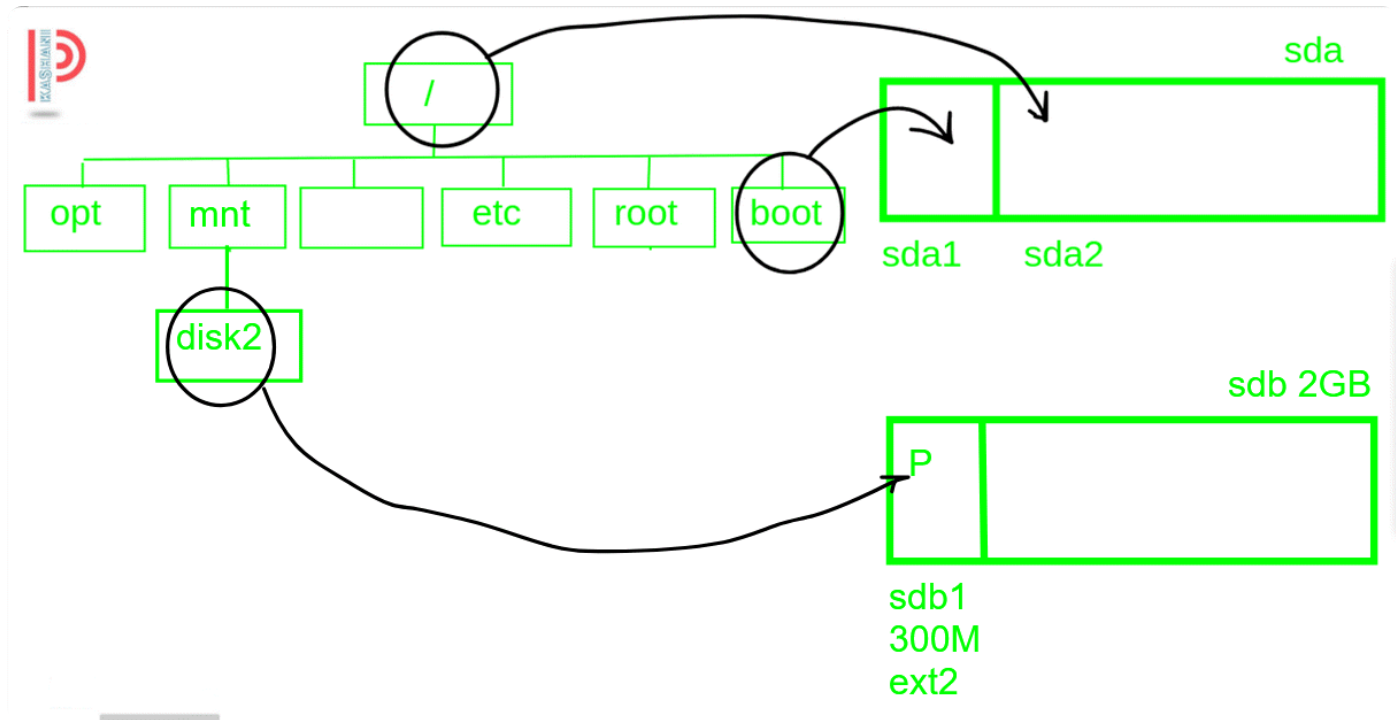
- All are fine for using in CD, DVD, etc. Almost similar and have minor differentiation.
- **swap**
 - part of Linux hard file system, considered as different partition with **swap** file system. useful in systems where RAM might become limited.
 - whole programs stays on RAM for working.
 - IF RAM become limited, **swap** become temporary location
- **NFS** network file system
 - sometime in our system **mountpoint** which seems to be local but data are store on other Linux hard in network stored.
 - NFS refer to 3 things:
 - name of service network file sharing
 - file system of NFS
 - NFS protocol
 - NFS service on other location and make mount point for it in local Disk.
- How Windows do this?
 - file sharing protocol **CIFS** old → **SMB**
 - **SIMBA** to connect Windows to Linux
- **proc**
 - semi filesystem with storing footprints of kernel in filesystem
 - where to see them?
 - **/proc** and **/sys**
- What is definition of **FILE SYSTEM** ?
 - A **file system** is a **system** , parts which are working together, which **manage**
 - **storing** and **recovery** of files in disk.

Comparison_of_file_systems

filesystem	year	max partition size	max file size
ext ext2 ext3 ext4 btrfs reiserfs	1992	2 GB	2 GB
NTFS	1993	16 EB	16 EB
ReFS	2012	3.76 ZB	16 EB
exFAT	2006	64 ZB	16 EB
ZFS	2004	256 ZB	16 EB

- practical numbers may differ

- per need and use case we have to investigate and decide
- Mounting logic explanation - start session 2 - continue example



- under `/mnt/disk2`
- add `SATA`
 - `VMDK` (Virtual Machine Disk) and `dynamically allocated` - `sdb`

```
ls -l /dev/sd* # to get list sd? on /dev/
fdisk /dev/sdb # provide fdisk possibilities
    # n new
        # Partition type: p primary, e extended
        # Partiotion number (1-4, default 1)
        # First sector (2048 - 4194303 , default 2048):
        # Last sector , +sectors or +size{K, M, G} (2048 - 4194303 ,
        # MiBi Byte - hardware 1000, software 1024
        # Id 83 = native Linux file system
    # p print
    # d delete
    # w write table to disk and exit
    # if it can not write in filesystem MBR, ask for reboot
init 6 # reboot is needed
mkfs -t ext2 /dev/sdb1 # -t type selection
    # -t ext2
```

- review what reported at the end

```
[root@kashani3 ~]# mkfs -t ext2 /dev/sdb1
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
Stride=0 blocks, Stripe width=0 blocks
76912 inodes, 307200 blocks
15360 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=67633152
38 block groups
8192 blocks per group, 8192 fragments per group
2024 inodes per group
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729, 204801, 221185

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done
```

- notes
 - software vs hardware size difference
 - 5% of storage reserved for supper user
 - `df` only show `mounted` items
 - Till now, it is made but not mounted to address so that to be accessible
 - unmount `umount` physical or logical address
 - we need to be out of place to unmount!

```
cd /mnt #
ls
mkdir /mnt/disk2 # empty forlder
cd disk2/
cd .. # during mount and umount time, we should not be in folder
mount /dev/sdb1 -t ext2 /mnt/disk2/
echo $? # check what returned? 0 = done
df -h # or check
pwd
cd dick2/
ls # not empty any more
# lost+found dirctory
vi primary.txt # -> 1403/08/09\nThis is my PRIMARY partition.
cat primary.txt
# history command
```

- do same in my machine as sample
 - note the order of steps and switches
- unmounting

```
umount <> # Physical address or Logical address (Mounted on)
# if in location: target is busy
```

- get Type of file system in `df - T = Type` column added

```

root@hes:~# df -hT
Filesystem                                Type      Size  Used Avail Use% Mounted on
tmpfs                                     tmpfs     197M   1.1M  196M   1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv ext4       8.1G   7.6G   33M 100% /
tmpfs                                     tmpfs     985M     0   985M   0% /dev/shm
tmpfs                                     tmpfs      5.0M     0    5.0M   0% /run/lock
/dev/sda2                                ext4       1.7G    94M   1.5G   6% /boot
Linux_shared                             vboxsf    327G  299G    28G  92% /media/sf_Linux_shared
tmpfs                                     tmpfs     197M    12K   197M   1% /run/user/0

```

Linux books

Folder address : `D:\Hesam\Data Science\Apache Spark\apache-resource\Books\Linux`

- Wiley Linux.Bible.9th.Edition.pdf
- Wiley Linux Command Line and Shell.Scripting.Bible.3rd.Edition.pdf

What is `$?` in Bash?

- `$?` is a special parameter in Bash that holds the `exit status of the most recently executed foreground pipeline`.
- It expands to the decimal exit status of the last executed command.
- An exit status of `0` indicates success, while non-zero values indicate various types of errors or failures.

Key Points:

- `$?` is typically used immediately after running a command to check its exit status.
- It can be used in conditional statements like `if() { }` to check if a command succeeded or failed.
- The exit status is usually a small integer value, often 0 for success and 1-255 for various error conditions.
- `$?` is reset to 0 after each successful command execution.



