Linux

B1 - 2022-2023

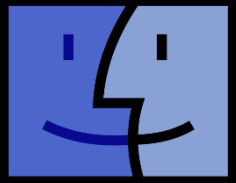
AIX ynov CAMPUS

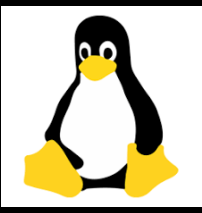
UP CONNECT
NETWORK - SYSTEM - CONSULTING

Sommaire

1. Qu’est-ce qu’un OS ?

Un OS





Un système d'exploitation est un **ensemble de programmes permettant la gestion des ressources disponibles d'un ordinateur.**

Il gère :

- La mémoire physique : barrette de ram + cache processeur → sert à l'exécution des programmes.
- La mémoire virtuelle : emplacement sur disque dur (Swap) → sert à décharger la mémoire physique.

Il gère les accès aux périphériques (souris, clavier, webcam, carte réseau…).

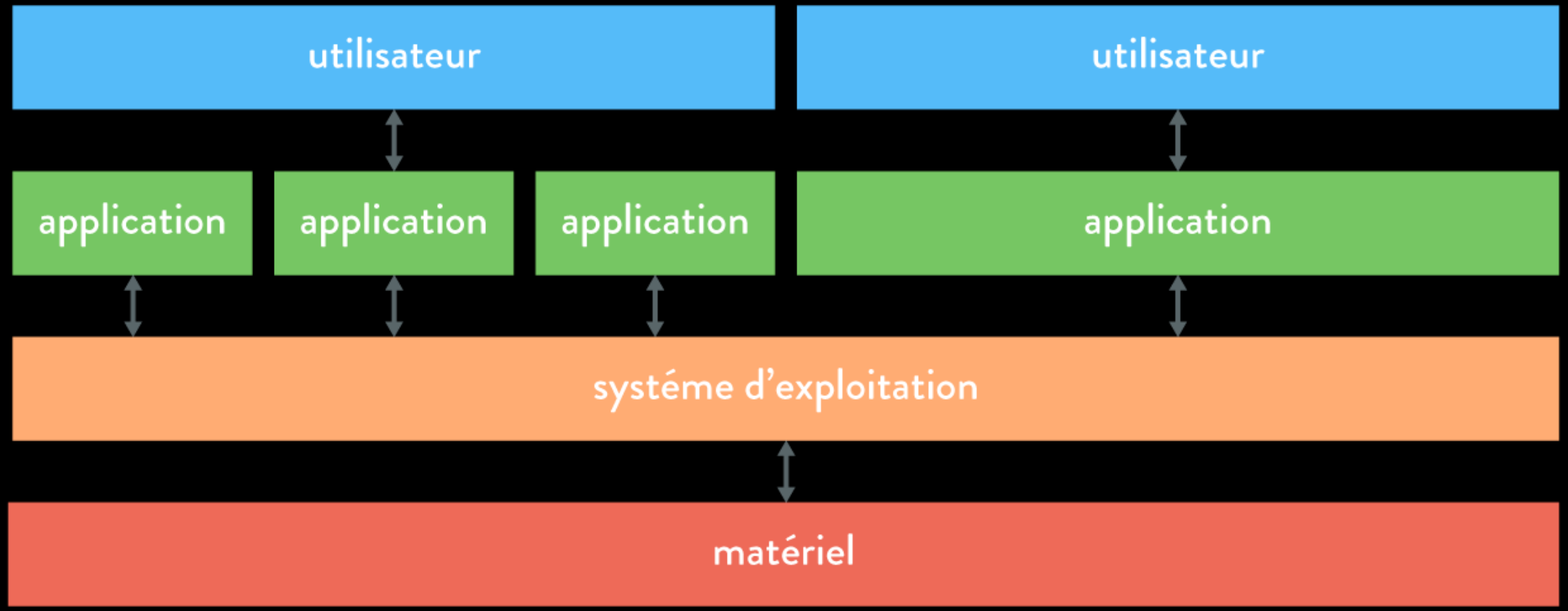
Il ordonnance les processus.

Il protège les fichiers contre tout accès non autorisé.

Il collecte les informations sur les programmes utilisés ou en cours d'utilisation.



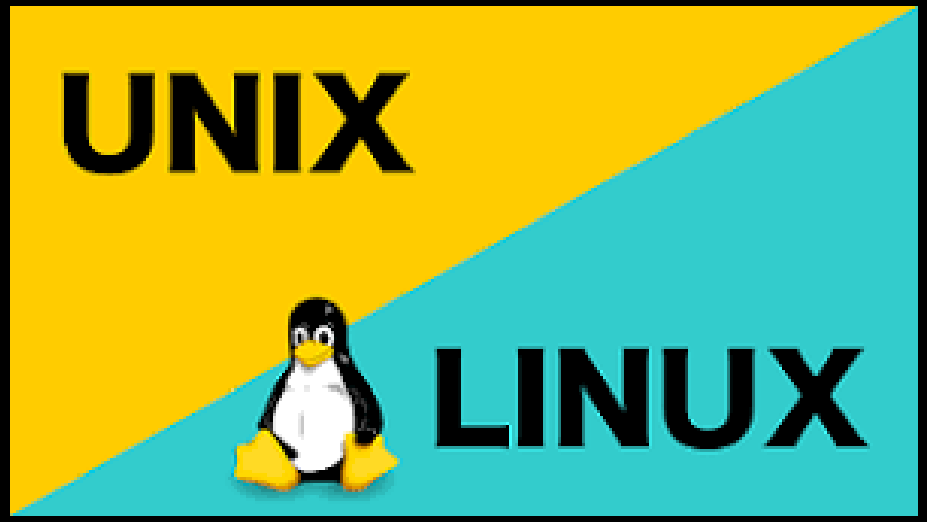
Un OS


utilisateur
utilisateur
application
application
application
application
systéme d'exploitation
matériel

2. Présentation de Linux

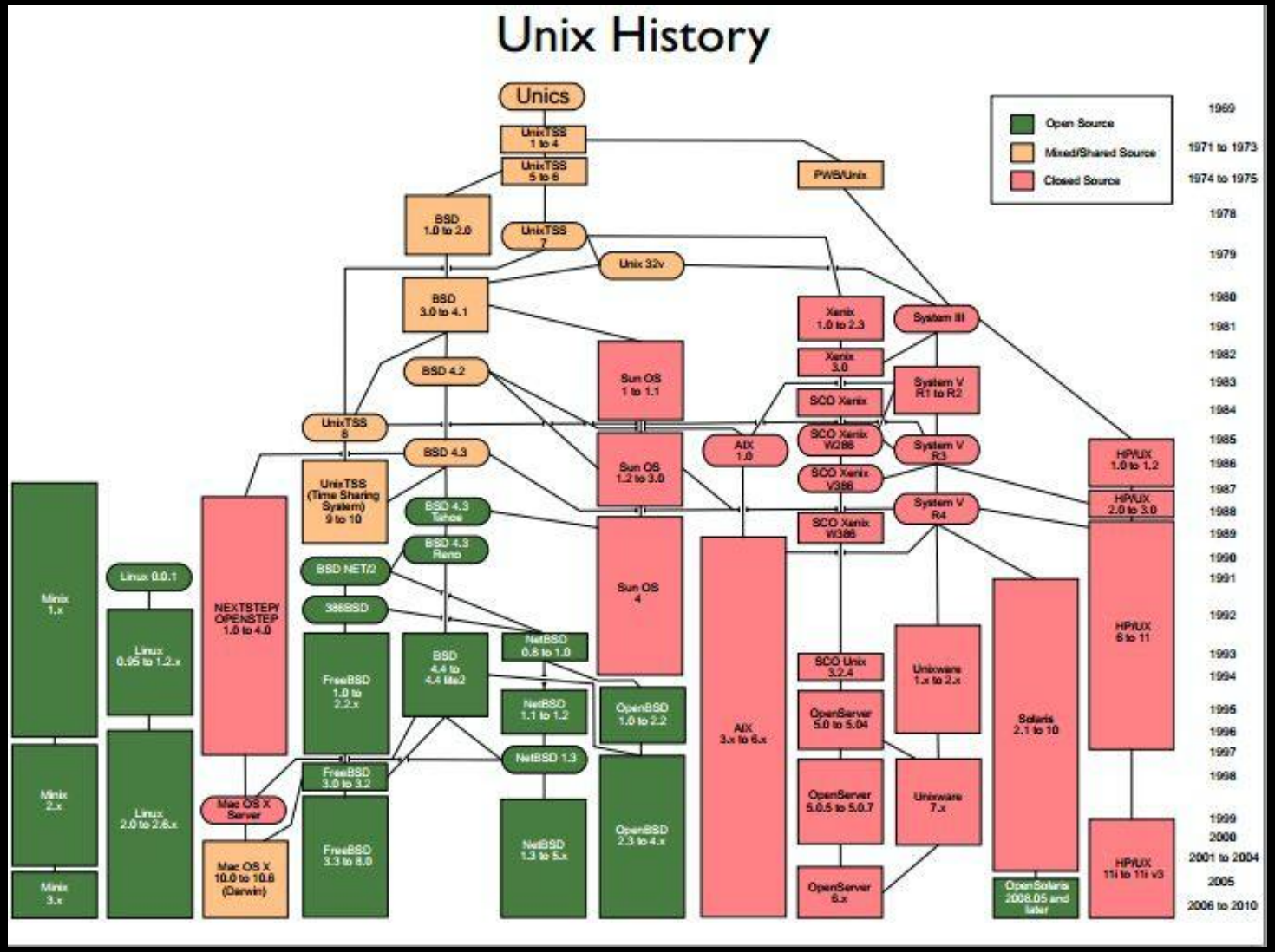
Unix et Linux

- Unix : Système d'exploitation **multi-tâches**
- « **Multics** » créé en **1965** pour le jeu « Space Travel »
- Besoin de réduire les performances de Multics sur petite machine : création de **UNICS** en **1969**
- Rapidement contractée en **Unix**
- **1 Janvier 1970** : considéré comme la date de naissance du système Unix



- 1985 : Professeur Hollandais a développé le système « **Minix** » pour ses étudiants
- 1991 : Linus Torvalds, étudiant finlandais conçoit l'OS : « **Linux** » (Arch 386)
- Respect des normes POSIX
- GNU : OS créé avant Linux mais rapidement rattrapé et dépassé par Linux : on parle maintenant de GNU/Linux

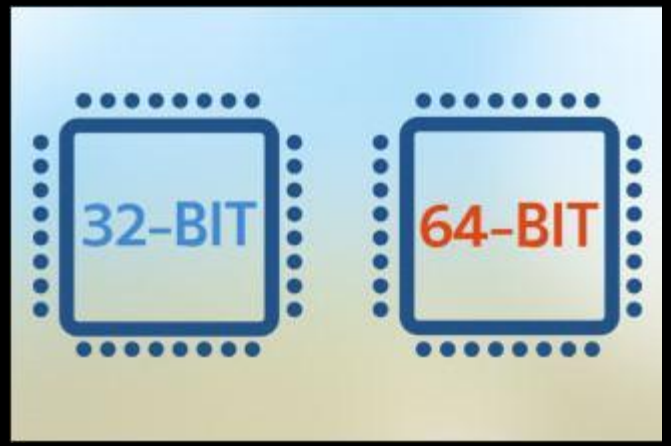
L’évolution d’Unix



Source : https://evomics.org

Architecture 32 bits ou 64 bits

- 1 **bit** : **0** ou **1**
- **Mots** : succession de bits : exemple « Bonjour »
- Processeur : cerveau de l'ordinateur
- Modèle 32 bits : processeur gère 2^32 bits d'informations
- Limite : la quantité d'information → la gestion de la RAM
- 2^ 32 ~= 4 milliards d'adresses, soit environ 4 Go
- Modèle 64 bits : pas de limitation de RAM : 2^64 ~= 18 milliards de Giga-Octets



L’utilisation de Linux

- La stabilité de l’OS
  - Robustesse
  - Réactif
  - Rapide…
- La sécurité
  - Séparation des processus
  - Système de permissions
  - Couche réseau
- Le coût
  - Peu gourmand en ressource
  - Pas de licence

Des exemples d’utilisateurs

- L’infrastructure d’Internet
  - Amazon
  - Google
  - Facebook
  - Azure ! (pas que linux)
- Les 500 superordinateurs
  - la NASA par exemple
- Notre quotidien
  - Routeur opérateur
  - Télévision
  - Smartphone
  - Objets connectés
- La Gendarmerie Française

Un logiciel libre

D’après FSF (Free Software Foundation) :

- Liberté d’utiliser le logiciel
- Liberté de le copier
- Liberté d’en étudier le fonctionnement
- Liberté de le modifier et de redistribuer cette version modifiée

FREE SOFTWARE
FOUNDATION

Un peu plus de détail - L’architecture

- Le noyau (ou kernel) est le premier composant logiciel.
  - Il est le cœur du système UNIX.
  - C’est lui qui gère les ressources matérielles du système.
  - Les autres composants logiciels passent obligatoirement par lui pour accéder au matériel.

