

Bonjour tout le monde

mot de passe sous Guacamole : password

Distribution
= File System
+

GUI & Windows

Serveur SSH

Files de conf.....

Kernel (Noyau) → Kernel.org

↳ Drivers

BASH

Gestion des processus

Fonctions de base "en mode noyau"

Distributions

RH ←

Centos 8

Fedora
↓
Ubuntu

} FS -
emplacement
des
conf

Centos 6
7
8

- Common
- Différence

Ubuntu 16
18
20
22

File System

/ → intégrité

dev / hda
hda
null
zero
Hy
std's ...

sd 0
1
2
...

Types de
fichiers
→ c (character)
→ b (binary)
→ d (directory)
→ p (pipe)

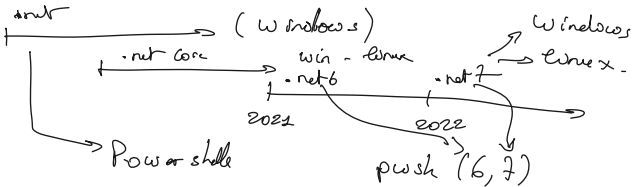
/proc

/etc → conf

/var → variables

DATA

pwsh $\rightarrow$ Powershell



↳ WSL 1 → Nécessaire d'être installé avec Windows
→ Ubuntu

→ WSL2 → VM sous hyperv.

→ Proj et Fonction - Windows

① → Acting " Sous Systeme Linux pour Windows "

② → choisir une distro via le Store
(ex: Ubuntu)

Recap Linux

- Recap Linux
- gestionnaire de paquets
 - apt (Debian, Ubuntu)
 - Yum (RHEL, CentOS)
 - dnf, rpm (la base)
 - zypper, chocolatey
 - organisation du filesystem
 - commune
 - Fonction de la distribution
 - Commandes
 - shell (cd, ls, mkdir, chown, ...)
 - externes - ps, ping, traceroute, vi, nano

Avoir de la doc :

calc --help (deviens une norme)

calc -h

sinon google

sinon pages github

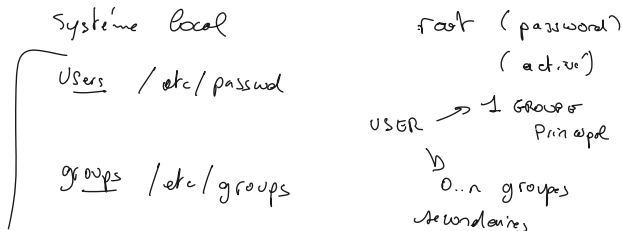
sinon, vieille méthode :

man calc

info calc

Norme concernant les paramètres :

--nomducommutateur = -ndc (version abrégée)



RBAC Role Based Access Control

↳ bonne pratique

Groupe $\Leftrightarrow$ Role (idem)

U sur → le propriétaire

G roup → " grp "

O thers → ceux qui ne sont pas du tout prop.

interpréteur à utiliser

Droits

Read

Write

execute

Fichiers

Dossiers

#!/bin/bash

cd
ls → modif du Contn
exéc SS Reps.

ch mod - champ de bits (absolu)

$$- \begin{array}{c|c|c} U & + & R \\ G & - & w \\ 0 & = & x \end{array}$$

ch mod 0+w fichier $\rightarrow$ ajout w autre

ch mod g-x fichier $\rightarrow$ retirer x en groupe prop
 $\left. \begin{array}{c} R \\ x \end{array} \right\}$ n'est pas modifié

ch mod g=0 lecture seule $\alpha' \begin{array}{c} G \\ 0 \end{array}$
retirer $w x$