- **Exercise for next time:**

we know that with `df -hT` for mounted. How we get similar information for unmounted items?

```

root@hes:~# df -hT
Filesystem                                Type      Size  Used Avail Use% Mounted on

```

SHELL

tmpfs	tmpfs	197M	1.1M	196M	1%	/run
/dev/mapper/ubuntu--vg-ubuntu--lv	ext4	8.1G	7.2G	444M	95%	/
tmpfs	tmpfs	985M	0	985M	0%	/dev/shm
tmpfs	tmpfs	5.0M	0	5.0M	0%	/run/lock
/dev/sda2	ext4	1.7G	94M	1.5G	6%	/boot
Linux_shared	vboxsf	327G	313G	14G	96%	/media/sf_Linux_shared
tmpfs	tmpfs	197M	12K	197M	1%	/run/user/0

- solution reference:

The drives on any system can either be mounted or unmounted. The mounted drives are the ones that are ready to be accessed at any time whereas the data residing on the unmounted drives can only be accessed after these drives are mounted.

1. `sudo fdisk -l`

```

root@hes:~# fdisk -l
Disk /dev/loop0: 4 KiB, 4096 bytes, 8 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: 73.88 MiB, 77463552 bytes, 151296 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/loop2: 272.11 MiB, 285323264 bytes, 557272 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/loop3: 505.09 MiB, 529625088 bytes, 1034424 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/loop4: 91.69 MiB, 96141312 bytes, 187776 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/loop5: 38.83 MiB, 40714240 bytes, 79520 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/sda: 10 GiB, 10737418240 bytes, 20971520 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 8BDAD3B2-0B2C-4563-9AAB-442A602539CC
```

Device	Start	End	Sectors	Size	Type
/dev/sda1	2048	4095	2048	1M	BIOS boot
/dev/sda2	4096	3674111	3670016	1.8G	Linux filesystem
/dev/sda3	3674112	20969471	17295360	8.2G	Linux filesystem

```
Disk /dev/sdb: 1 GiB, 1073741824 bytes, 2097152 sectors
Disk model: VBOX HARDDISK
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/mapper/ubuntu--vg-ubuntu--lv: 8.25 GiB, 8854175744 bytes, 17293312 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
```

2. `sudo blkid`

```
root@hes:~# sudo blkid
/dev/sr0: BLOCK_SIZE="2048" UUID="2024-01-11-12-47-49-66" LABEL="VBox_GAs_6.1.50" TYPE="iso
/dev/mapper/ubuntu--vg-ubuntu--lv: UUID="bbe08565-5d30-42a8-bb84-2aecebbbee7eb" BLOCK_SIZE="
/dev/sda2: UUID="ecb84121-5aa7-4b01-8a6d-b4433dc5e7c4" BLOCK_SIZE="4096" TYPE="ext4" PARTUU
/dev/sda3: UUID="VEA5dH-S90u-ndm8-tBsu-f3bx-Nef0-jA2z2Z" TYPE="LVM2_member" PARTUUID="467a3
/dev/loop1: BLOCK_SIZE="131072" TYPE="squashfs"
/dev/loop6: BLOCK_SIZE="131072" TYPE="squashfs"
/dev/loop4: BLOCK_SIZE="131072" TYPE="squashfs"
/dev/loop2: BLOCK_SIZE="131072" TYPE="squashfs"
/dev/loop0: BLOCK_SIZE="131072" TYPE="squashfs"
/dev/sda1: PARTUUID="ca0a7439-c349-4cee-9dbf-7598c4b48a4f"
/dev/loop5: BLOCK_SIZE="131072" TYPE="squashfs"
/dev/loop3: BLOCK_SIZE="131072" TYPE="squashfs"
```

3. `lsblk`

```
root@hes:~# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0                               7:0    0    4K  1 loop /snap/bare/5
loop1                               7:1    0  73.9M  1 loop /snap/core22/1663
loop2                               7:2    0  272.1M  1 loop /snap/firefox/5134
```

```

loop3          7:3      0 505.1M  1 loop /snap/gnome-42-2204/176
loop4          7:4      0  91.7M  1 loop /snap/gtk-common-themes/1535
loop5          7:5      0  38.8M  1 loop /snap/snapd/21759
loop6          7:6      0 273.6M  1 loop /snap/firefox/5187
sda            8:0      0    10G   0 disk
├─sda1         8:1      0     1M   0 part
├─sda2         8:2      0    1.8G   0 part /boot
└─sda3         8:3      0    8.2G   0 part
   └─ubuntu--vg-ubuntu--lv 252:0    0    8.2G   0 lvm  /
sdb            8:16     0     1G   0 disk
sr0           11:0     1  61.1M   0 rom

```

4. `sudo parted -l`

```

(base) [root@ ~]# parted -l
Model: VMware Virtual disk (scsi)
Disk /dev/sda: 85.9GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type    File system  Flags
  1      1049kB  1075MB  1074MB  primary xfs           boot
  2      1075MB  85.9GB  84.8GB  primary                lvm

Error: /dev/sdb: unrecognised disk label
Model: VMware Virtual disk (scsi)
Disk /dev/sdb: 429GB
Sector size (logical/physical): 512B/512B
Partition Table: unknown
Disk Flags:

Error: /dev/sdc: unrecognised disk label
Model: VMware Virtual disk (scsi)
Disk /dev/sdc: 268GB
Sector size (logical/physical): 512B/512B
Partition Table: unknown
Disk Flags:

Model: Linux device-mapper (linear) (dm)
Disk /dev/mapper/centos-home: 26.8GB
Sector size (logical/physical): 512B/512B
Partition Table: loop
Disk Flags:

Number  Start   End     Size    File system  Flags
  1      0.00B  26.8GB  26.8GB  xfs

Model: Linux device-mapper (linear) (dm)

```

```
:Disk /dev/mapper/centos-swap: 4295MB
Sector size (logical/physical): 512B/512B
Partition Table: loop
Disk Flags:
```

Number	Start	End	Size	File system	Flags
1	0.00B	4295MB	4295MB	linux-swap(v1)	

```
Model: Linux device-mapper (linear) (dm)
Disk /dev/mapper/centos-root: 478GB
Sector size (logical/physical): 512B/512B
Partition Table: loop
Disk Flags:
```

Number	Start	End	Size	File system	Flags
1	0.00B	478GB	478GB	xfs	



- تمرین: روی هارد دوم خود یک پارتیشن logical به سائز 200M بسازید که ext3 فرمت شود و از مسیر `/root/alaki/` در دسترس باشد.

2:59:50

Mounting Steps for reference:

- ✓ cable connection (data) + power: not needed as it already existed disk
- 2. partitioning → `fdisk`
 - need to make an `extended` first as `Logical` could be added there
 - make 1 `extended` larger than 200 MB
 - not work with `extended` - only to make it possible to `Logical`
 - `boot, format` not possible on `extended`
- 3. choose filesystem
- 4. format → `mkfs`
 - (example: `ext4` → Block size (4KB))
 - at the end of format we have `inode table`
- 5. mount: make a directory and point it to partition → `mount != umount`
- 6. to view → `df -hT`

```
fdisk Command (m for help): m
```

Help:

```
DOS (MBR)
a toggle a bootable flag
b edit nested BSD disklabel
```


c toggle the dos compatibility flag

Generic

d delete a partition
F list free unpartitioned space
l list known partition types
n add a new partition
p print the partition table
t change a partition type
v verify the partition table
i print information about a partition

Misc

m print this menu
u change display/entry units
x extra functionality (experts only)

Script

I load disk layout from sfdisk script file
O dump disk layout to sfdisk script file

Save & Exit

w write table to disk and exit
q quit without saving changes

Create a new label

g create a new empty GPT partition table
G create a new empty SGI (IRIX) partition table
o create a new empty MBR (DOS) partition table
s create a new empty Sun partition table

Command (m for help): p

Disk /dev/sdb: 1 GiB, 1073741824 bytes, 2097152 sectors

Disk model: VBOX HARDDISK

Units: sectors of 1 * 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disklabel type: dos

Disk identifier: 0xbbe4b1b0

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	1048576	1046529	511M	83	Linux
/dev/sdb2		1050624	1071103	20480	10M	5	Extended

```
ls -l /dev/sd*
```

```
fdisk /dev/sdb
```

```
p # p print the partition table
```

```
n # n new
```

```
e
```

```
2 # default = Enter for default
```

```
Last Sector +1G
```

Command (m for help): n

Partition type

p primary (1 primary, 1 extended, 2 free) #
l logical (numbered from 5)

Select (default p): l

Adding logical partition 5

First sector (1052672-1071103, default 1052672):

Last sector, +/-sectors or +/-size{K,M,G,T,P} (1052672-1071103, default 1071103): +2M

Created a new partition 5 of type 'Linux' and of size 2 MiB.