- Le Shell est un utilitaire qui interprète les commandes de l’utilisateur et assure leur exécution.
  - Principaux shell : Bourne shell, C shell, Korn shell et Bourne Again shell (bash).

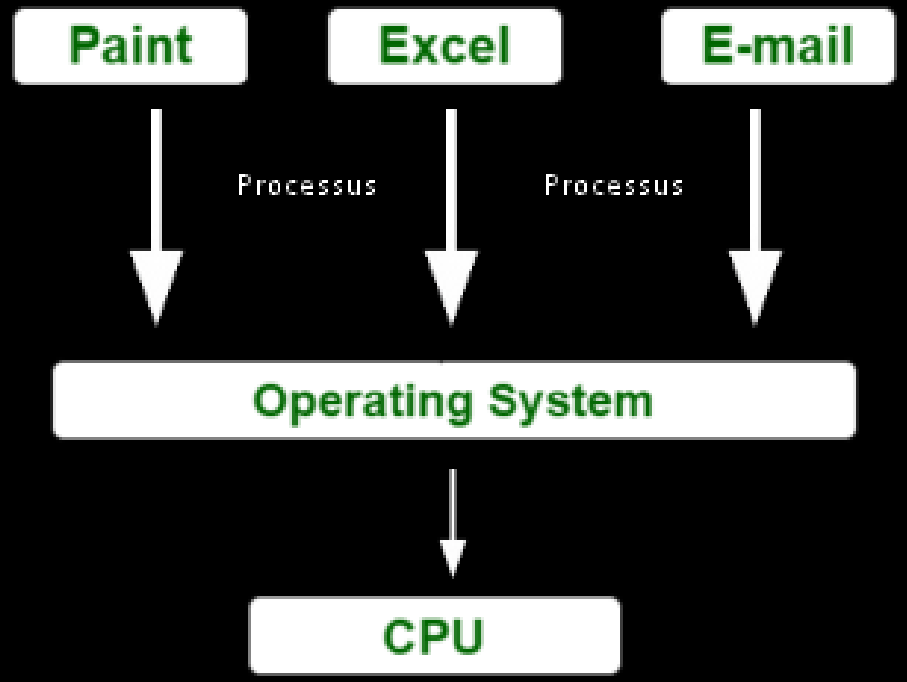
- Les applications regroupent les programmes utilisateurs comme :
  - le navigateur internet.
  - le traitement de texte.
  - ...

Un peu plus de détail - Le multitâche

Linux fait partie de la famille des systèmes d'exploitation à temps partagé. Il partage le temps d'utilisation processus entre plusieurs programmes, passant de l'un à l'autre de façon transparente pour l'utilisateur.

Cela implique :

- Exécution simultanée de plusieurs programmes.
- Distribution du temps CPU par l'ordonnanceur.
- Réduction des problèmes dus à une application défaillante.
- Diminution des performances lorsqu'il y a trop de programmes lancés.

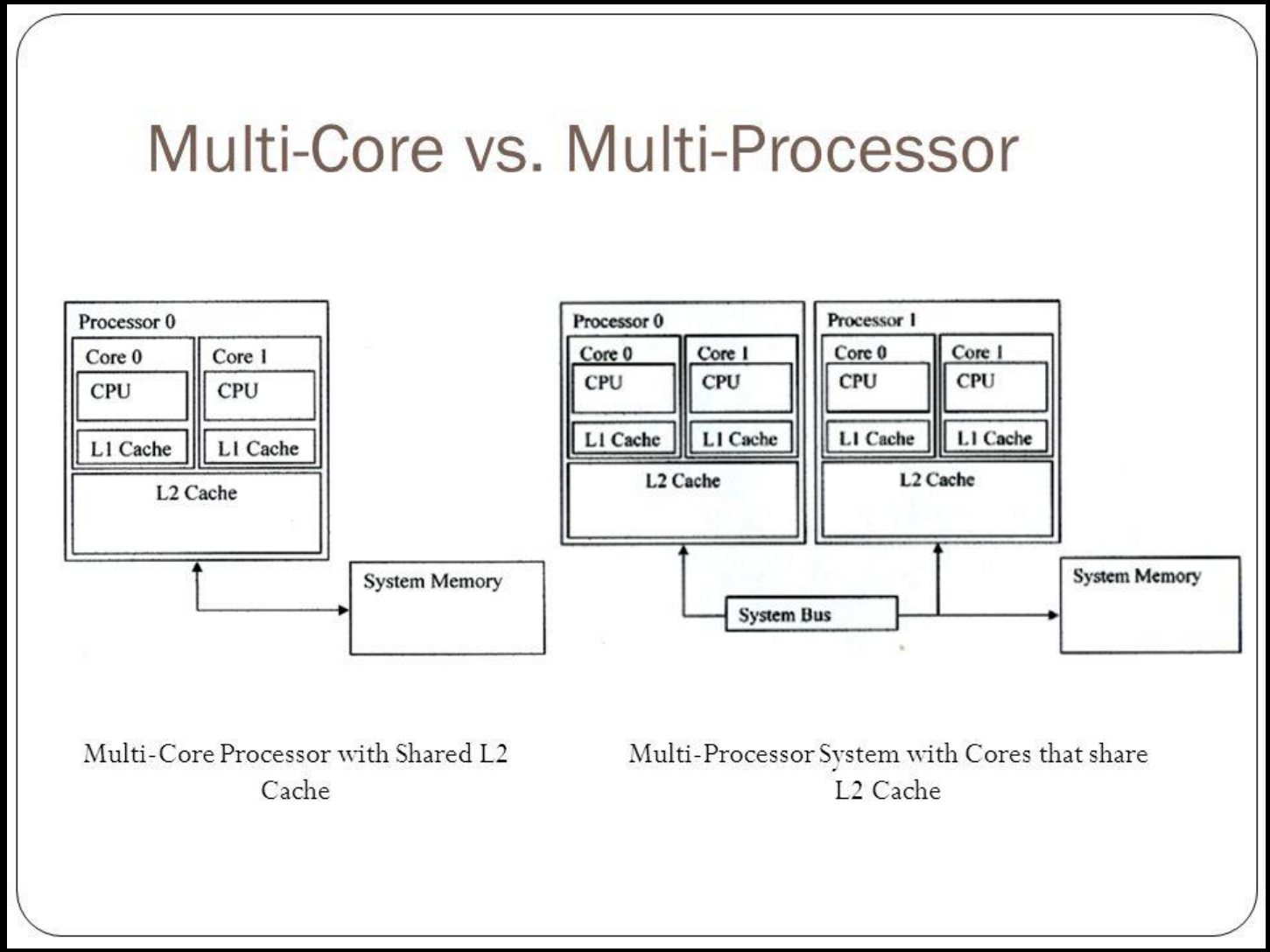


Un peu plus de détail - Le multiutilisateur

La finalité de Multics était de permettre à plusieurs utilisateurs de travailler à partir de plusieurs terminaux (écran et clavier) sur un seul ordinateur (très coûteux à l'époque).

Linux étant un descendant de ce système d'exploitation, il a gardé cette capacité à pouvoir fonctionner avec plusieurs utilisateurs simultanément et en toute indépendance, chacun ayant son compte utilisateur, son espace de mémoire et ses droits d'accès aux fichiers et aux logiciels.

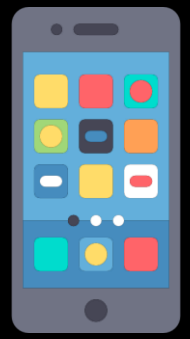
Un peu plus de détail - Le multiprocesseur



Un peu plus de détail - Le multiplateforme

Linux est écrit en langage de haut niveau pouvant s'adapter à différents types de plateformes lors de la compilation. Il fonctionne donc sur :

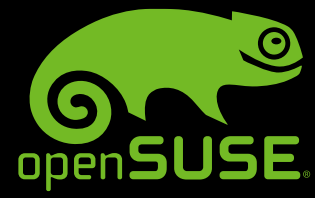
- Les ordinateurs des particuliers (le PC ou l'ordinateur portable).
- Les serveurs (données, applications,...).
- Les ordinateurs portables (les smartphones ou les tablettes).
- Les systèmes embarqués (ordinateur de voiture).
- Les éléments actifs des réseaux (routeurs, commutateurs).
- Les appareils ménagers (téléviseurs, réfrigérateurs,...).



Le choix de la distribution

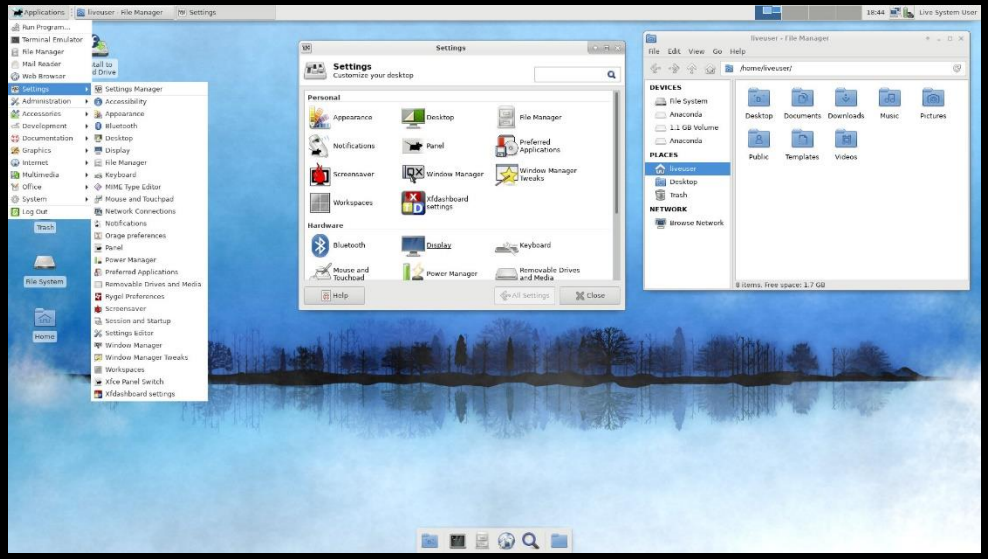
- Distributions communautaires
  - Debian
  - CentOS
  - Fedora
  - …
- Distributions commerciales
  - RedHat
  - Oracle
  - …

- La connaissance de l'OS
- Les prérequis applicatifs
- Le choix du support
- Les mises à jours

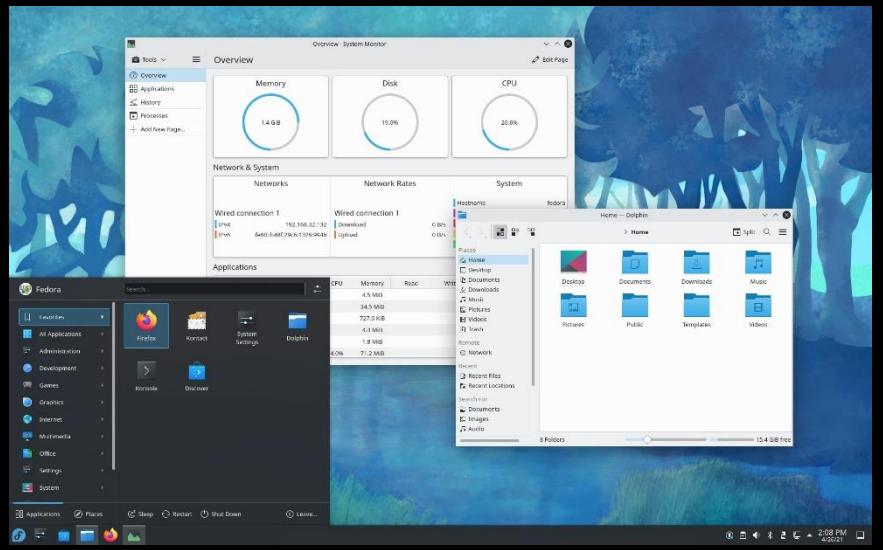


Les environnements de bureaux

Gnome 3



XFCE



KDE

Le shell

Le shell, interface de commandes en français, permet aux utilisateurs d'envoyer des ordres au système d'exploitation.