Rwx
4 2 1

Rx
4+1=5

U G O
 $\begin{pmatrix} 4 & 2 & 1 \\ R & w & x \end{pmatrix}$ 4 2 1 4 2 1

U G O
Rx R
4+1 4 0

ch mod S40 fichier

Package locate

updatedb $\rightarrow$ index et maj l'index

locate < fichiers $\rightarrow$ recherche ultra
Rapide

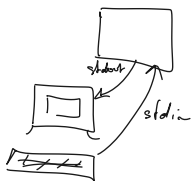
ls; cd .. → un commande
après l'autre

find / ... & → on arrive plus

>
<
>> <<

\$ → variable (d'environnement)

ls -l > liste
cat liste

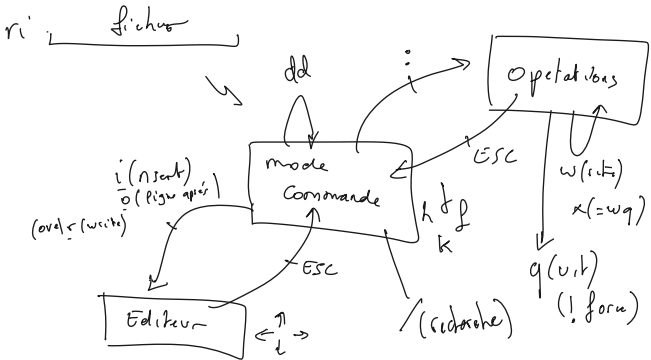


echo "ok"
→ stdout

read
grep ← stdin

cat liste | grep snap

Vi :



```
st01@main:~$ nano monscript.sh
st01@main:~$ source monscript.sh ( pas besoin de export )
ok
st01@main:~$ /bin/bash monscript.sh
ok
st01@main:~$ . ./monscript.sh
ok
st01@main:~$ ./monscript.sh ( doit être exécutable chmod +x, besoin de export
pour récupérer les variables modifiées )
ok
```

cat monscript.sh

```
#!/bin/bash
```

```
echo "ok"
```

chmod +x monscript.sh

Exercice 4 Atelier 6 : Sauvegardez votre alias dans un fichier de profile, rebootez (sudo reboot), vous devez pouvoir continuer de l'exécuter ensuite

Linux Admin

<http://lab.ib-formation.fr/guacamole/#/>

login 22201664

pass : lbformation!

Sinon : 22202383

pass : lbformation!

Machine dispo : 22201664

Machine :

- Gael ->
- Huy
-

Créer un script :

- créer un fichier
 - indiquer son interpreteur en première ligne :
 - `#!/bin/bash`
- le rendre exécutable : `chmod +x <nom du script>`
- l'exécuter :
 - en sous process : `./nomduscript`
 - dans le process actuel : `source <nomduscript>`

Bonnes pratiques :

- Pour définir des commandes à exécuter au démarrage pour tout le monde, ajoutez un fichier dans

`/etc/profile.d/<votre fichier>` (il sera exécuté automatique)

(il ne faut plus éditer `/etc/profile` ou `/etc/bashrc`)

- Pour définir les commandes a exécuter par utilisateur, vous les ajoutez dans le `~/.bash_profile`

exécution en sous-process

./script

export → env. de

→ aucune
manipulation de
mon contexte actuel

• ./script

export → utilité



mode de lancement

recommandé

Source ./script

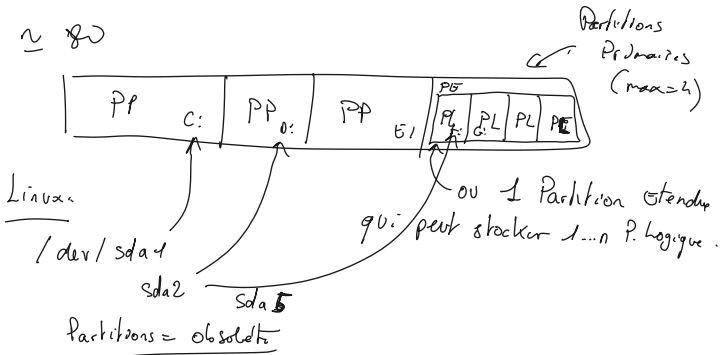
(pas forcément
besoin de
export)

Récupérer le stdout d'une commande en tant que variable (très pratique !)