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	1048576	1046529	511M	83	Linux
/dev/sdb2		1050624	1071103	20480	10M	5	Extended
/dev/sdb5		1052672	1056767	4096	2M	83	Linux

Command (m for help): w

The partition table has been altered.

Calling ioctl() to re-read partition table.

Syncing disks.

root@hes:~# df -hT

Filesystem	Type	Size	Used	Avail	Use%	Mounted on
tmpfs	tmpfs	197M	1.1M	196M	1%	/run
/dev/mapper/ubuntu--vg-ubuntu--lv	ext4	8.1G	7.6G	33M	100%	/
tmpfs	tmpfs	985M	0	985M	0%	/dev/shm
tmpfs	tmpfs	5.0M	0	5.0M	0%	/run/lock
/dev/sda2	ext4	1.7G	94M	1.5G	6%	/boot
Linux_shared	vboxsf	327G	299G	28G	92%	/media/sf_Linux_shared
tmpfs	tmpfs	197M	12K	197M	1%	/run/user/0

reboot message

init 6

ls -l /dev/sd* # root@hes:~# ls -l /dev/sd*

```
brw-rw---- 1 root disk 8,  0 Dec 30 15:59 /dev/sda
brw-rw---- 1 root disk 8,  1 Dec 30 15:59 /dev/sda1
brw-rw---- 1 root disk 8,  2 Dec 30 15:59 /dev/sda2
brw-rw---- 1 root disk 8,  3 Dec 30 15:59 /dev/sda3
brw-rw---- 1 root disk 8, 16 Dec 30 15:59 /dev/sdb
brw-rw---- 1 root disk 8, 17 Dec 30 15:59 /dev/sdb1
brw-rw---- 1 root disk 8, 18 Dec 30 15:59 /dev/sdb2
brw-rw---- 1 root disk 8, 21 Dec 30 15:59 /dev/sdb5
```

mkfs -t ext3 /dev/sdb5 # equivalent

mkfs.ext3 /dev/sdb5

why???

root@hes:~# mksf -t ext3 /dev/sdb5

```
-bash: mkfs: command not found
```

```
pwd # ensure to be on /root
mkdir /root/alaki
df -h
mount /dev/sdb5 -t ext3 /root/alaki/
mount /dev/sdb5 /root/alaki/ # format not needed. it infer itself
df -h
cd /alaki
echo "1403/08/09
This is my LOGICAL partition." > logical.txt
cat logical.txt
```

```
root@hes:~# mkfs.
mkfs.bfs      mkfs.btrfs  mkfs.cramfs  mkfs.ext2    mkfs.ext3    mkfs.ext4    mkfs.minix

# not working in my Linux
```

- With `reboot`, partitions are not there any more
 - system does not know to were connect what
 - stored in RAM → we need to `persist`
 - `config` file store this located at `/etc`
 - `fstab` = file system table
 - need a row per partition

```
vi /etc/fstab
```

SHELL

```
root@hes:~# vi /etc/fstab
```

/etc/fstab: static file system information.

Use 'blkid' to print the universally unique identifier for a

device; this may be used with UUID= as a more robust way to name devices

that works even if disks are added and removed. See fstab(5).

/ was on /dev/ubuntu-vg/ubuntu-lv during curtin installation

/dev/disk/by-id/dm-uuid-LVM-
cm7KMhr6LDVL7GVxYSJRkfXEtZfykJkwD4OUp0fwK0bRwpPPRjvoTor5GkNQatMU / ext4 defaults 0 1

/boot was on /dev/sda2 during curtin installation

/dev/disk/by-uuid/ecb84121-5aa7-4b01-8a6d-b4433dc5e7c4 /boot ext4 defaults 0 1
/swap.img none swap sw 0 0
~
...

- 6 item with 1 space between

```
#  
# /etc/fstab  
# Created by anaconda on Sun Dec 3 03:45:10 2017  
#  
# Accessible filesystems, by reference, are maintained under '/dev/disk'  
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info  
#  
/dev/mapper/centos-root / xfs defaults 0 0  
UUID=78062850-3fda-4190-817b-91b219ff0374 /boot xfs defaults 0 0  
/dev/mapper/centos-swap swap swap defaults 0 0  
/dev/sdb1 /mnt/disk2 ext2 defaults 0 0
```

col1	col2	col3	col4	col5	col6
Physical address or	mount pount	filesystem tipe	option	dump backup	file system check
UUID or			example:ro	0 no	0 no, 1 earliest (only for /, 2 then rest
label				1 yes	

filesystem check types

- automatic → /etc/fstab
will be read at after reboot - if 1 or 2 → check first and then mount
 - manual → manual by admin
umount ... unmount to not corrupt data
fsck file system check
mount mount again
- file we write at /etc/fstab will be considered at next reboot
 - how to do without reboot?
 - mount -a = All

- [Exercise](#): how to find [Universal UniqID](#) of partitions in Linux? ??????????????

Here are the key methods to find the UUID of partitions in Linux:

```
// Using blkid
```

The `blkid` command is the most common way to retrieve UUIDs:

```
sudo blkid
```

This will display UUIDs for all partitions along with other metadata.

```
// Using lsblk
```

The `lsblk` command provides a more readable output:

```
lsblk -f
```

This shows UUIDs along with other partition information.

```
// Checking /dev/disk/by-uuid
```

Linux creates symbolic links in `/dev/disk/by-uuid` pointing to actual devices:

```
ls -l /dev/disk/by-uuid
```

This allows you to easily find the UUID for a specific partition.

```
// Using findmnt
```

To get the UUID of the root filesystem:

```
findmnt -n -o UUID /
```

```
// Key Points:
```

- These methods work for both local and remote systems
- No need for root access in most cases
- UUIDs remain consistent even if device names change

- Useful for scripting and automating system configurations

// Best Practices:

- Use UUIDs instead of device names (/dev/sdX) in fstab for reliability
- Combine with other commands like `df` or `mount` for more context
- Be aware of potential caching issues with `blkid`

By using these methods, you can reliably retrieve UUIDs for partitions in Linux, which is essential for many system administration and scripting tasks.



Session 8

missed

My location on 2:27:32 ???

Session 9

missed

My location on 2:27:32 ???

Session 10

My location on 2:27:32 ???

Session 11

14:07 joined

exercise

2. write a line of command to - part 2 -

Part on `processes` finished

User and group management

```
useradd peter
```

SHELL

1. check user is not already taken
2. assign the first existing `UID`
3. make a `GID`
4. add him to this group
 1. `opensuse` : group `users`
5. make directory of
6. ownership and access of this directory to be assigned to user
7. `/etc` ?????
8. `/var/spool/mail/peter` file which stores emails related to this user
9. etcetera
10. in `/etc/passwd` = `Linux password file` make one record for this user
 1. take a look at this file: 7 columns
11. `passwd peter`
 1. `shadowing feature` is active to make passwords invisible
 2. passwords stored at `/etc/shadow` - `SHA512`

```
guseradd edari
```

1. check user is not already taken
2. assign the first existing `UID`
3. make a `GID`
4. etc
5. a record for this group added to `/etc/group`
6. `gpasswd edari`
7. `/etc/gpasswd`

```
useradd peter
```

SHELL

difference between `ubuntu` and `centos`

exercise at class

```
groupadd it
groupadd IT
groupadd research
groupadd developers
groupadd managers
yum install -y zsh
which zsh
```

```

which -a zsh
useradd -d /my_HHH_dir ??????????????

cat /etc/

# dar group asli esm fard are naanvested

groups smith
groups

id peter
id # currnet ligind user information

```

- add smith to research group
 - `vi /etc/group`
 - `usermod -G research peter`
- `groups smith`
- `usermod -Ga managers smith` # wrong
- `usermod -aG managers smith`
- Lock/Unlock the user
 - `usermod -L smith`
 - `usermod -U smith`
 - `passwd -l smith`
 - `passwd -u smith`
- `cat /etc/passwd | grep smith`
- `usermod -s /bin/bash smith`
- `chsh` # change shell
- what are differences of `/sbin/nologin` and `/bin/false` ?
`UID` and `GID` user root: 0

in 2:55

`redhat` based: 500 `UID` and `GID` , (from 0 to 499)

`debian` based: 1000 `UID` and `GID`, from 0 to 999)

are reserved for services.