Il est moins visible aujourd'hui, depuis la mise en place des interfaces graphiques, mais reste un moyen privilégié sur les systèmes Linux qui ne possèdent pas tous des interfaces graphiques et dont les services ne possèdent pas toujours une interface de réglage.

Il offre un véritable langage de programmation comprenant les structures classiques : boucles, alternatives et les constituants courants : variables, passage de paramètres, sous-programmes. Il permet donc la création de scripts pour automatiser certaines actions (sauvegardes, création d'utilisateurs, surveillance du système,...).

Il existe plusieurs types de Shell disponibles et configurables sur une plate-forme ou selon le choix préférentiel de l'utilisateur :

- **sh**, le shell aux normes POSIX.
- **csh**, shell orienté commandes en C.
- **bash**, Bourne Again Shell, shell de Linux.
- ...

Les fonctionnalités du shell

- Permet d'exécuter des commandes (vérification + exécution de la commande).
- Redirection Entrées/Sorties (renvoi des données dans un fichier par exemple).
- Gère les connexions des utilisateurs (processus de connexion).
- Permet la création de scripts (langage de programmation interprété).
- Donne accès aux informations propres au système en cours de fonctionnement (variable d'environnement).

```
chris@ubuntu:~$ bash
chris@ubuntu:~$ zsh
chris@ubuntu ~ % tcsh
ubuntu:~> dash
$
```

```
#!/bin/bash
#INPUT_SAMPLE_LIST=$1
cd /Volumes/PhilDrive_EMS/TestDec7/snv_postprocess/
...
. paths.txt
...

echo "Debug level set for $DEBUG_LEVEL"
echo "log found in scripts directory"
...
cp $HIGH_SNP_OUT ./
cp $LOW_SNP_OUT ./
cp $GERM_SNP_OUT ./
#  echo "${SCRIPT_DIR}/run_somatic_mutation_analysis ${i} no_false_snp"
if [ $DEBUG_LEVEL -gt 0 ]
then
echo "INFO: ${SCRIPT_DIR}run_somatic_mutation_analysis.sh $SAMPLE no_false_snp
`basename ${LOW_SNP_OUT}` `basename ${GERM_SNP_OUT}` `basename ${HIGH_SNP_OUT}`
${D_BAM_FILE} ${G_BAM_FILE}\n">${LOG}

fi
${SCRIPT_DIR}run_somatic_mutation_analysis.sh

echo "End of somatic mutation analysis">> $LOG
```

3. Installation du Système

L’OS Debian GNU/Linux

- Exclusivement composée de logiciels libres
- Développé par Debian Project fondée en 1993
- Première version stable en 1996
- Environ 60 000 paquets logiciels

A vos VM !

Pratique

Installation de la machine sous Debian :

- Disposition du clavier : Français
- Partitionner le disque dur automatiquement (en LVM de préférence)
- Configurer le réseau (NAT) en DHCP + nom d'hôte (nom-prenom)
- Créer un utilisateur non root (votre prénom)
- Paquets : installation interface graphique (KDE par exemple)
- Se connecter sur la machine

Questions :

1. Quelles difficultés avez-vous rencontrées durant ce TP ?
2. Comment avez-vous fait face à ces difficultés ?

4. Les commandes de bases

PWD

> La commande **pwd** permet de trouver le chemin du répertoire de travail (dossier) dans lequel vous êtes actuellement.

```
root@sli-haproxy-1 /etc/letsencrypt:# pwd
/etc/letsencrypt
```

CD

La commande **cd** permet de naviguer dans les fichiers et répertoires du système.

```
root@sli-haproxy-1 /etc/letsencrypt:# cd /opt/scripts/
root@sli-haproxy-1 /opt/scripts:#
```

- « **cd ..** » (avec deux points) pour se déplacer d'un répertoire vers le haut
- « **cd** » pour aller directement au dossier principal (home)
- « **cd –** » (avec un tiret) pour passer à votre répertoire précédent

```
root@sli-haproxy-1 /opt/scripts:# cd /etc/letsencrypt
root@sli-haproxy-1 /etc/letsencrypt:# cd -
/opt/scripts
root@sli-haproxy-1 /opt/scripts:#
```

LS

La commande **ls** permet de visualiser le contenu d'un répertoire.

```
root@sli-haproxy-1 /opt/scripts:# ls
apiip.php
root@sli-haproxy-1 /opt/scripts:#
```

- « **ls –R** » liste tous les fichiers dans les sous-répertoires
- « **ls –a** » liste les fichiers cachés
- « **ls -al**» liste les fichiers et répertoires avec informations détaillés (autorisation, tailles…) en format de liste

```
root@sli-haproxy-1 /opt:# ls -R
.:
scripts

./scripts:
apiip.php
root@sli-haproxy-1 /opt:#
```

CAT

La commande **cat** permet de visualiser le contenu d'un fichier.

```
root@sli-haproxy-1 /opt/scripts:# cat apiip.php
<?php

//-- Fonction : Call de l'api

function CallApi($FK_CLIENT,$S_API_REQUEST,$S_METHODE = 'GET',$S_DATA = array(),$S_TOKEN = '') {

        //--
```

- « **cat > nouveauFichier** » crée un nouveau fichier
- « **cat fichier1 fichier2 > fichier3** » concatène les deux fichiers en un troisième

```
root@sli-haproxy-1 /opt/scripts:# cat apiip.php | tr a-z A-Z > resultat.txt
root@sli-haproxy-1 /opt/scripts:# cat resultat.txt
<?PHP

//-- FONCTION : CALL DE L'API

FUNCTION CALLAPI($FK_CLIENT,$S_API_REQUEST,$S_METHODE = 'GET',$S_DATA = ARRAY(),$S_TOKEN = '') {

        //--
```

CP

La commande **cp** permet de copier les fichiers du répertoire actuel dans un autre répertoire.

```
root@sli-haproxy-1 /opt/scripts:# cp apiip.php dossier/
root@sli-haproxy-1 /opt/scripts:# ls dossier/
apiip.php
```

- « **cp –r** » copie récursivement les répertoires

MKDIR

La commande **mkdir** permet de créer un nouveau répertoire.

```
root@sli-haproxy-1 /opt/scripts/dossier:# mkdir newdossier
root@sli-haproxy-1 /opt/scripts/dossier:# ls
newdossier  test.php
```

- « **mkdir -p** » créer les répertoires de manière récursif

```
root@sli-haproxy-1 /opt/scripts/dossier:# ls -al
total 16
drwxr-xr-x 3 root root 4096 mars  28 14:02 .
drwxr-xr-x 3 root root 4096 mars  28 13:56 ..
drwxr-xr-x 2 root root 4096 mars  28 14:02 newdossier
-rwxr-xr-x 1 root root 3371 mars  28 13:56 test.php
root@sli-haproxy-1 /opt/scripts/dossier:# mkdir -p newdossier2/dossierRecursif
root@sli-haproxy-1 /opt/scripts/dossier:# ls -al
total 20
drwxr-xr-x 4 root root 4096 mars  28 14:06 .
drwxr-xr-x 3 root root 4096 mars  28 13:56 ..
drwxr-xr-x 2 root root 4096 mars  28 14:02 newdossier
drwxr-xr-x 3 root root 4096 mars  28 14:06 newdossier2
-rwxr-xr-x 1 root root 3371 mars  28 13:56 test.php
root@sli-haproxy-1 /opt/scripts/dossier:# ls -al newdossier2/
total 12
drwxr-xr-x 3 root root 4096 mars  28 14:06 .
drwxr-xr-x 4 root root 4096 mars  28 14:06 ..
drwxr-xr-x 2 root root 4096 mars  28 14:06 dossierRecursif
```

RMDIR

La commande **rmdir** permet de supprimer un répertoire qui est vide.

```
root@sli-haproxy-1 /opt/scripts/dossier:# rmdir newdossier2/dossierRecursif/
root@sli-haproxy-1 /opt/scripts/dossier:# ls -al newdossier2
total 8
drwxr-xr-x 2 root root 4096 mars  28 14:36 .
drwxr-xr-x 4 root root 4096 mars  28 14:06 ..
```

RM

La commande **rm** permet de supprimer les répertoires et leur contenu.

```
root@sli-haproxy-1 /opt/scripts/dossier:# rm -r newdossier2/
root@sli-haproxy-1 /opt/scripts/dossier:# ls -al
total 16
drwxr-xr-x 3 root root 4096 mars  28 14:37 .
drwxr-xr-x 3 root root 4096 mars  28 13:56 ..
drwxr-xr-x 2 root root 4096 mars  28 14:02 newdossier
-rwxr-xr-x 1 root root 3371 mars  28 13:56 test.php
```

NB : Soyez très prudent avec la commande « **rm** » car vous pouvez facilement supprimer des fichiers systèmes et corrompre votre machine.

TOUCH

> La commande **touch** permet de créer un fichier vierge.

```
root@sli-haproxy-1 /opt/scripts/dossier:# touch fichiervide.txt
root@sli-haproxy-1 /opt/scripts/dossier:# ls -al
total 16
drwxr-xr-x 3 root root 4096 mars  28 14:39 .
drwxr-xr-x 3 root root 4096 mars  28 13:56 ..
-rw-r--r-- 1 root root    0 mars  28 14:39 fichiervide.txt
drwxr-xr-x 2 root root 4096 mars  28 14:02 newdossier
-rwxr-xr-x 1 root root 3371 mars  28 13:56 test.php
```

LOCATE

La commande **locate** permet de localiser un fichier. Avec l'argument –i , la recherche sera insensible à la casse.

```
root@sli-haproxy-1 /opt:# locate -i *vide.txt
/opt/scripts/dossier/fichiervide.txt
```

NB : Locate effectue cette recherche dans une base de données rafraîchie automatiquement toutes les 24h. Pour actualiser manuellement l'indexation, il suffit de lancer la commande « **updatedb** »

LOCATE

La commande **locate** permet de localiser un fichier. Avec l'argument –i , la recherche sera insensible à la casse.

```
root@sli-haproxy-1 /opt:# locate -i *vide.txt
/opt/scripts/dossier/fichiervide.txt
```

NB : Locate effectue cette recherche dans une base de données rafraîchie automatiquement toutes les 24h. Pour actualiser manuellement l'indexation, il suffit de lancer la commande « **updatedb** »

FIND

La commande **find** permet de localiser un fichier ou un répertoire.

```
root@sli-haproxy-1 /opt:# find /opt -name *vide.txt
/opt/scripts/dossier/fichiervide.txt
```

Find cherche dans le répertoire local par défaut, mais on peut spécifier un répertoire de recherche, à la différence de « **locate** ».

GREP

La commande **grep** permet de rechercher tout le texte d'un fichier donné.

```
root@sli-haproxy-1 /opt/scripts:# grep api apiip.php
//-- Fonction : Call de l'api
        $ch = curl_init('https://api.com'.$S_API_REQUEST.'');
//-- Votre clé api récupérable sur le manager -> parametres -> gestion des comptes
```

La commande renverra toutes les lignes ou le mot recherché figure.