REP=\$(pwd)

echo "Le répertoire actuel est : \$REP"

~ 80



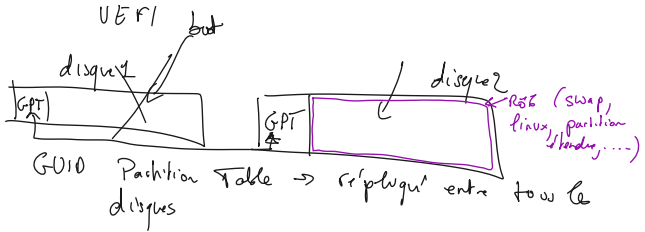
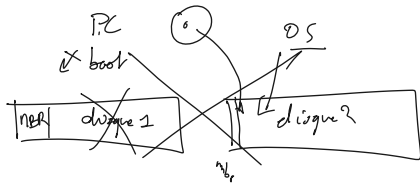
Volumes
disque

Volumes

— Simple
— Mirror
— bande
— etc....

(LV)

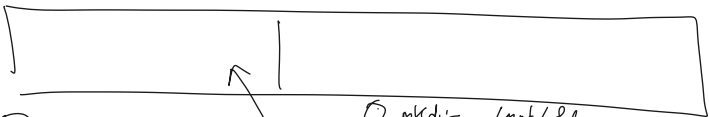




/dev/sd* → disques

/proc/partition → Table par disque

ls SCSI → disque par type



① fdisk → new

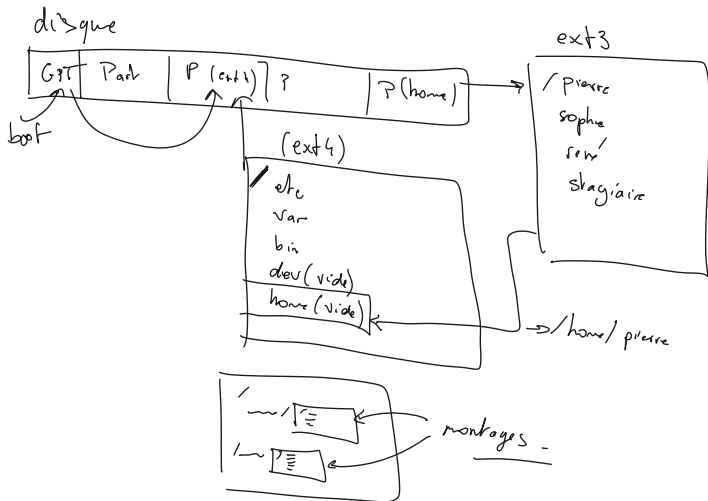
② mkfs /dev/sdb1
-t ext4

○ mkfs /mnt/files

○ mount <periph> <dossier>
/dev/sdb1 /mnt/files

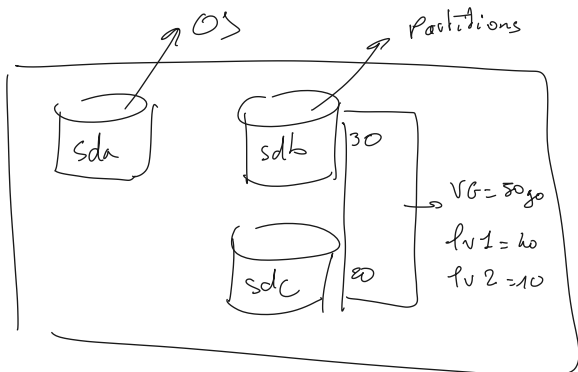
(ex)

○ /mnt/files → fs



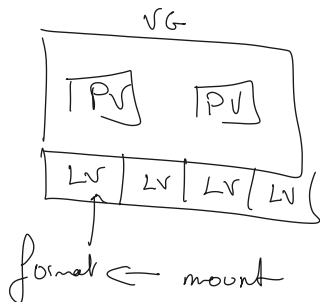
Gestion des volumes sous Centos :

https://www.server-world.info/en/note?os=CentOS_7&p=lv&f=1

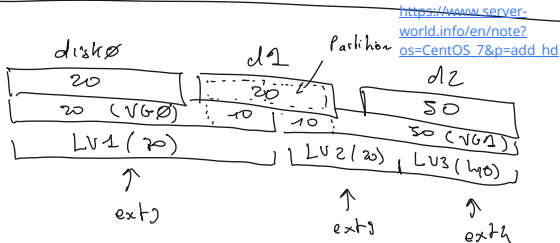


lvcreate -l 100%FREE -n lv10 vg0

PV		display
VG		create
LV		scan etc ...