This and other default are stored at `logins.def` at ???

and can be modified

```

su - peter
- whoami
- ls
- ls -a
- cd /etc/skel/

```

SHELL

15:02 - 15:07 missed - min 1118

non-login mode ~/ .bashrc environment

```
su # means switch to root
su - zahra
```

SHELL

- when files at `~/.bash_profile` runs?
 - when login
- when files at `~/.bashrc` runs?
 - when kernel ?

```
.bash_logout
.bash_profile ~/.bash_login ???
```

SHELL

```
ll filename # ll f1
```

SHELL

```
newgrp
```

@ min 157

user management

administrative like activity in Linux: 4 person to do admin tasks

- change their `UID-GID` to `0` : not proper way
- `sudoers` - listen carefully
 - make group : myadmin
 - add persons to this group
 - limit on set of commands
-

```
history
$HISTSIZE
$HISTFILE
echo $HISTTIMEFORMAT # ???
```

SHELL

- We can assign `uid` to user home directory
- `userdel`
 - `-r`
- `groupdel`

I need `Windows` `bash` script to - monitor battery level continuously and make beep when power is less than 6%. - monitor battery level continuously and make beep when power is above than 96%.

Shadowing feature

`pwconf`

table

what happens between starting up till ?

runlevels

- BIOS (Basic Input/Output System) → Firmware
- hardware clock → CMOS battery
- software clock
-
- CHS (cylinder/head/sector)
- LBA(Logical Block Addressing)
- EFI (Extensible Firmware Interface)
- UEFI (Universal Extensible Firmware Interface) → 2005
 - 35 to 100 MB - EFI boot partition
- MBR (Master Boot Record)
512 first Byte
- 446 B (boot loader information)
- 64 B (boot I?)
- 2 B - parity check of file

Bootloaders:

Name	abb name	address	
LILO	lilo	<code>/etc/lilo.conf</code>	
Grand unified Bootloader (GRUB legacy)	grub	<code>/boot/grub/menu.lst</code>	<code>/boot/grub/grub.conf</code>
GRUB2		<code>/boot/grub/grub.cfg</code>	<code>/boot/grub2/grub.cfg</code>

- Sample Grub file
 - General section
 - Private section(s)

Kernel ring bugger

- `dmesg @ /var/log/dmesg`
-

Session 12

Missed 10 min of start

```
runlevel # show current and preivous runlevels
> N 5 # 5 Full multi user + graphical

# change runlevel
init
telinit
```

SHELL

- `rc` directories

```
init 2 # change to run level2

startx # add graphical service
```

SHELL

service (server)	application	daemon (service)
WEB	apache	httpd
SSH	openssh-server	sshd
DNS	bind	named
database	Mysql	mysqld
cache/proxy	squid	squid
		network-manager

- `cd /etc/init.d`

```
fuser 22/tcp # if provide response
/etc/init.d/ sshd status
netstats -ntulp | grep :22
netstats -ntulp | grep ssh
/etc/init.d/ sshd stop
/etc/init.d/ sshd start # new process with new pid
/etc/init.d/ sshd restart
/etc/init.d/ sshd reload # send reload signal = 1 = SIGHUP to ?
kill 1
kill -s SIGHUP 4053
```

SHELL

Symbolic Link (SL)

```
/etc/init.d/ssh      status|stop|reload|start|restart|
                    httpd
                    mysqld
```

SHELL

each service has one SL on each rc

SL Snnsshd → start

SL Knnsshd → kill

```
/etc/rc0.d/Knnssd
/etc/rc1.d/Knnssd
/etc/rc2.d/Knnssd
/etc/rc3.d/Knnssd
/etc/rc4.d/Knnssd
/etc/rc5.d/Knnssd
/etc/rc6.d/Knnssd
```

SHELL

```
K\d{1,2}name # priority with highest = 00
# sample Network is more important than SSH
```

SHELL

commands to manage ?

difference between `redhat` vs `debian` - base

- Redhat base

```
1. LSB /etc/init.d/ssh staus
2. service sshd status
3. chkconfig # command to manage things
   1. chkconfig --list crond
   2. chkconfig crond off
   3. chkconfig crond on
   4. chkconfig --level 135 crond on
----
4. systemctl
```

SHELL

- Debian

```
1. LSB /etc/init.d/ssh staus
2. service sshd status|... # recently added
3. update-rc.d # command to manage things
   1. update-rc.d crond defaults
   2. update-rc.d -f dovecot remove # mailserver
```

SHELL

```
3. update-rc.d -f dovcot stop 24 2 3 4 5 # /etc/rc[2-5].d/K24dovcot
4. ?
----
4. systemctl
```

Service management methods

- **system V (Sysvinit)** → init (CentOS 5, older) (Ubuntu 6.06, older)
service sshd stop
- **upstart** → init (CentOS 6) (Ubuntu 6.10 to 14.10)

```
initctl stop sshd
chkconfig or update-rc.d
init 0
```

SHELL

- **systemd** → init (CentOS 7 onward) (Ubuntu 15.04, newer)

```
systemctl stop sshd.service # در لحظه
systemctl stop sshd # در لحظه
systemctl start sshd # در لحظه
systemctl enable sshd # in next boot and always
systemctl disable sshd # in next boot and always
systemctl poweroff # = init 0
```

SHELL

BIOS	BIOS
MBR	MBR
B.L.	B.L.
Kernel	Kernel
init	systemd
runlevel --> service	target unit --> service unit

SHELL

```
CentOS 5: cat /etc/inittab |grep initdefault
min 97?
```

SHELL

write [unit] to make service up

systemd

types of units

1. service sshd.service = sshd
2. socket

3. target multi-user.target = multi-user
4. timer
5. mount
6. path
7. slice

/etc/systemd/system

/lib/systemd/system

```
systemctl status sshd
journalctl -xe # to check those which have issues
systemctl --failed

systemctl -list-unit --type=service
```

SHELL

```
shutdown
-h (halt, shutdown)
-r (reboot)
-c (cancel)
-f (fast boot -> NO file system check)
-F (force filesystem check -> YES file system check )
```

SHELL

package management

- default

1. Redhat Base `rpm` `.rpm` redhat package management
2. Debian Base `dpkg` `*.deb` debian package management

a-b.c.d-e.f.rpm

a = package name

b = version

c = major release (major revision)

d = minor release (minor revision)

e = build number

f = arch

```
rpm
-i install
-u (update)
-U (install/update)
-v (verbose)
-h (hash)
-e (erase)
-q (query)
-a (list all package names installed on this system)
```

SHELL

```
-f (file)
-l (list of all package files)
-c (list of all package config files)
```

```
uname -r
rpm -qa | grep ssh #
which ssh
rpm -qf /usr/bin/ssh

rpm -qa | grep chmod
rpm -qa | grep -i chmod
which chmod
rpm -qf ???

rpm -qc openssh-clients
rpm -ql openssh-server

rpm -qlzsh | wc -l
```

exercise image

```
rpm -i zsh
rpm -ivh zsh
rpm -Uvh zsh

wget <address to download>

cd /media/
mkdir DVD_DRIVE
```

SHELL

exercise image

- issue with `rpm`
 - `not installing dependencies`
- Wrapper based installation to cover above shortcomings

wrapper:

```
1. yum/dnf (Redhat Base) # Yellowdog Updater Modified
/etc/yum.conf
yum update | install | remove
2. apt (Debian Base) # Advanced Package Tools
```