SUDO

> La commande **sudo** permet d'effectuer des tâches qui nécessitent des autorisations administratives (root).

```
user@sli-haproxy-1:~$ apt update
Lecture des listes de paquets... Fait
E: Impossible d'ouvrir le fichier verrou /var/lib/apt/lists/lock - open (13: Permission non accordée)
E: Impossible de verrouiller le répertoire /var/lib/apt/lists/
W: Problème de suppression du lien /var/cache/apt/pkgcache.bin - RemoveCaches (13: Permission non accordée)
W: Problème de suppression du lien /var/cache/apt/srcpkgcache.bin - RemoveCaches (13: Permission non accordée)
user@sli-haproxy-1:~$ sudo apt update
sudo: impossible de déterminer le nom de l'hôte sli-haproxy-1: Nom ou service inconnu

Nous espérons que vous avez reçu de votre administrateur système local les consignes traditionnelles. Généralement, elles se concentrent sur ces trois éléments :

    #1) Respectez la vie privée des autres.
    #2) Réfléchissez avant d'utiliser le clavier.
    #3) De grands pouvoirs confèrent de grandes responsabilités.

[sudo] Mot de passe de user :
user n'apparaît pas dans le fichier sudoers. Cet événement sera signalé.
```

> NB : La configuration du « sudoers » est nécessaire pour donner les droits nécessaires aux utilisateurs.

DF

> La commande **df** permet d'obtenir un rapport sur l'utilisation de l'espace disque du système.

```
root@sli-haproxy-1 /opt/scripts:# df -h
Sys. de fichiers                      Taille Utilisé Dispo Uti% Monté sur
udev                                    982M       0  982M   0% /dev
tmpfs                                   200M     21M  179M  11% /run
/dev/mapper/TEMPLATE--DEBIAN--vg-root    23G    1,8G   20G   9% /
tmpfs                                   998M       0  998M   0% /dev/shm
tmpfs                                   5,0M       0  5,0M   0% /run/lock
tmpfs                                   998M       0  998M   0% /sys/fs/cgroup
/dev/sda1                               236M     84M  140M  38% /boot
tmpfs                                   200M       0  200M   0% /run/user/0
```

> NB : Le « -h » permet d'afficher des valeurs (taille) en Octet

DU

La commande **du** permet de vérifier l'espace occupé par un fichier ou un répertoire

```
root@sli-haproxy-1 /opt/scripts:# du * -h
4,0K    apiip.php
4,0K    dossier/newdossier
12K     dossier
4,0K    resultat.txt
```

NB : Le « -h » permet d'afficher des valeurs (taille) en Octet et l'étoile permet de tout sélectionner dans le dossier actuel.

HEAD

La commande **head** permet visualiser les premières lignes de n'importe quel fichier texte. Par défaut, elle affichera les 10 premières lignes.

```
root@sli-haproxy-1 /opt/scripts:# head resultat.txt
<?PHP

//-- FONCTION : CALL DE L'API
```

NB : Le « -n » permet d'afficher le nombre de ligne que vous souhaitez.

TAIL

La commande **tail** permet visualiser les dernières lignes de n'importe quel fichier texte. Par défaut, elle affichera les 10 dernières lignes.

```
root@sli-haproxy-1 /opt/scripts:# tail resultat.txt -n 3
}

PRINT R($RESULTAPI);
```

NB : Le « -n » permet d'afficher le nombre de ligne que vous souhaitez.

TAR

La commande **tar** permet d'archiver des fichier dans un « tarball ».

```
root@sli-haproxy-1 /opt/scripts:# tar -cvf resultat.tar /opt/scripts/resultat.txt
tar: Suppression de « / » au début des noms des membres
/opt/scripts/resultat.txt
root@sli-haproxy-1 /opt/scripts:# ll
total 32
drwxr-xr-x 3 root root  4096 mars  28 17:02 .
drwxr-xr-x 3 root root  4096 mars  21 10:32 ..
-rwxr-xr-x 1 root root  3371 mars  21 11:51 apiip.php
drwxr-xr-x 3 root root  4096 mars  28 14:39 dossier
-rw-r--r-- 1 root root 10240 mars  28 17:02 resultat.tar
-rw-r--r-- 1 root root  3371 mars  28 13:52 resultat.txt
```

NB : Les arguments permettent de compléter la commande principale.

Quelques exemples ici : https://www.linuxtechi.com/17-tar-command-examples-in-linux/

CHMOD

La commande **chmod** permet de modifier les permissions de lecture, d'écriture et d'exécution des fichiers et des répertoires.

```
root@sli-haproxy-1 /opt/scripts:# ls -al
total 32
drwxr-xr-x 3 root root  4096 mars  28 17:02 .
drwxr-xr-x 3 root root  4096 mars  21 10:32 ..
-rwxr-xr-x 1 root root  3371 mars  21 11:51 apiip.php
drwxr-xr-x 3 root root  4096 mars  28 14:39 dossier
-rw-r--r-- 1 root root 10240 mars  28 17:02 resultat.tar
-rw-r--r-- 1 root root  3371 mars  28 13:52 resultat.txt
root@sli-haproxy-1 /opt/scripts:# chmod 770 resultat.txt
root@sli-haproxy-1 /opt/scripts:# ls -al
total 32
drwxr-xr-x 3 root root  4096 mars  28 17:02 .
drwxr-xr-x 3 root root  4096 mars  21 10:32 ..
-rwxr-xr-x 1 root root  3371 mars  21 11:51 apiip.php
drwxr-xr-x 3 root root  4096 mars  28 14:39 dossier
-rw-r--r-- 1 root root 10240 mars  28 17:02 resultat.tar
-rwxrwx--- 1 root root  3371 mars  28 13:52 resultat.txt
```

NB : Le nombre 770 représente les droits que l'on attribue. Nous verrons les droits plus en détail plus tard.

CHOWN

La commande **chown** permet de modifier la propriété d'un fichier à un utilisateur spécifique.

```
root@sli-haproxy-1 /opt/scripts:# ls -al
total 32
drwxr-xr-x 3 root root  4096 mars  28 17:02 .
drwxr-xr-x 3 root root  4096 mars  21 10:32 ..
-rwxr-xr-x 1 root root  3371 mars  21 11:51 apiip.php
drwxr-xr-x 3 root root  4096 mars  28 14:39 dossier
-rw-r--r-- 1 root root 10240 mars  28 17:02 resultat.tar
-rwxrwx--- 1 root root  3371 mars  28 13:52 resultat.txt
root@sli-haproxy-1 /opt/scripts:# chown user resultat.tar
root@sli-haproxy-1 /opt/scripts:# ls -al
total 32
drwxr-xr-x 3 root root  4096 mars  28 17:02 .
drwxr-xr-x 3 root root  4096 mars  21 10:32 ..
-rwxr-xr-x 1 root root  3371 mars  21 11:51 apiip.php
drwxr-xr-x 3 root root  4096 mars  28 14:39 dossier
-rw-r--r-- 1 user root 10240 mars  28 17:02 resultat.tar
-rwxrwx--- 1 root root  3371 mars  28 13:52 resultat.txt
```

NB : Les deux colonnes représentent l'utilisateur et le groupe propriétaire du fichier/répertoire. Nous verrons plus en détails plus tard.

JOBS

> La commande **jobs** permet d'afficher les jobs actuels avec leur statut. Un job est un processus qui est lancé par le shell.

```
root@sli-haproxy-1 ~:# ping 1.1.1.1 > /dev/null &
[2] 8477
root@sli-haproxy-1 ~:# jobs
[1]+  Stoppé                 vi test.txt
[2]-  En cours d'exécution   ping 1.1.1.1 > /dev/null &
```

```
root      8470  0.0  0.3  15488  6132 pts/0    T    17:21   0:00 vi test.txt
root      8477  0.0  0.0   9060  1212 pts/0    S    17:24   0:00 ping 1.1.1.1
root      8482  0.0  0.1  10632  3068 pts/0    R+   17:26   0:00 ps auxwww
```

PS

La commande **ps** permet d'afficher les processus en cours sur la machine.

```
root@sli-haproxy-1 ~:# ps aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.5 105104 10448 ?        Ss   mars12   0:13 /sbin/init
root         2  0.0  0.0      0     0 ?        S    mars12   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        I<   mars12   0:00 [rcu_gp]
root         4  0.0  0.0      0     0 ?        I<   mars12   0:00 [rcu_par_gp]
root         6  0.0  0.0      0     0 ?        I<   mars12   0:00 [kworker/0:0H-kblockd]
root         8  0.0  0.0      0     0 ?        I<   mars12   0:00 [mm_percpu_wq]
root         9  0.0  0.0      0     0 ?        S    mars12   0:02 [ksoftirqd/0]
root        10  0.0  0.0      0     0 ?        I    mars12   1:22 [rcu_sched]
```

NB : L'argument « **aux** » permet d'afficher les processus de tous les utilisateurs (**a**), plus d'information (**u**) et liste les processus qui n'appartient à aucune TTY (**x**).

KILL

La commande **kill** permet de tuer un processus qui ne répond plus par exemple.

```
root        8393  0.0  0.0   2456  1576 ?        Ss   17:10   0:00 /usr/lib/openssh/sftp-server
root        8414  0.0  0.1   8904  2660 ?        Ss   17:10   0:00 SCREEN -S test
root        8415  0.0  0.1   7172  3852 pts/2    Ss+  17:10   0:00 /bin/bash
root        8434  0.0  0.4  17204  8408 ?        Ss   17:15   0:00 sshd: root@pts/0
root        8436  0.0  0.4  16900  8284 ?        Ss   17:15   0:00 sshd: root@notty
root        8442  0.0  0.2   8116  4972 pts/0    Ss   17:15   0:00 -bash
root        8451  0.0  0.0   2456  1528 ?        Ss   17:15   0:00 /usr/lib/openssh/sftp-server
root        8483  0.0  0.0      0     0 ?        I    17:27   0:00 [kworker/0:1-ata_sff]
root        8490  0.0  0.0      0     0 ?        I    17:32   0:00 [kworker/0:2-ata_sff]
root        8496  0.0  0.1  10632  3124 pts/0    R+   17:34   0:00 ps aux
root@sli-haproxy-1 ~:# kill -9 8414
```

NB : Il existe 64 signaux, mais on utilise généralement :

- 15 : demande au programme de s'arrêter en enregistrant le travail
- 9 : force l'extinction du programme

PING

La commande **ping** permet de vérifier l'état de connectivité à un hôte.

```
root@sli-haproxy-1 ~:# ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=57 time=3.20 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=57 time=2.74 ms
^C
--- 1.1.1.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 2ms
rtt min/avg/max/mdev = 2.737/2.970/3.204/0.239 ms
```

NB : On peut arrêter la commande via le clavier : « **Ctrl + c** ».