PV → disque
 → partition de
 Type 8e



https://www.server-world.info/en/note?os=CentOS_7&p=lv&f=3

git clone <https://git.postgresql.org/git>

systemctl list-units

systemctl list-units --type=target --all

systemctl start stop restart enable disable reload status <nom de service>

Passer en mode maintenance :

systemctl set-default multi-user.target

systemctl set-default graphical.target

reboot

systemctl isolate multi-user.target -> hadoc

equivalent init 1 :

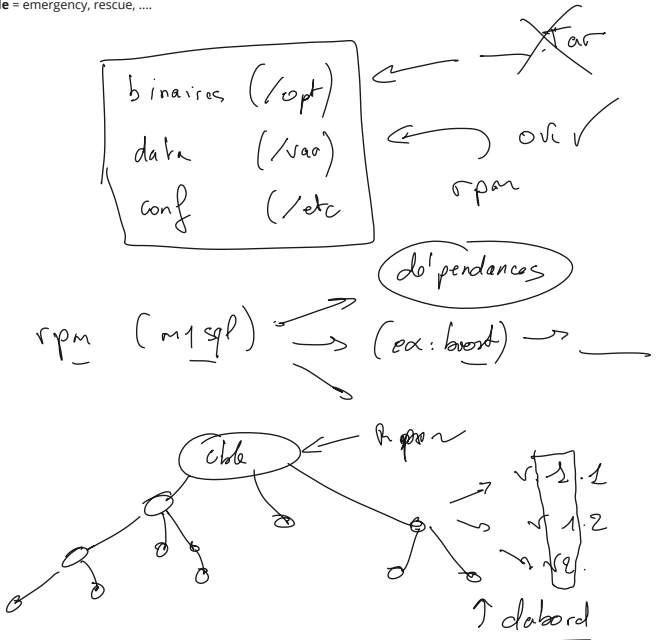
rescue.target

Choisir un target au boot :

Booter, e pour choisir les paramètres du noyau

ajouter à la ligne **linux**, le paramètre systemd.unit=<cible>.target

cible = emergency, rescue,



Redhat

(rpm)

↓
yum

↘ 8+

(update)

dnf (mêmes paramètres)

Deb

— .deb

(dpkg)

↙ (ubuntu)

aptitude

(apt)

→ update

→ upgrade

Gestionnaires de paquets

machine

OS

liste de sources
de paquets
(serveurs de contenu)

Repo Redhat

→ paquets officiels,
Secure, validés

pas les dernières
versions —

EPEL

Extra Packages for
Enterprise Linux

ajouter

Repo vers le produit - - - -

1) Ajouter le repo vers l'EPEL :

1. yum | dnf install epel-release : installer un repo supplémentaire
2. yum | dnf update : mettre à jour la liste de contenu et les packages installés.

Exercice : installer vscode en passant par snap :