SHELL

```
apt-get update
apt-get update
apt-get update
apt-get update
apt-get update
apt-get update
```

- Graphical
- sandbox
 - snap - flatpack

How to install from local

min 205

```
cd /etc/yum.repos.d/
# make local
```

SHELL

```
[LocalRepo]
name=
```

Session 13

joined at 6:50

- use `alias` for few line repetitive commands

Scripting

- shell scripting
- bash scripting

```
bash myscript
!rm # last use of rm will be repeated
chown root:it myscript
chmod +x myscript = chmod a+x myscript

myscript # will not run, why?
          # command not found, currnet dircty is not part of Linux PATH
echo $PATH
```


1. put file in one of addresses in PATH list
2. `PATH=$PATH+`
3. `PATH=; .` # do not use
4. provide path `./root/class/myclass` or `./myscript`

```
#!/bin/bash      # shebang
ls /boot > /root/class/list1
cp /root/class/list1 /root/class/list2
#cp /root/class/list1 /root/class/list2 #
```

Good script should have

1. `x` permission
2. `.sh` extension
3. `shebang` line `#!`

How to write a new C or ? program

1. `apt-get update` / `yum update`
2. `apt` gcc, g++, ...
3. `gcc <filename> -o myapp`

```
#include<stdio.h>

int main():
{
    printf("Hi there!");
    return 0
}
```

```
which python
```

bash scripting

```
expr 7+8 # 7+8
expr 7 + 8 # 15
date
sleep 4 # second
echo salam # print
echo $HOME
seq 1 10 # with steps
v1=3
v2=5
expr $v1 + $v2
```

```
v3 = `expr $v1 + $v2` # ` = $(  
  
read age  
  
ls -l /bin/top  
which top  
ls -l `which top`  
ls -l $(which top)
```

```
# /bin/bash  
echo "Start time is: `date`"  
sleep 8  
echo "End time is: `date`"
```

- read `date` command details

```
date +%R  
date
```

```
# /bin/bash  
echo "Start time is: `date    ???? `"  
sleep 8  
echo "End time is: $(date ?? )"
```

```
# /bin/bash  
echo "your 1st number is: $1"  
echo "your 2st number is: $2"  
total=`expr $1 + $2`  
echo "your total is: $total"
```

```
./myscript.sh 7 + 8
```

```
# /bin/bash  
echo -n "please enter your age: "  
read age  
total=`expr $age \* 365` # \ is scape character  
echo "your age in days is : $total"
```

exercise

get 2 numbers from user and print sum of them

```
# /bin/bash  
echo -n "please enter number 1: "
```

```
read number1
echo -n "please enter number 2: "
read number2
total=`expr $number1 \+ $number1 ` # \ is scape character
echo "sum of ? and ? is : $total"
```

```
# /bin/bash
if [ $USER = root ]
then echo "You are allowed :)"
else echo "You are not allowed :("
fi
```

67:40

```
# /bin/bash
for i in 1 2 3 4 5
do
    echo "your room number is: #$i "
done

for i in `seq 1 500`
do
    echo "your room number is: #$i "
    touch test$i.txt # make file
done
```

```
man touch
```

```
find / -mtime -2 # 2 days
```

exercise 2 min 75:00

rewrite exercise 5000 karbar with `bash`

get 2 numbers from user and print sum of them

```
paste -d "@" users domain > mailing.list
| split -200
```

min 80:00

```
#!/bin/bash
`date +%F_%T`.bck
```

```
- crond^
crontab -l --> list
crontab -r --> remove
crontab -e --> edit
```

MIN HRS DOM MOY DOW COM

15 8 * * /rrot/class/myscript

15 * * * /rrot/class/myscript # every hour at 15'

- * * * * /rrot/class/myscript # every minutes

15 17-20 * * /rrot/class/myscript # 17 till 20

15 17,20 * * /rrot/class/myscript # 17 and 20

0 sun

1 mon

2 tue

3 wed

4 thu

5 fri

6 sat

7 sun

read from end to start to understand

anacron anacron for desktops

- every 7 minutes or every 7 hours?

exercise 3

how to run script with cron every 30second

how to run script with cron every 15 min ?

how to run script with cron every 5 min ?

how to run script with cron every 7 min ?

how to run script with cron every 3 hour ?

how to run script with cron every 9 min ?

exercise 4

memorize dictation of month and days

Networking

what is computer network?

2 device connect in a medium with same protocol

OSAI layers

All people seem ti need data processing

Please do not Throw sausages Pizza Away

client Server

Application A

Presentation P

Session S

Transport T

Network N

Data Link D

Physical P

Network Interface Card (NIC) کارت شبکه

1. Ethernet -> eth0 # has priority
2. Wireless -> wlan0

eth0, eth1, ... and wlan0,

naming approaches

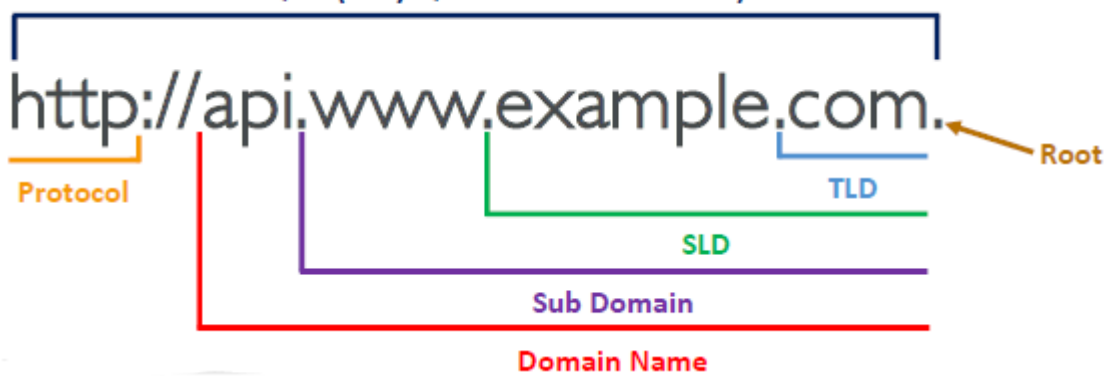
1. Hostname

- Hostname = computer_name.domain_name
www.google.com
mail.yahoo.com
kashani.lpir.org
TLD (top level domain)
- General TLD
com, net, org, gov, edu
- CC TLD
.ir, fr, ca, ...
SLD (Second level domain)
Sub-Domain Name

2. Physical address - MAC address Burned, Hardware addr

3. IP Address

FQDN (Fully Qualified Domain Name)



```
hostname
hostname test.lpir.org # changed temporary

# CentOS /etc/sysconfig/network
HOSTNAME=
# CentOS7 /etc/hostname
```

Physical address - MAC address Burned, Hardware addr
Hexadecimal
manufacturer number part number