WGET

La commande **wget** permet de télécharger des fichiers via le protocole HTTP.

```
root@sli-haproxy-1 /opt/scripts:# wget https://www.python.org/ftp/python/3.8.1/Python-3.8.1.tgz
--2022-03-28 17:46:07--  https://www.python.org/ftp/python/3.8.1/Python-3.8.1.tgz
Résolution de www.python.org (www.python.org)… 151.101.240.223, 2a04:4e42:39::223
Connexion à www.python.org (www.python.org)|151.101.240.223|:443… connecté.
requête HTTP transmise, en attente de la réponse… 200 OK
Taille : 23978360 (23M) [application/octet-stream]
Sauvegarde en : « Python-3.8.1.tgz »

Python-3.8.1.tgz                                  100%[=============================================================================================================>]  22,87M  37,1MB/s    ds 0,6s

2022-03-28 17:46:08 (37,1 MB/s) – « Python-3.8.1.tgz » sauvegardé [23978360/23978360]

root@sli-haproxy-1 /opt/scripts:# ls -al
total 23452
drwxr-xr-x 3 root root     4096 mars  28 17:46 .
drwxr-xr-x 3 root root     4096 mars  21 10:32 ..
-rwxr-xr-x 1 root root     3371 mars  21 11:51 apiip.php
drwxr-xr-x 3 root root     4096 mars  28 14:39 dossier
-rw-r--r-- 1 root root 23978360 déc.  18  2019 Python-3.8.1.tgz
-rw-r--r-- 1 user root    10240 mars  28 17:02 resultat.tar
-rwxrwx--- 1 root root     3371 mars  28 13:52 resultat.txt
```

UNAME

> La commande **uname** permet de donner les informations détaillées de votre système Linux.

```
root@sli-haproxy-1 /opt/scripts:# uname -a
Linux sli-haproxy-1 4.19.0-12-amd64 #1 SMP Debian 4.19.152-1 (2020-10-18) x86_64 GNU/Linux
root@sli-haproxy-1 /opt/scripts:#
```

TOP

> La commande **top** permet d'afficher une liste des processus qui sont en cours en temps réel.

```
top - 17:50:39 up 16 days,  1:32,  2 users,  load average: 0,00, 0,00, 0,00
Tasks: 102 total,   1 running, 101 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0,0 us,  0,0 sy,  0,0 ni,100,0 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
MiB Mem :   1994,3 total,   1342,4 free,     88,7 used,    563,2 buff/cache
MiB Swap:   2048,0 total,   2048,0 free,      0,0 used.   1709,1 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 8563 root      20   0   11108   3560   3072 R   6,7   0,2   0:00.01 top
    1 root      20   0  105104  10448   7872 S   0,0   0,5   0:13.99 systemd
    2 root      20   0       0      0      0 S   0,0   0,0   0:00.22 kthreadd
    3 root       0 -20       0      0      0 I   0,0   0,0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0,0   0,0   0:00.00 rcu_par_gp
    6 root       0 -20       0      0      0 I   0,0   0,0   0:00.00 kworker/0:0H-kblockd
    8 root       0 -20       0      0      0 I   0,0   0,0   0:00.00 mm_percpu_wq
    9 root      20   0       0      0      0 S   0,0   0,0   0:02.92 ksoftirqd/0
   10 root      20   0       0      0      0 I   0,0   0,0   1:22.32 rcu_sched
   11 root      20   0       0      0      0 I   0,0   0,0   0:00.00 rcu_bh
   12 root      rt   0       0      0      0 S   0,0   0,0   0:02.45 migration/0
   14 root      20   0       0      0      0 S   0,0   0,0   0:00.00 cpuhp/0
```

HISTORY

La commande **history** permet d'afficher l'historique des commandes saisies.

```
root@sli-haproxy-1 /opt/scripts:# history
    1  vi /etc/hostname
    2  vi /etc/network/interfaces
    3  reboot
    4  ha
    5  apt-get install certbot
    6  ~/.secrets/certbot/ovh.ini
    7  vi ~/.secrets/certbot/ovh.ini
    8  mkdir .secrets
```

MAN

> La commande **man** permet d'afficher le manuelle concernant la commande demandée.

```
PING(8)                                                                 iputils                                                                 PING(8)

NAME
       ping - send ICMP ECHO_REQUEST to network hosts

SYNOPSIS
       ping [-aAbBdDfhLnOqrRUvV46] [-c count] [-F flowlabel] [-i interval] [-I interface] [-l preload] [-m mark] [-M pmtudisc_option] [-N nodeinfo_option] [-w deadline] [-W timeout] [-p pattern]
            [-Q tos] [-s packetsize] [-S sndbuf] [-t ttl] [-T timestamp option] [hop...] destination

DESCRIPTION
       ping uses the ICMP protocol's mandatory ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams (pings) have an IP and ICMP header, followed by
       a struct timeval and then an arbitrary number ofpadbytes used to fill out the packet.

       ping works with both IPv4 and IPv6. Using only one of them explicitly can be enforced by specifying -4 or -6.
```

ECHO

La commande **echo** permet d'afficher du texte dans le shell, mais utilisé surtout pour ajouter du texte dans un fichier.

```
root@sli-haproxy-1 /opt/scripts:# touch FichierVide
root@sli-haproxy-1 /opt/scripts:# cat FichierVide
root@sli-haproxy-1 /opt/scripts:# echo "Fichier plus vide" > FichierVide
root@sli-haproxy-1 /opt/scripts:# cat FichierVide
Fichier plus vide
```

NB : Attention, le chevron « **>** » efface le contenu du fichier cible. A la différence du double chevron « **>>** » qui permet d'ajouter à la fin du fichier.

ZIP & UNZIP

Les commandes **zip** et **unzip** permettent de compresser et décompresser les fichiers en une archive ZIP et inversement.

```
root@sli-haproxy-1 /opt/scripts:# zip FichierVide.zip FichierVide
  adding: FichierVide (stored 0%)
root@sli-haproxy-1 /opt/scripts:# ll
total 20
drwxr-xr-x 2 root root 4096 mars  28 17:59 .
drwxr-xr-x 3 root root 4096 mars  21 10:32 ..
-rwxr-xr-x 1 root root 3371 mars  21 11:51 apiip.php
-rw-r--r-- 1 root root   18 mars  28 17:55 FichierVide
-rw-r--r-- 1 root root  190 mars  28 17:59 FichierVide.zip
```

```
root@sli-haproxy-1 /opt/scripts:# rm -f FichierVide
root@sli-haproxy-1 /opt/scripts:# ll
total 16
drwxr-xr-x 2 root root 4096 mars  28 18:00 .
drwxr-xr-x 3 root root 4096 mars  21 10:32 ..
-rwxr-xr-x 1 root root 3371 mars  21 11:51 apiip.php
-rw-r--r-- 1 root root  190 mars  28 17:59 FichierVide.zip
root@sli-haproxy-1 /opt/scripts:# unzip FichierVide.zip
Archive:  FichierVide.zip
 extracting: FichierVide
root@sli-haproxy-1 /opt/scripts:# ls -al
total 20
drwxr-xr-x 2 root root 4096 mars  28 18:00 .
drwxr-xr-x 3 root root 4096 mars  21 10:32 ..
-rwxr-xr-x 1 root root 3371 mars  21 11:51 apiip.php
-rw-r--r-- 1 root root   18 mars  28 17:55 FichierVide
-rw-r--r-- 1 root root  190 mars  28 17:59 FichierVide.zip
```

NANO

La commande **nano** permet d'ouvrir un éditeur de texte.

HOSTNAME

La commande **hostname** permet d'obtenir des informations comme le nom de la machine.

```
root@sli-haproxy-1 /opt/scripts:# hostname
sli-haproxy-1
root@sli-haproxy-1 /opt/scripts:# hostname -i
10.0.40.108
```

HOSTNAME

La commande **hostname** permet d'obtenir des informations comme le nom de la machine.

```
root@sli-haproxy-1 /opt/scripts:# hostname
sli-haproxy-1
root@sli-haproxy-1 /opt/scripts:# hostname -i
10.0.40.108
```

USERADD

La commande **useradd** permet de créer un nouvel utilisateur.

```
root@sli-haproxy-1 /opt/scripts:# useradd tech
root@sli-haproxy-1 /opt/scripts:# tail /etc/passwd -n 3
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
haproxy:x:106:112::/var/lib/haproxy:/usr/sbin/nologin
tech:x:1001:1001::/home/tech:/bin/sh
```

PASSWD

La commande **passwd** permet de modifier le mot de passe d'un utilisateur (ou de l'utilisateur local).

```
root@sli-haproxy-1 /opt/scripts:# passwd tech
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd: password updated successfully
```

USERDEL

La commande **userdel** permet de supprimer un utilisateur

```
root@sli-haproxy-1 /opt/scripts:# userdel tech
root@sli-haproxy-1 /opt/scripts:# tail /etc/passwd -n 3
user:x:1000:1000:user,,,:/home/user:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
haproxy:x:106:112::/var/lib/haproxy:/usr/sbin/nologin
```

Récapitulatif

pwd
cd
ls
cat
cp
mv
mkdir
rmdir
rm
touch
locate
find

grep
sudo
df
du
head
tail
tar
chmod
chown
jobs
ps
kill

ping
wget
uname
top
history
man
echo
zip & unzip
nano
hostname
useradd
passwd

A vos VM !

Pratique

Sur votre machine Debian :

- Créer un répertoire /opt/tp
- Créer un fichier vide : *fichier1.txt*
- Créer un fichier vide *ficher2.txt* et lui ajouter « *Le fichier n'est plus vide* »
- Créer un utilisateur « *technique* » et lui saisir un mot de passe
- Créer un fichier fichier3.txt vide dans /root/ et le déplacer vers /opt/tp
- Créer un fichier fichier4.txt vide dans /opt/tp et le copier vers /root/
- Supprimer le fichier ficher1.txt se trouvant /opt/tp
- Supprimer le répertoire /opt/tp
- Afficher l'historique des commandes

A vos VM !

Pratique

Sur votre machine Debian :

- Visualiser la configuration réseau
  - Récupérer votre adresse IP
  - Récupérer votre masque de sous réseau
  - Récupérer votre passerelle par défaut

- Modifier votre configuration réseau en statique par l'adresse IP, le masque et la passerelle que vous avez récupéré.

- Vérifier la connectivité.

N.B. : le fichier de configuration du réseau sous Debian se trouve dans **/etc/network/interfaces**.

A vos VM !

Pratique

Sur votre machine Debian :

- Vérifier que le démon ssh tourne : **systemctl status sshd**
- Si c'est le cas, vous pouvez vous connecter en SSH depuis votre PC en indiquant l'adresse IP de la machine et votre prénom comme nom d'utilisateur
  - Depuis putty
  - Depuis Mobaxterm
  - Depuis une CMD Windows :
    - **ssh votre-prénom@ip-du-serveur**
- Si le démon sshd ne tourne pas, il faut le lancer : **systemctl start sshd**
- Si le démon sshd n'est pas installé, il faut lancer : **apt-get install openssh-server**

5. Les utilisateurs et groupes

Les utilisateurs sur Linux

Le fichier **/etc/passwd** contient l'ensemble des informations qui régissent la connexion des utilisateurs.

```
root@sli-haproxy-1 ~:# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
```

- « **root** » est l'id de connexion
- « **x** » signifie que le mot de passe chiffré se situe dans le fichier /etc/shadow
- « **0** » correspond à l'UID (que le système utilise pour gérer les droits d'accès)
- « **0** » correspond au GID (groupe utilisateur)
- « **root** » est le nom complet de l'utilisateur
- « **/root** » correspond au répertoire de connexion de l'utilisateur
- « **/bin/bash** » correspond au shell de l'utilisateur (la commande que le système exécute à la connexion de l'utilisateur)

Les utilisateurs sur Linux

- Les utilisateurs doivent être membre d'au moins un groupe (son groupe principal)
- Ils peuvent appartenir à d'autres groupes (groupes secondaires)
- Il y a 3 types d'utilisateurs :
  - **superutilisateur** (root) : compte unique et disposant de toutes les autorisations.
  - **système** : compte créés par le système et services, ils servent à faire fonctionner certains services du système comme l'impression, le son, les sessions des utilisateurs…
  - **normal** : compte utilisateurs utilisé par des personnes physiques.

Les utilisateurs sur Linux

Le fichier **/etc/shadow** contient les mots de passe de connexion des utilisateurs. Seul root à accès à ce fichier.

```
root@sli-haproxy-1 ~:# cat /etc/shadow
root:$6$8nC6ni226fqRq3NH$Elve6clQOxI.X9S87C4MaTF.mccKFwyr0jBTC2O5KCm53wVj1bXaCudx/ZAehV3wNKsrzwE./M5k0.qBanEr8.:18589:0:99999:7:::
daemon:*:18589:0:99999:7:::
bin:*:18589:0:99999:7:::
sys:*:18589:0:99999:7:::
sync:*:18589:0:99999:7:::
games:*:18589:0:99999:7:::
man:*:18589:0:99999:7:::
lp:*:18589:0:99999:7:::
```

- « **root** » est l'id de connexion
- « **$6….** » est le mot de passe hashé
- « **18589** » correspond à la date du dernier changement du mot de passe
- « **0** » correspond à l'âge minimum du mot de passe
- « **99999** » correspond à l'âge maximum du mot de passe
- « **7** » correspond à la période d'expiration du mot de passe

Les groupes sur Linux

Le fichier **/etc/group** contient la liste des utilisateurs appartenant aux différents groupes.