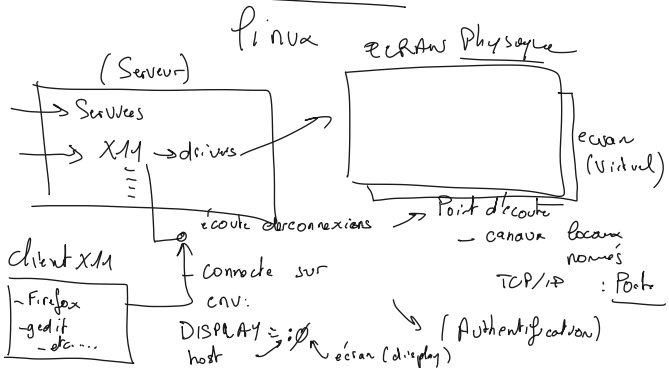
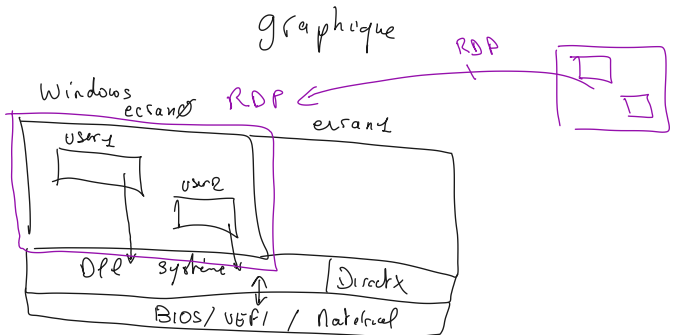
ajouter : `sudo ln -s /var/lib/snapd/snap /snap` après install de snap en plus de systemctl

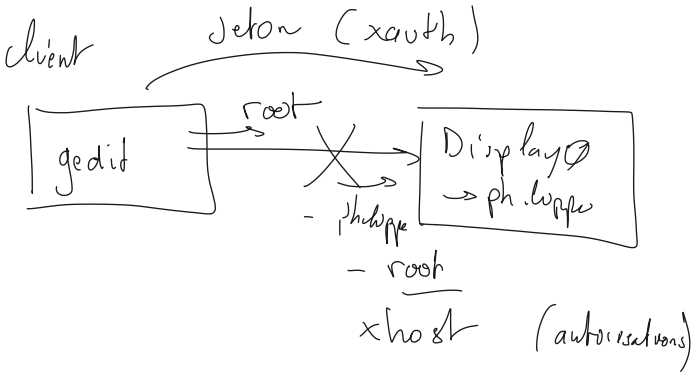
`systemctl enable snapd`

`systemctl start snapd`

`sudo snap install --classic code`

Exercice : installer la dernière version de git via gestionnaire de paquet (<https://git-scm.com/download/linux>)





xhost + si:localuser:root

Exercice final :

Installer un serveur postgres depuis les sources

1. Obtenir les sources :
 - a. `cd /usr/src`
 - b. `wget https://ftp.postgresql.org/pub/source/v14.4/postgresql-14.4.tar.bz2`
2. recommandé :
 - a. `mkdir -p /opt/<editeur>/<produit>/<version majeure>`
 - b. exemple `/opt/oracle/database/18`
3. extraire les sources
4. configurer la compilation de postgres et contrôler les prérequis :
 - a. `./configure ( option )`
5. installez les dépendances (versions **-devel**)
 - a. suggestions de conf :
 - i. `--prefix`
 - ii. `( --datarootdir )`
6. Lancer la compilation
 - a. `make ( -j <nombre de threads> - 8 pour 8 coeurs CPU par ex. )`
7. Installez/écrasez les binaires (selon ce qui a été défini dans prefix)
 - a. `make install`
8. créer un user postgres
9. Préparez un dossier et créez une bdd avec postgres:postgres propriétaire dans `/var/pg/db1`
10. Lancer le serveur pour tester
 - a. `./pg_ctl -D /var/pg/db1 start`
11. Se connecter au serveur pour tester
 - a. `psql -> \l` doit afficher les tables
12. Monter postgres en tant que service, et vérifier (après un reboot)
 - a. modèles dans `<src>/contrib/start-scripts`
 - b. `cp /usr/srv/postgres/contrib/start-scripts/linux /etc/init.d/pg1 &&`
`chmod +x /etc/init.d/pg1`
 - c. `init.d :`

Installation prefix

prefix=/opt/pg/14/pgsq

Data directory

PGDATA="/var/pg/db1"

Fabriquer le service script SystemD :

<https://unix.stackexchange.com/questions/220362/systemd-postgresql-start-script>