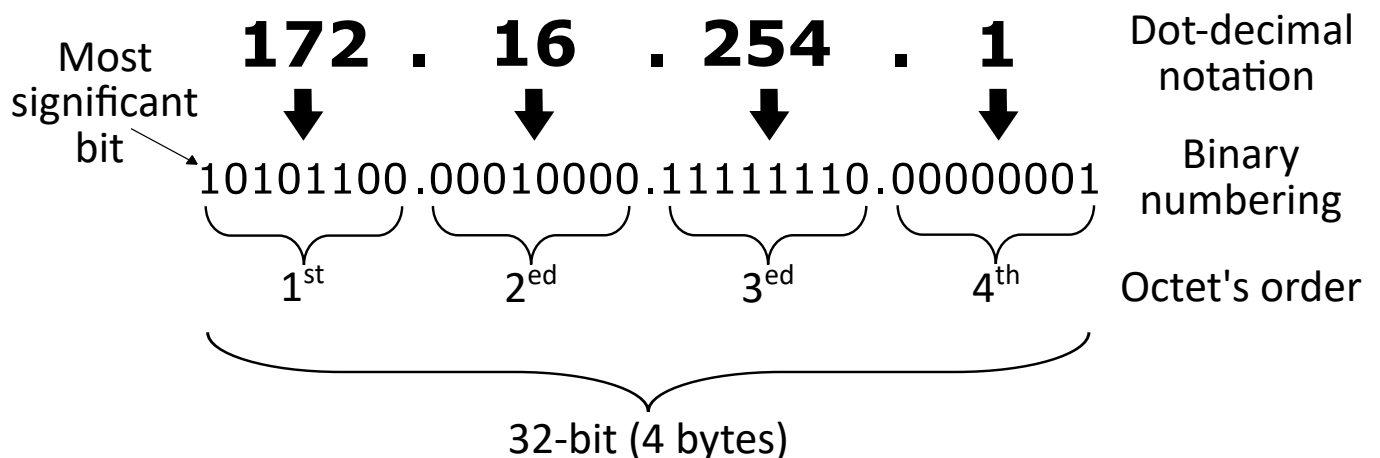
```
yum install net-tools --> ifconfig, rout, ...
    iproute --> ip

ifconfig # check
    # not possible to change MAC address phusically
```

loopback IP address 147:51

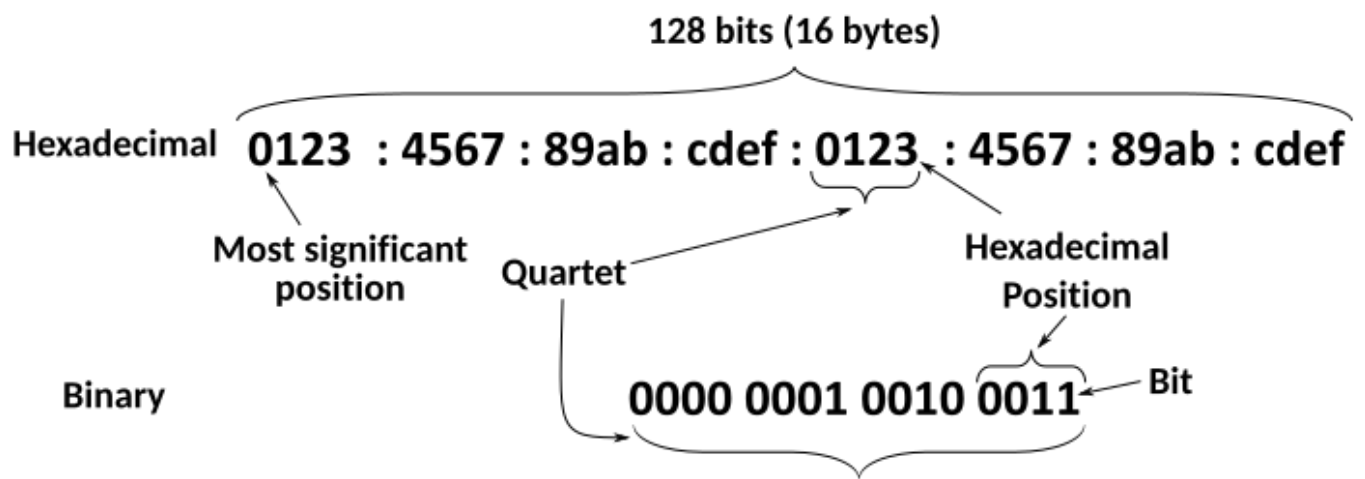
IP address

1. IPv4 : 32 bit -> $2^{32} = 4,294,967,296$



0.0.0.0 to 255.255.255.255

2. IPv6 : 128 bit -> 2^{128} = huge numbers



- switch vs router (gateway)

NAT server = router with change IP functionality

default gateway = internet

image

IP

- static
 - servers
 - clients if less than 11
- dynamic (DHCP)
 - Windows `ipconfig/release ipconfig/renew`
 - linux `dhclient (IP, gateway, dns)`

•

```
ifconfig eth0 192.168.10.11
ifconfig eth0 192.168.10.11 netmask 255.255.255.0 # (Classfull model) netmask to write netma
ifconfig eth0 192.168.10.11/24 (CIDR Model)

# /24 prefix
```

- netmask
 - differentiate `netid(network)` from `hostid(host)` # min 178
 - IP (host id) = 2^8 = 256
- 192.168.10.0 -> network interface
192.168.10.255 -> broadcast
- host count = 256 - 2

```
ifconfig eth0 192.168.10.11/28 # 16 subnets each with 16 hostids
# we need to make route to connect these
```

- Debian based

`/etc/network/interfaces`

```
auto enp0s3
iface enp03 inet dhcp

auto enp0s3
    iface enp03 inet static
    address 192.168.10.11
    netmask 255.255.255.0
    gateway 192.168.10.1
    dns-nameservers 192.168.10.2
```

- Redhat based

`/etc/network/interfaces` ??????????????????

```
auto enp0s3
iface enp03 inet dhcp

auto enp0s3
    iface enp03 inet static
    address 192.168.10.11
    netmask 255.255.255.0
    gateway 192.168.10.1
    dns-nameservers 192.168.10.2
```

```
route -n # numeric
route add default ????
```

DNS

`/etc/hosts`

```
192.168.10.11 oradb.lpir.org oradb
192.168.10.12 appsrv.lpir.org appsrv
```

```
/etc/resolve.conf
nameserver 192.168.10.1
nameserver 192.168.10.2 # go to this if upper is not proving answer
                        #
nameserver 8.8.8.8
nameserver 4.2.2.4
```

SHELL

port

- port count $2^{16} = 65536$
- 0 to 1023 = privileged - well-known ports
 - page 425 426 book
- ping protocol = Internet_Control_Message_Protocol - ICMP

```
wget http://www.google.com
```

summarizer - hash

- hash algorithm

```
cksum myfile1
md5sum myfile1
sha1sum myfile1
sha224sum myfile1
sha256sum myfile1
sha384sum myfile1
sha512sum myfile1
sha1024sum myfile1
sha32048sum myfile1
```

```
service network stop
systemctl stop network
ifconfig eth0 down = ifdown eth0
ifconfig eth0 up = ifup eth0
```

```
ifconfig
```

```
ip
ip address show
ip a show
ip a s
ip a
```

```
route -n # routing table
ip rout show
ip r s
ip s
```

```
# link means network card
```

```
/etc/sysconfig/network-scripts/
```

```
# in Rocky Linux 9
```

```
nmcli connection migrate
```

```
/etc/NetworkManager/sysyem-connections/
```

```
change file
systemctl restart
```

To change network setting

- GUI : via menu
- TUI (Text based User Interface) : `nmtui` , `ntsysv` , `nt` , `nc` , `mc` = like `nc` in Window
- CLI : `nmcli`

make up and down

picture

SSH

- port 22
- `openssh-client` `openssh-server` server_A
 `openssh-client` `openssh-server` server_B

1. remote console

```
ssh
```

2. secure copy

```
scp username_A@IP_A:/path username_B@IP_B:/path
# similar to cp
```

- Exam next week = @ 13:00

Networking in Linux

Virtualization in Linux

- **NAT**: under the management of Host for internet 192.168.167.2
- **Bridge**: Replicate the NIC state
- **Host-only**: internal Network Host without connection host
- **Internal**: between VMs Network

```
nmcli
enp0s3: unmanaged
    "Intel 82540EM"
    ethernet (e1000), 08:00:27:26:92:DD, hw, mtu 1500

enp0s8: unmanaged
```

SHELL

```
"Intel 82540EM"
```

```
ethernet (e1000), 08:00:27:6D:32:FF, hw, mtu 1500
```

```
lo: unmanaged
```

```
"lo"
```

```
loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536
```

Use "nmcli device show" to get complete information about known devices and "nmcli connection show" to get an overview on active connection profiles.