Il contient :

- Le nom du groupe.
- Un champ spécial (généralement vide).
- L'identifiant du groupe (GID).
- Les membres du groupe.

Exemple :

news:x:13:news,fabio

6. Le système de fichiers

Le partitionnement

Le partitionnement va permettre l'installation de plusieurs systèmes d'exploitation car il est impossible d'en faire cohabiter plusieurs sur un même lecteur logique.
Le partitionnement permet également de cloisonner des données (sécurité, optimisation d'accès, ...).

Les devices, ou périphériques, sont les fichiers identifiant les différents matériels détectés par la carte mère. Ces fichiers sont stockés dans **/dev**. Le service qui détecte les nouveaux périphériques et leur donne des noms s'appelle « **udev** ».

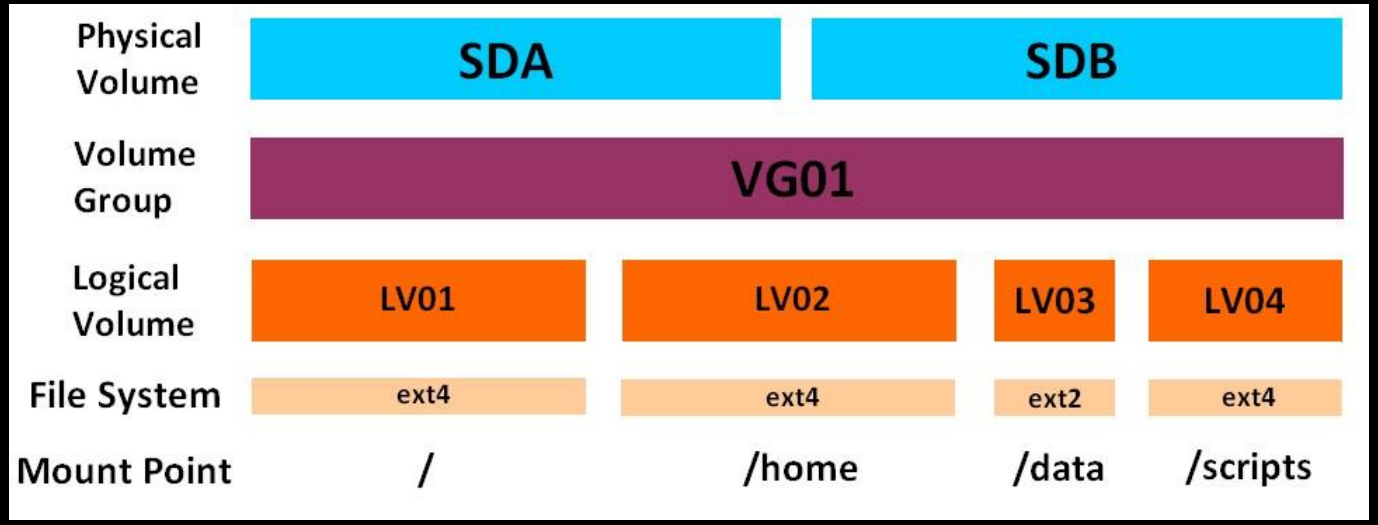
Les périphériques de stockage se nomment **hd** pour les disques durs IDE et **sd** pour les autres supports.
Puis, il y a une lettre qui commence par **a** pour le premier périphérique, puis **b**, **c**, ... (sda, sdb…)
Enfin il y a un chiffre qui définit le volume partitionné : **1** pour la première partition (sda1, sda2…)

Stockage via LVM

LVM (Logical Volume Manager) répond principalement aux besoins :

- d'**évolutivité des capacités de stockage**
- tout en assurant la **disponibilité du service.**

Plus simplement il s'agit de **redimensionner un système de fichiers (FS) dynamiquement** (en augmentant ou en réduisant le nombre de disques physiques disponibles) avec un minimum d'interruption.



Un article a été rédigé par moi-même à ce sujet : https://net-security.fr/system/les-commandes-lvm/

Structure d’un système de fichiers

Un système de fichiers est en charge des actions suivantes :

- Sécuriser les droits d’accès et de modification des fichiers.
- Manipuler des fichiers : créer, lire, modifier et supprimer.
- Localiser les fichiers sur le disque.
- Gérer l’espace mémoire.

Le système d’exploitation Linux est capable d’exploiter différents systèmes de fichiers (ext2, ext3, ext4, FAT16, FAT32, NTFS, HFS, BtrFS, JFS, XFS, ...).

- **Bloc de boot :** occupe le premier bloc sur le disque et présent sur chaque partition (de démarrage). Il contient le programme assurant le démarrage et l’initialisation du système

- **Super bloc** : présent sur chaque partition et décrit le système de fichiers (nom du volume, nom du système de fichiers, type du FS, état du FS, taille du FS…)

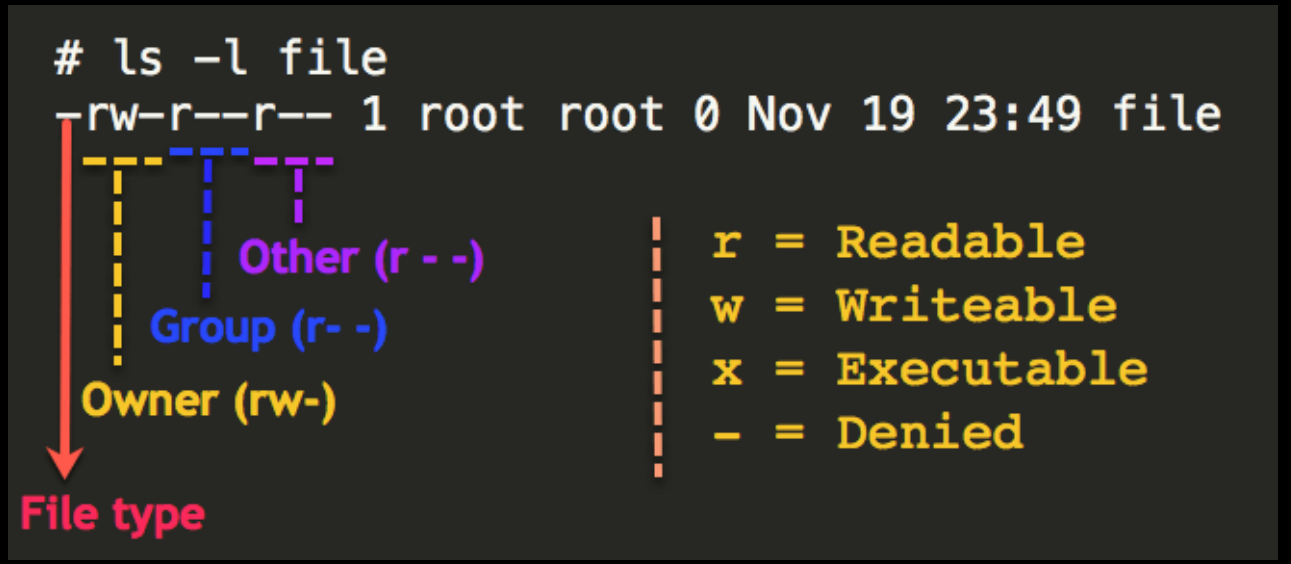
- **Table des inodes** : tout fichier possède son unique inode et correspond aux permissions, au propriétaire, groupe propriétaire, date dernier accès…. L'inode contient la totalité des informations sur le fichier, sauf le nom et sont tous de même taille.

- **Zone de données :** le reste de l’espace disponible de la partition

7. Les permissions

Le principe de base

- Seul le propriétaire du fichier (**u**) et le super utilisateur (root) peuvent changer les droits
- Un groupe (**g**) peut aussi avoir des droits particulier sur le fichier
- Enfin, on regroupe « les autres » (**o**) pour accorder des privilèges spécifique sur le fichier



- NB : Un dossier (ou périphérique) est aussi un fichier

https://www.sumitbera.com

Le changement de propriétaire

- Via la commande « **chown** », on peut modifier le propriétaire du fichier

```
root@sli-haproxy-1 /opt/scripts:# ll
total 24
drwxr-xr-x 2 root root 4096 mars  28 19:01 .
drwxr-xr-x 3 root root 4096 mars  21 10:32 ..
-rwxr-xr-x 1 root root 3371 mars  21 11:51 apiip.php
-rw-r--r-- 1 root root   18 mars  28 17:55 FichierVide
-rw-r--r-- 1 root root   50 mars  28 19:01 FichierVide.save
-rw-r--r-- 1 root root  190 mars  28 17:59 FichierVide.zip
root@sli-haproxy-1 /opt/scripts:# chown user FichierVide
root@sli-haproxy-1 /opt/scripts:# ls -al
total 24
drwxr-xr-x 2 root root 4096 mars  28 19:01 .
drwxr-xr-x 3 root root 4096 mars  21 10:32 ..
-rwxr-xr-x 1 root root 3371 mars  21 11:51 apiip.php
-rw-r--r-- 1 user root   18 mars  28 17:55 FichierVide
-rw-r--r-- 1 root root   50 mars  28 19:01 FichierVide.save
-rw-r--r-- 1 root root  190 mars  28 17:59 FichierVide.zip
```

- On peut aussi changer le groupe propriétaire via la commande chown

```
root@sli-haproxy-1 /opt/scripts:# ll
total 24
drwxr-xr-x 2 root root 4096 mars  28 19:01 .
drwxr-xr-x 3 root root 4096 mars  21 10:32 ..
-rwxr-xr-x 1 root root 3371 mars  21 11:51 apiip.php
-rw-r--r-- 1 root root   18 mars  28 17:55 FichierVide
-rw-r--r-- 1 root root   50 mars  28 19:01 FichierVide.save
-rw-r--r-- 1 root root  190 mars  28 17:59 FichierVide.zip
root@sli-haproxy-1 /opt/scripts:# chown user:user FichierVide
root@sli-haproxy-1 /opt/scripts:# ll
total 24
drwxr-xr-x 2 root root 4096 mars  28 19:01 .
drwxr-xr-x 3 root root 4096 mars  21 10:32 ..
-rwxr-xr-x 1 root root 3371 mars  21 11:51 apiip.php
-rw-r--r-- 1 user user   18 mars  28 17:55 FichierVide
-rw-r--r-- 1 root root   50 mars  28 19:01 FichierVide.save
-rw-r--r-- 1 root root  190 mars  28 17:59 FichierVide.zip
```

Le changement de groupe propriétaire

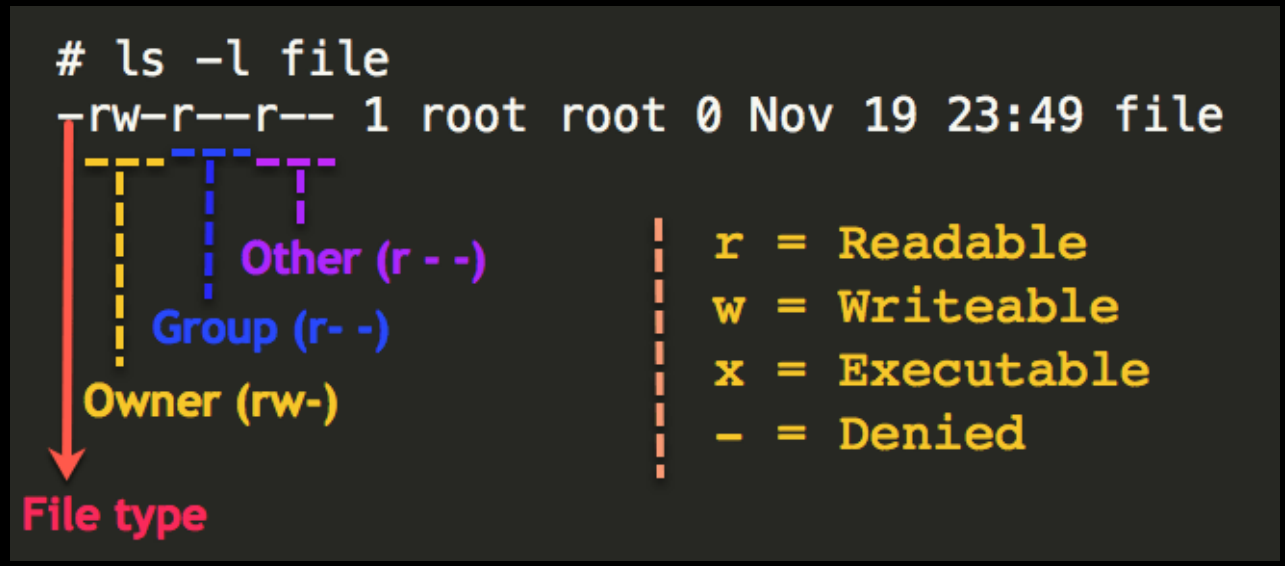
- Via la commande « **chgrp** », on peut modifier le groupe propriétaire du fichier

```
root@sli-haproxy-1 /opt/scripts:# ll
total 24
drwxr-xr-x 2 root root 4096 mars  28 19:01 .
drwxr-xr-x 3 root root 4096 mars  21 10:32 ..
-rwxr-xr-x 1 root root 3371 mars  21 11:51 apiip.php
-rw-r--r-- 1 user user   18 mars  28 17:55 FichierVide
-rw-r--r-- 1 root root   50 mars  28 19:01 FichierVide.save
-rw-r--r-- 1 root root  190 mars  28 17:59 FichierVide.zip
root@sli-haproxy-1 /opt/scripts:# chgrp root FichierVide
root@sli-haproxy-1 /opt/scripts:# ll
total 24
drwxr-xr-x 2 root root 4096 mars  28 19:01 .
drwxr-xr-x 3 root root 4096 mars  21 10:32 ..
-rwxr-xr-x 1 root root 3371 mars  21 11:51 apiip.php
-rw-r--r-- 1 user root   18 mars  28 17:55 FichierVide
-rw-r--r-- 1 root root   50 mars  28 19:01 FichierVide.save
-rw-r--r-- 1 root root  190 mars  28 17:59 FichierVide.zip
```

Les droits

- Les permissions se déclinent en trois possibilité :
  - Lecture (**r**) : liste du contenu d'un dossier, et lecture d'un fichier
  - Ecriture (**w**) : modifier contenu et supprime le fichier
  - Exécution (**x**) : exécute si fichier exécutable (script par exemple)

On appelle **RWX** des drapeaux.



La représentation octale

Symbole	Octal	Binaire
r	4	100
w	2	010
x	1	001

7 : r w x
6 : r w -
5: r - x
4 : r - -
3 : - w x
2 : - w -
1 : - - x

u : RWX
g : RW-
o : - - -

u : 7
g : 6
o : 0

760

Le changement de permission des fichiers et dossiers

u : RWX
g : RW-
o : ---

u : 7
g : 6
o : 0

760

chmod 760 fichier

chmod u+rw fichier
chmod g+rw fichier
chmod o-r

La représentation symbolique

Type utilisateur	
User	u
Group	g
Other	o
All	a

Opérateur	
Ajouter	+
Enlever	-
Remplacer	=

Droits	
Read	r
Write	w
eXecute	X

chmod u+rwx,g=wx,o-r /tmp/fichier

ls -l /tmp/fichier

-rwx-wx--- 1 root root ... /tmp/fichier

Les droits particuliers

suid	sgid	sticky-bit
s	s	t
4	2	1

user			group			other		
r	w	w	r	w	w	r	w	w
4	2	1	4	2	1	4	2	1

- set-user-ID (SUID) : n'importe quell utilisateur peut éxécuter en héritant des permissions du propriétaire. Ce droit ne s'active que sur les fichiers.

- set-group-ID (SGID) :
  - Sur un fichier : similaire au SUID mais appliqué aux groupes.
  - Sur un repertoires : hérite le droit du groupe du repertoire parent (et non celui de l'utilisateur créant le répertoire).

- sticky-bit :
  - Sur un fichier : Gnu/Linux n'en tient pas compte. Ce droit dans les vieux systèmes Unix indique que le fichier sera utilisé fréquement et sera donc stocké dans un fichier d'échange.
  - Sur un répertoire : permet d'éviter la suppression d'un fichier s'il n'est pas le propriétaire de celui-ci.

A vos VM !

Pratique

Sur votre machine Debian :

- Créez le script suivant : testscript.sh

```
#!/bin/bash

echo « Test script shell »
exit
```

- Vérifiez les permissions du fichier
- Essayez d'exécuter le script
- Rendez le scripts exécutable
- N'accordez les droits qu'au seul propriétaire

8. Les points de montage

Le contenu du système de fichier

- **/bin** et **/sbin :** binaires nécessaire au démarrage + commandes essentielles
- **/dev :** fichier périphériques
- **/etc :** fichiers de configuration
- **/lib :** contient les modules du noyaux + biblio partagée avec /bin et /sbin
- **/mnt** ou **/media :** points de montage des systèmes externes
- **/proc :** information du noyau
- **/boot :** noyau Linux
- **/home :** dossier utilisateur
- **/root :** dossier utilisateur root
- **/tmp :** fichiers temporaires
- **/usr :** contenu statique et partagé concernant les bibliothèques système non essentielles
- **/opt :** dossier où l'on place nos scripts
- **/var :** données variage (journaux)

Les commandes mount et umount

Mount permet de relier une partition ou un périphérique à un répertoire.

mount /dev/cdrom /media/cdrom

Umount permet de délier une partition ou un périphérique à un répertoire.

umount /media/cdrom

NB : *Aucun fichier ne doit être ouvert, ni aucun processus en cours de fonctionnement sur la partition.*

Lister des partitions

La commande fdisk –l permet de lister les partitions :

```
root@sli-haproxy-1 ~:# fdisk -l
Disque /dev/sda : 25 GiB, 26843545600 octets, 52428800 secteurs
Modèle de disque : Virtual disk
Unités : secteur de 1 × 512 = 512 octets
Taille de secteur (logique / physique) : 512 octets / 512 octets
taille d'E/S (minimale / optimale) : 512 octets / 512 octets
Type d'étiquette de disque : dos
Identifiant de disque : 0xb6091e3e

Périphérique Amorçage  Début      Fin Secteurs Taille Id Type
/dev/sda1    *          2048   499711   497664   243M 83 Linux
/dev/sda2             501758 52426751 51924994  24,8G  5 Étendue
/dev/sda5             501760 52426751 51924992  24,8G 8e LVM Linux


Disque /dev/mapper/TEMPLATE--DEBIAN--vg-root : 22,8 GiB, 24436015104 octets, 47726592 secteurs
Unités : secteur de 1 × 512 = 512 octets
Taille de secteur (logique / physique) : 512 octets / 512 octets
taille d'E/S (minimale / optimale) : 512 octets / 512 octets


Disque /dev/mapper/TEMPLATE--DEBIAN--vg-swap_1 : 2 GiB, 2147483648 octets, 4194304 secteurs
Unités : secteur de 1 × 512 = 512 octets
Taille de secteur (logique / physique) : 512 octets / 512 octets
taille d'E/S (minimale / optimale) : 512 octets / 512 octets
```

Lister des partitions

```
root@sli-haproxy-1 ~:# blkid
/dev/sda1: UUID="0d323086-1a1d-43ed-9f79-8ed35197dd21" TYPE="ext2" PARTUUID="b6091e3e-01"
/dev/sda5: UUID="Dbp4NT-khrq-VUvD-xN1b-F1I2-wKjR-Dbn4pc" TYPE="LVM2_member" PARTUUID="b6091e3e-05"
/dev/sr0: UUID="2020-09-26-10-23-56-00" LABEL="Debian 10.6.0 amd64 1" TYPE="iso9660" PTUUID="6f77041c" PTTYPE="dos"
/dev/mapper/TEMPLATE--DEBIAN--vg-root: UUID="2f82391c-8ed2-4d68-9567-e8348bffebd0" TYPE="ext4"
/dev/mapper/TEMPLATE--DEBIAN--vg-swap_1: UUID="7ede5513-da8c-475c-a2bb-12943bb50af0" TYPE="swap"
```

```
root@sli-haproxy-1 ~:# lsblk -fe7
NAME                         FSTYPE      LABEL                 UUID                                   FSAVAIL FSUSE% MOUNTPOINT
sda
├─sda1                       ext2                              0d323086-1a1d-43ed-9f79-8ed35197dd21    139,5M    36% /boot
├─sda2
└─sda5                       LVM2_member                       Dbp4NT-khrq-VUvD-xN1b-F1I2-wKjR-Dbn4pc
  ├─TEMPLATE--DEBIAN--vg-root ext4                             2f82391c-8ed2-4d68-9567-e8348bffebd0     19,4G     8% /
  └─TEMPLATE--DEBIAN--vg-swap_1 swap                           7ede5513-da8c-475c-a2bb-12943bb50af0                  [SWAP]
sr0                          iso9660     Debian 10.6.0 amd64 1 2020-09-26-10-23-56-00
```

Le fichier fstab

Il liste les partitions qui seront montées automatiquement au démarrage.

Il liste les partitions qui seront montées automatiquement au démarrage.

```
# <file system>   <mount point>    <type>    <options>    <dump>    <pass>

  /dev/sdb2       /media/toto      ext4      defaults     0         2

  UUID=33b870b8-a81e-4203-a4fd-7affa9f412fb    /media/toto    ext4    defaults  0  2
```

mount -a

9. Le système de paquet

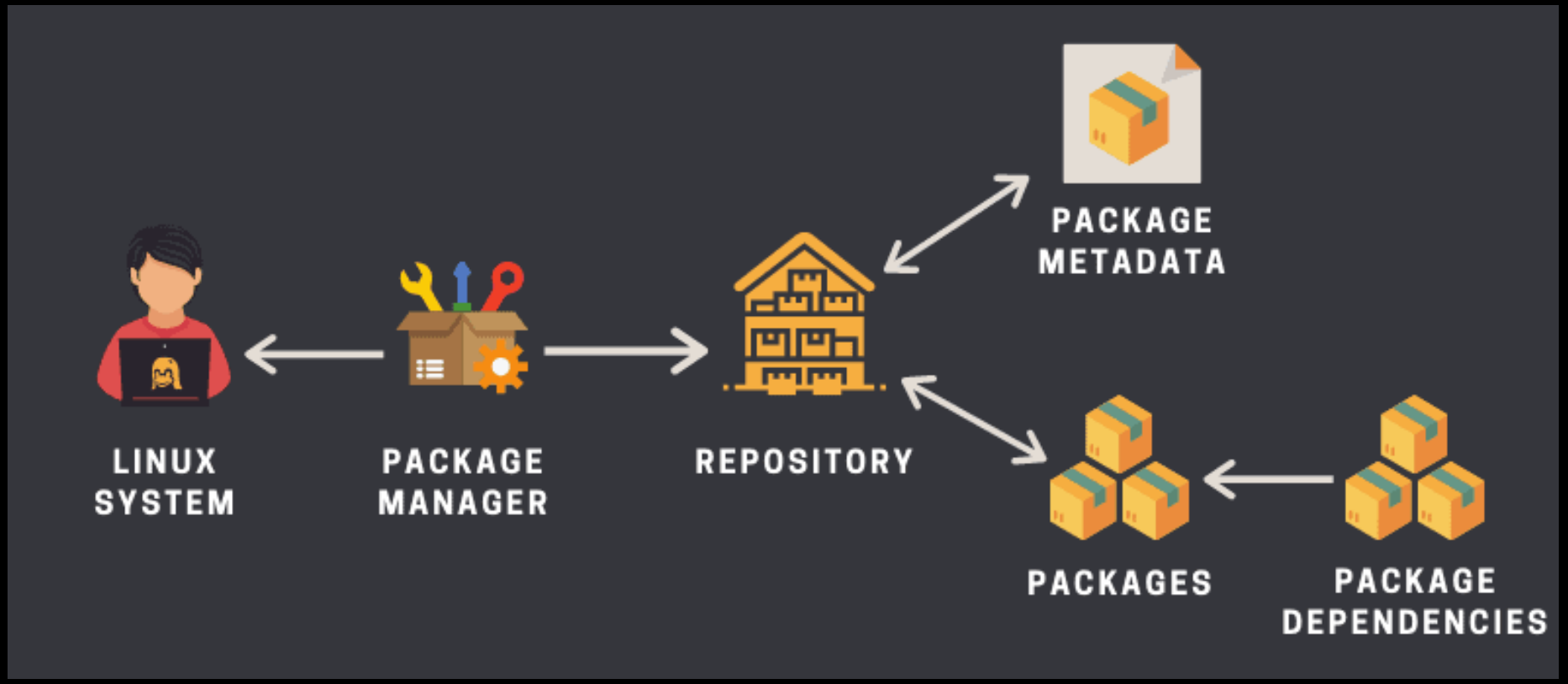
Un paquet Debian

- Un paquet contient tous les fichiers nécessaires pour implémenter un ensemble de commandes ou de fonctionnalités.
- Un paquet est donc un logiciel.
- Un paquet utilise des dépendances pour fonctionner.
- Les dépendances sont définies par le responsable du paquet.