Consult nmcli(1) and nmcli-examples(7) manual pages for complete usage details.

```
root@hes:~# ip add
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
```

```
        valid_lft forever preferred_lft forever
```

```
    inet6 ::1/128 scope host noprefixroute
```

```
        valid_lft forever preferred_lft forever
```

```
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group defau
    link/ether 08:00:27:26:92:dd brd ff:ff:ff:ff:ff:ff
```

```
    inet 172.17.17.18/24 brd 172.17.17.255 scope global enp0s3
```

```
        valid_lft forever preferred_lft forever
```

```
    inet6 fe80::a00:27ff:fe26:92dd/64 scope link
```

```
        valid_lft forever preferred_lft forever
```

```
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group defau
    link/ether 08:00:27:6d:32:ff brd ff:ff:ff:ff:ff:ff
```

```
    inet 10.0.3.15/24 metric 100 brd 10.0.3.255 scope global dynamic enp0s8
```

```
        valid_lft 85446sec preferred_lft 85446sec
```

```
    inet6 fe80::a00:27ff:fe6d:32ff/64 scope link
```

```
        valid_lft forever preferred_lft forever
```

```
root@hes:~# ip route
```

```
default via 10.0.3.2 dev enp0s8 proto dhcp src 10.0.3.15 metric 100
```

```
10.0.3.0/24 dev enp0s8 proto kernel scope link src 10.0.3.15 metric 100
```

```
10.0.3.2 dev enp0s8 proto dhcp scope link src 10.0.3.15 metric 100
```

```
10.131.58.172 via 10.0.3.2 dev enp0s8 proto dhcp src 10.0.3.15 metric 100
```

```
10.132.57.138 via 10.0.3.2 dev enp0s8 proto dhcp src 10.0.3.15 metric 100
```

```
10.132.57.208 via 10.0.3.2 dev enp0s8 proto dhcp src 10.0.3.15 metric 100
```

```
172.17.17.0/24 dev enp0s3 proto kernel scope link src 172.17.17.18
```

```
root@hes:~#
```



Scheduling in Windows

Zip and unzip in Linux

```
gzip test.txt
gunzip test.txt

gzip *.txt # all .txt files
```

Using tar Utility

A **tar.gz** file is a combination of a **.tar** file and a **.gz** file. It is an archive file with several other files inside it, which is then compressed.

You can unzip these files the same way you would unzip a regular zipped file:

```
tar -xvzf documents.tar.gz
```

- **x** – instructs tar to extract the files from the zipped file
- **v** – means verbose, or to list out the files it's extracting
- **z** – instructs **tar** to decompress the files – without this, you'd have a folder full of compressed files
- **f** – tells **tar** the filename you want it to work on

To list the contents of a **.tar** file before you extract it, enter:

```
tar -tzf documents.tar.gz
```

To instruct **tar** to put the extracted unzipped files into a specific directory, enter:

```
tar -xvzf documents.tar.gz -C /home/user/destination
```

```
du -a | sort -n -r | head -n 30 # disk usage on server :
df -h
dmiencode -t memory # memory capacity on server :
```

Certainly! Here's a table that shows the equivalent commands in Python and Windows for various Linux shell commands:

Linux (Bash)	Python	Windows (cmd/powershell)
<code>ls (-a)</code>	<code>os.listdir()</code>	<code>dir /a</code> (cmd) / <code>Get-ChildItem -Force</code> (PowerShell)
<code>pwd</code>	<code>os.getcwd('folder_name')</code>	<code>cd</code> (cmd) / <code>\$pwd</code> (PowerShe

Linux (Bash)	Python	Windows (cmd/powershell)
<code>cat path_to_file</code>	-	<code>Get-Content path_to_file</code>
<code>wc path_to_file</code>	-	<code>`(Get-Content path_to_file</code>
<code>mkdir folder_name</code>	<code>os.mkdir('folder_name')</code>	<code>mkdir folder_name</code>
<code>cd path_to_folder</code>	<code>os.chdir('path_to_folder')</code>	<code>cd path_to_folder</code>
<code>touch filename</code>	<code>open('filename', 'a').close()</code>	<code>echo. > filename</code>
<code>scp (-R) origin destination</code>	<code>shutil.copytree('origin', 'destination')</code>	<code>xcopy /E origin destination (cmd) / Copy-Item -Recurse origin destination (PowerShell)</code>
<code>mv origin destination</code>	<code>shutil.move('origin', 'destination')</code>	<code>move origin destination</code>
<code>rm (-R) path</code>	<code>shutil.rmtree('path')</code>	<code>rmdir /s /q path (cmd) / Remove-Item -Recurse path (PowerShell)</code>
<code>tar -czvf compressed.tar.gz folder</code>	<code>shutil.make_archive('compressed', 'gztar', 'folder')</code>	<code>Compress-Archive -Path folder -DestinationPath compressed.tar.gz (PowerShell)</code>
<code>tar -xzvf compressed.tar.gz</code>	-	<code>Expand-Archive -Path compressed.tar.gz -DestinationPath folder (PowerShell)</code>
<code>echo "message"</code>	<code>print("message")</code>	<code>echo message</code>
<code>output > file.txt</code>	-	<code>output > file.txt (cmd) / Set-Content -Path file.tx -Value output (PowerShell)</code>
<code>output >> file.txt</code>	-	<code>output >> file.txt (cmd) , Add-Content -Path file.tx -Value output (PowerShell)</code>
<code>sudo command</code>	Run Python script as Administrator	Run command as Administrator
<code>ssh remote_machine_address</code>	<code>paramiko</code> or <code>subprocess</code> module	<code>ssh user@remote_machine_addre</code>

For Python equivalents, some functionalities might require specific libraries such as `os` , `shutil` , `subprocess` , or third-party libraries like `paramiko` for SSH functionality.

For Windows commands, keep in mind that some options might vary between Command Prompt (`cmd`) and PowerShell.

- What is the process of installing / deploying in Linux?
- How it differ from Windows? provide me with detailed explanation and guideline to install `Vaultwarden` on `ubuntu` .

Windows vs Linux

Here's a detailed explanation of the process of installing/deploying software in Linux compared to Windows, along with guidelines to install `Vaultwarden` on Ubuntu:

Process of Installing/Deploying Software in Linux vs Windows

In Linux, the process of installing and deploying software typically involves the following steps:

1. Package Management:
 - Most Linux distributions use package managers like `apt` (Debian-based), `yum` (Red Hat-based), or `pacman` (Arch Linux).
 - These package managers handle dependency resolution and installation of software packages.
2. Manual Installation:
 - Some software may require manual compilation from source code.
 - This often involves downloading the source, configuring, compiling, and installing.
3. Version Control:
 - Many Linux distributions use version control systems like git for managing software versions.
4. Configuration Files:
 - Linux often uses text-based configuration files instead of GUI settings.
5. Service Management:
 - Services are managed using init scripts, systemd units, or other service management tools.

In contrast, Windows typically uses:

1. Installer Packages:
 - `.exe` files that handle installation and sometimes uninstallation.
2. Microsoft Store:
 - For UWP apps and some traditional Windows applications.
3. Command Prompt or PowerShell:
 - Used for manual installations and configurations.
4. Registry Editor:
 - For system-wide settings and configurations.
5. Services.msc:
 - For managing Windows services.

Installing Vaultwarden on Ubuntu

Here's a step-by-step guide to install Vaultwarden on Ubuntu:

Step 1: Update System

```
sudo apt update && sudo apt upgrade -y
```

Step 2: Install Required Dependencies

```
sudo apt install -y curl wget gnupg2 ca-certificates
```

Step 3: Install Rust

Vaultwarden requires Rust to compile. Install it using rustup:

```
curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh  
source $HOME/.cargo/env
```

Follow the prompts to complete the installation.

Step 4: Clone Vaultwarden Repository

```
git clone https://github.com/dani-garcia/vaultwarden.git  
cd vaultwarden
```

Step 5: Build Vaultwarden

```
cargo build --release
```

Step 6: Create Systemd Service File

Create a new systemd service file:

```
sudo nano /etc/systemd/system/vaultwarden.service
```

Add the following content:

```
[Unit]
Description=Vaultwarden Server
After=network.target

[Service]
User=$USER
ExecStart=/path/to/vaultwarden/target/release/vaultwarden
Restart=always
WorkingDirectory=/path/to/vaultwarden

[Install]
WantedBy=multi-user.target
```

Replace `/path/to/vaultwarden` with the actual path to your vaultwarden directory.

Step 7: Enable and Start the Service

```
sudo systemctl enable vaultwarden
sudo systemctl start vaultwarden
```

Step 8: Configure Vaultwarden

Create a `.env` file in the vaultwarden directory:

```
nano /path/to/vaultwarden/.env
```

Add the following content, adjusting as needed:

```
DOMAIN=https://your-domain.com/vaultwarden/
ADMIN_TOKEN=<generated-hash-from-vaultwarden hash command>
SIGNUPS_ALLOWED=false
SMTP_HOST=smtp.yourdomain.com
SMTP_FROM=vaultwarden@yourdomain.com
SMTP_PORT=587
SMTP_SSL=true
SMTP_USERNAME=your-smtp-username
SMTP_PASSWORD=your-smtp-password
```


Generate the admin token:

```
/path/to/vaultwarden/target/release/vaultwarden hash
```

Copy the output and paste it into the ADMIN_TOKEN field in the .env file.

Step 9: Access Vaultwarden

Access Vaultwarden by navigating to <https://your-domain.com/vaultwarden/> in your web browser.

Key Points to Consider

1. Security: Ensure your server is properly secured, especially if exposing services to the internet.
2. Backup: Regular backups of your Vaultwarden data are crucial.
3. Updates: Keep Vaultwarden and its dependencies up-to-date.
4. Monitoring: Set up monitoring for the Vaultwarden service to ensure it's running smoothly.

Summary

This guide provides a detailed walkthrough for installing Vaultwarden on Ubuntu. It covers system preparation, dependency installation, building from source, creating a systemd service, configuring the environment, and accessing the application. The process demonstrates Linux-specific tools and techniques such as package management, manual compilation, and systemd service creation, highlighting the differences between Linux and Windows software deployment methods.

Citations:

- [1] https://www.reddit.com/r/vaultwarden/comments/14g8bd6/selfhosting_vaultwarden_how_is_it_done/
- [2] <https://www.bloovis.com/posts/2023-10-06-vaultwarden-without-docker/>
- [3] <https://vaultwarden.discourse.group/t/installation-for-a-noob/1609>
- [4] <https://github.com/rsmsctr/vaultwardenGuide>
- [5] https://ipv6.rs/tutorial/Ubuntu_Server_Latest/Vaultwarden/
- [6] <https://bitwarden.com/help/install-on-premise-linux/>
- [7] <https://www.informaticar.net/installing-on-premise-bitwarden-on-ubuntu-22-04/>
- [8] <https://docs.vultr.com/how-to-install-vaultwarden-on-ubuntu-20-04>
- [9] https://www.youtube.com/watch?v=v_7vJDwGWug
- [10] <https://linuxconfig.org/how-to-deploy-a-self-hosted-vaultwarden-instance>



To find the location of a file in Linux, you can use several commands. Here are some general methods:

1. `which`: This command shows the location of executables in your `PATH`.

```
which command_name
```

2. `whereis`: This command locates the binary, source, and manual page files for a command.

```
whereis command_name
```

3. `find`: This command searches for files in a directory hierarchy. You specify a starting directory and a file name pattern.

```
find / -name "file_name" 2>/dev/null
```

4. `locate`: This command is faster than `find` because it searches a pre-built database of files on the system.

```
locate file_name
```

5. `command -v`: This built-in command is similar to `which`, providing the path to executables.

```
command -v command_name
```

Example: Finding `systemctl`

You can use any of the commands mentioned above to find the location of `systemctl`. Here are examples:

1. Using `which`:

```
which systemctl
```

2. Using `whereis`:

```
whereis systemctl
```

3. Using `command -v`:

```
command -v systemctl
```

Sample Output

For instance, if you run `which systemctl` , you might get output like:

```
/usr/bin/systemctl
```

This indicates that the `systemctl` binary is located in `/usr/bin/` .

End of document

Operating System Comparison for Migration from CentOS 7

Comparison Matrix

Criteria	Ubuntu 22.04 LTS	Red Hat 8	Rocky Linux	Solaris 10.9	
Migration Ease from CentOS 7	Moderate <ul style="list-style-type: none">Different package manager (apt vs yum)Different file locationsSome reconfigurations needed	Very High <ul style="list-style-type: none">Same package managerSimilar file structureMinimal reconfiguration	Very High <ul style="list-style-type: none">Direct CentOS replacementSame package managerNearly identical structure	Low <ul style="list-style-type: none">Different architectureMajor reconfiguration neededDifferent toolset	
Perl Support	Good <ul style="list-style-type: none">Perl 5.34Good CPAN integrationModern module support	Excellent <ul style="list-style-type: none">Enterprise-grade Perl supportStable module ecosystemGood backward compatibility	Excellent <ul style="list-style-type: none">Same as RHEL/CentOSFull CPAN supportCompatible with existing scripts	Good <ul style="list-style-type: none">Limited module availabilityOlder Perl versionLess community support	
Python Support	Excellent <ul style="list-style-type: none">Python 3.10 by defaultLarge package repositoryModern features	Very Good <ul style="list-style-type: none">Python 3.9Enterprise stabilityConservative updates	Very Good <ul style="list-style-type: none">Same as RHELFull pip supportCompatible with CentOS tools	Limited <ul style="list-style-type: none">Older Python versionsLimited package supportManual installations needed	

Criteria	Ubuntu 22.04 LTS	Red Hat 8	Rocky Linux	Solaris 10.9	
Performance Under Load	Very Good <ul style="list-style-type: none"> • Modern kernel features • Good resource management • Efficient process handling 	Excellent <ul style="list-style-type: none"> • Enterprise-grade optimization • Proven performance • Good resource control 	Excellent <ul style="list-style-type: none"> • Same as RHEL • Enterprise-grade performance • Good scalability 	Excellent <ul style="list-style-type: none"> • Superior thread management • Good for high loads • Resource-intensive 	
Offline Environment Support	Good <ul style="list-style-type: none"> • Local repos possible • Needs planning for packages • Good offline docs 	Excellent <ul style="list-style-type: none"> • Built for enterprise • Satellite support • Complete offline solution 	Very Good <ul style="list-style-type: none"> • Local mirrors easy • Similar to CentOS/RHEL • Good offline tools 	Moderate <ul style="list-style-type: none"> • Complex repository setup • Limited package availability • Needs careful planning 	
Security Features	Strong <ul style="list-style-type: none"> • Regular security updates • AppArmor by default • Modern security tools 	Very Strong <ul style="list-style-type: none"> • SELinux by default • Enterprise security tools • Regular updates 	Very Strong <ul style="list-style-type: none"> • Same as RHEL • SELinux integration • Regular updates 	Very Strong <ul style="list-style-type: none"> • Zones support • Built-in security • Limited modern tools 	
Support & Updates	Free <ul style="list-style-type: none"> • Community support • 5 years basic support • 10 years paid ESM 	Paid <ul style="list-style-type: none"> • Enterprise support • Regular updates • Professional services 	Free <ul style="list-style-type: none"> • Community support • Regular updates • Growing ecosystem 	Paid <ul style="list-style-type: none"> • Oracle support • Limited updates • Expensive licensing 	
Cost	Free	High (Subscription required)	Free	Very High (License + Support)	

Recommendation

Based on your specific requirements:

1. **Best Choice: Rocky Linux**
 - Direct CentOS 7 replacement
 - Minimal migration effort
 - Free and community-supported
 - Excellent compatibility with existing applications
 - Strong performance for API workloads

2. **Second Choice: RHEL 8**
 - If budget allows and enterprise support is needed
 - Same benefits as Rocky Linux
 - Added professional support
3. **Third Choice: Ubuntu 22.04**
 - Good alternative if modern packages are needed
 - Free but requires more migration effort
 - Strong community support
4. **Not Recommended: Solaris 10.9**
 - Too different from current setup
 - High cost
 - Complex migration path
 - Limited modern tool support

Migration Considerations

1. **Pre-Migration**
 - Document all custom configurations
 - List all installed packages and dependencies
 - Backup all application and API code
 - Create test environment first
2. **During Migration**
 - Set up local repository mirrors
 - Test application compatibility
 - Verify Perl/Python version compatibility
 - Test load balancer configurations
3. **Post-Migration**
 - Monitor application performance
 - Verify security configurations
 - Update documentation
 - Train team on new system if needed