Les métadonnées de paquets

- Un résumé
- Une description
- Une liste des fichiers présents dans le paquet
- La version du logiciel contenu ainsi que le numéro de version du paquet
- Quand, où et par qui il a été intégré
- L'architecture pour laquelle il a été conçu
- La somme de contrôle des fichiers présents dans le paquet
- La licence qui régit le paquet
- Les autres paquets requis pour fonctionner proprement
- …

Concept gestion de paquet



Source : https://maniacgeek.net

Quelques commandes bas niveau

Pour lister tous les paquets installés avec des droits priviliégés
- ***dpkg –l***

Pour vérifier qu'un paquet soit installé
- ***dpkg –s wget***

Pour lister les fichiers installés par un paquet
- ***dpkg –L wget***

Pour installer ou désinstaller un paquet .deb : utilisez plutôt apt ou aptitude
- ***dpkg –i***
- ***dpkg –r***

L’utilisation d’APT

Source d’apt pour rechercher les paquets

- ***cat /etc/apt/sources.list***

```
deb http://ftp.fr.debian.org/debian/ buster main
deb-src http://ftp.fr.debian.org/debian/ buster main

deb http://security.debian.org/debian-security buster/updates main
deb-src http://security.debian.org/debian-security buster/updates main

# buster-updates, previously known as 'volatile'
deb http://ftp.fr.debian.org/debian/ buster-updates main
deb-src http://ftp.fr.debian.org/debian/ buster-updates main
```

L’utilisation d’APT – La recherche

Recherche dans les descriptions de paquets
- ***apt-cache search wget***

Voir les informations d’un paquet
- ***apt-cache show wget***

Vérifier les dépendances d’un paquet
- ***apt-cache showpgk wget***

L’utilisation d’APT – La mise à jour et l’installation

Mise à jour de la liste de paquets

- ***apt-get update***

Mettre à jour tous les paquets sans ajout de nouveaux paquets

- ***apt-get upgrade***

Mettre à jour tous les paquets installés vers les dernières versions en installant de nouveaux paquets si nécessaire

- ***apt dist-upgrade***

Installation ou mise à jour d’un paquet

- ***apt install wget***

L’utilisation d’APT – La désinstallation

Retirer les paquets sans les configurations
- ***apt remove wget***

Retirer les paquets sans les dépendances
- ***apt autoremove wget***

Retirer totalement un paquet
- ***apt purge wget***

Suppression des fichiers mis en cache dans /var/cache/apt/archives
- ***apt clean***

A vos VM !

Pratique

Sur vos machines Debian :

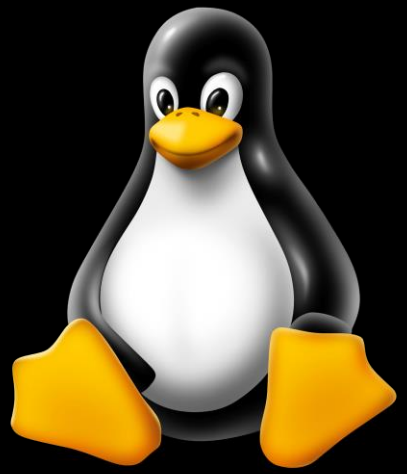
- Sur le serveur :
  - Créer un repo local en clonant seulement le dépôt
    “*deb http://ftp.fr.debian.org/debian/ buster main contrib non-free*”
  - La synchro devra être faite tous les jours à ooh

- Sur le client :
  - Configurer seulement le depot local
  - Essayer de faire une MAJ

10. La première stack applicative : LAMP

LAMP

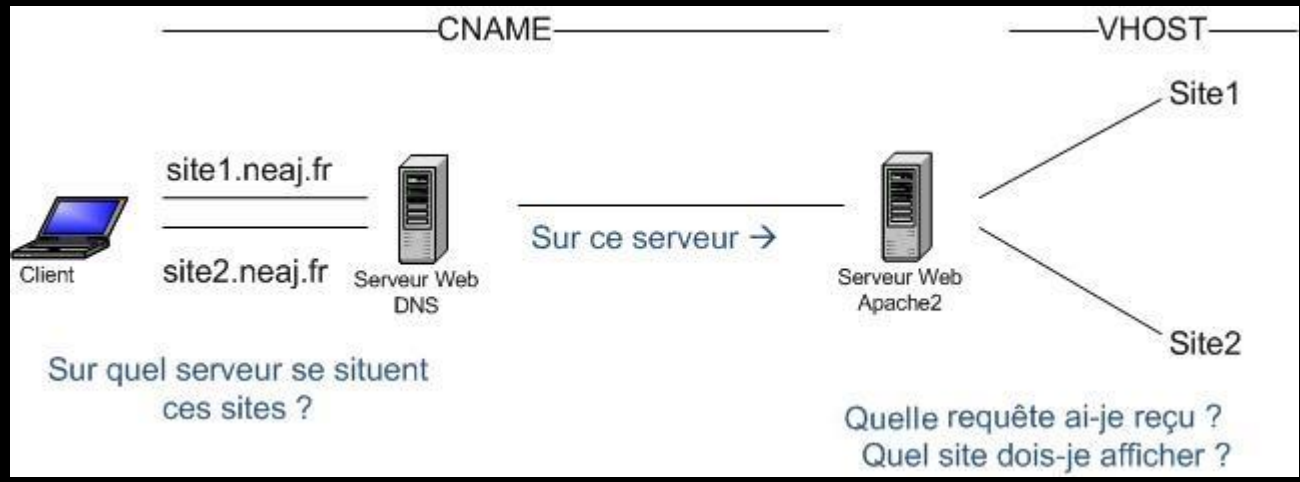
LAMP est un acronyme pour **L**inux, **A**pache, **M**ySQL, **P**HP



Installation d’Apache

apt install apache2

systemctl status apache2
systemctl start apache2
systemctl enable apache2



Installation de MariaDB

apt install mariadb-server
apt install mariadb-client

systemctl status mariadb-*
systemctl start mariadb-*
systemctl enable mariadb-*

```
CREATE DATABASE example;

CREATE USER 'userExample'@'localhost'
IDENTIFIED BY 'mot_de_passe';

GRANT ALL PRIVILEGES ON example.* TO
'userExample'@'localhost';

FLUSH PRIVILEGES;

QUIT;
```

Installation de PHP

apt install php php-mysql

D’autre modules existent pour PHP :

- php-curl
- php-gd
- php-intl
- php-json
- php-mbstring
- php-xml
- php-zip
- …

A vos VM !

Pratique

Sur votre machine Debian :

- Installer Apache MariaDB et PHP
- Ajouter une base de données sur MariaDB
- Vérifier le fonctionnement du serveur Apache
- Supprimer le fichier **/var/www/html/index.html** et créer un nouveau fichier **/var/www/html/index.php** et ajouter le contenu suivant :

```
<?php

echo 'Current PHP version: '. Phpversion();

?>
```

11. Le troubleshooting

UNAME

La commande **uname** permet d'indiquer le noyau utilisé :

```
root@sli-haproxy-1 ~:# uname  -a
Linux sli-haproxy-1 4.19.0-12-amd64 #1 SMP Debian 4.19.152-1 (2020-10-18) x86_64 GNU/Linux
```

DMESG

La commande **dmesg** permet d'afficher le contenu du journal de démarrage :

```
root@sli-haproxy-1 ~:# dmesg | grep error
[    7.658735] EXT4-fs (dm-0): re-mounted. Opts: errors=remount-ro
```

LSPCI

La commande **lspci** permet de répertorier tous les appareils détectés sur le bus PCI :

```
root@sli-haproxy-1 ~:# lspci
00:00.0 Host bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX Host bridge (rev 01)
00:01.0 PCI bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX AGP bridge (rev 01)
00:07.0 ISA bridge: Intel Corporation 82371AB/EB/MB PIIX4 ISA (rev 08)
00:07.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
00:07.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)
00:07.7 System peripheral: VMware Virtual Machine Communication Interface (rev 10)
00:0f.0 VGA compatible controller: VMware SVGA II Adapter
00:11.0 PCI bridge: VMware PCI bridge (rev 02)
00:15.0 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.1 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.2 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.3 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.4 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.5 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.6 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.7 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.0 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.1 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.2 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.3 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.4 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.5 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.6 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.7 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.0 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.1 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.2 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.3 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.4 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.5 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.6 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.7 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.0 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.1 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.2 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.3 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.4 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.5 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.6 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.7 PCI bridge: VMware PCI Express Root Port (rev 01)
03:00.0 Serial Attached SCSI controller: VMware PVSCSI SCSI Controller (rev 02)
0b:00.0 Ethernet controller: VMware VMXNET3 Ethernet Controller (rev 01)
```

LSOF

La commande **lsof** permet de lister tous les fichiers qui sont actuellement ouvert sur le système :

```
COMMAND     PID TID TASKCMD               USER   FD      TYPE             DEVICE SIZE/OFF       NODE NAME
systemd       1                           root  cwd       DIR              254,0     4096          2 /
systemd       1                           root  rtd       DIR              254,0     4096          2 /
systemd       1                           root  txt       REG              254,0  1489208     395982 /usr/lib/systemd/systemd
systemd       1                           root  mem       REG              254,0  1579448     392320 /usr/lib/x86_64-linux-gnu/libm-2.28.so
systemd       1                           root  mem       REG              254,0   149704     395023 /usr/lib/x86_64-linux-gnu/libudev.so.1.6.13
systemd       1                           root  mem       REG              254,0   137424     392639 /usr/lib/x86_64-linux-gnu/libgpg-error.so.0.26.1
systemd       1                           root  mem       REG              254,0    51136     393582 /usr/lib/x86_64-linux-gnu/libjson-c.so.3.0.1
systemd       1                           root  mem       REG              254,0    34904     395726 /usr/lib/x86_64-linux-gnu/libargon2.so.1
systemd       1                           root  mem       REG              254,0   432664     395740 /usr/lib/x86_64-linux-gnu/libdevmapper.so.1.02.1
systemd       1                           root  mem       REG              254,0    30776     395059 /usr/lib/x86_64-linux-gnu/libuuid.so.1.3.0
systemd       1                           root  mem       REG              254,0    26544     392051 /usr/lib/x86_64-linux-gnu/libattr.so.1.1.2448
systemd       1                           root  mem       REG              254,0  3031904     395768 /usr/lib/x86_64-linux-gnu/libcrypto.so.1.1
systemd       1                           root  mem       REG              254,0   593696     395770 /usr/lib/x86_64-linux-gnu/libssl.so.1.1
systemd       1                           root  mem       REG              254,0    26976     392077 /usr/lib/x86_64-linux-gnu/libcap-ng.so.0.0.0
```

IP ADDR

La commande **ip ad** permet d'afficher la configuration réseau de la machine :

```
root@sli-haproxy-1 ~:# ip ad
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
       valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:50:56:ae:32:db brd ff:ff:ff:ff:ff:ff
    inet 10.0.40.108/24 brd 10.0.40.255 scope global ens192
       valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:feae:32db/64 scope link
       valid_lft forever preferred_lft forever
```

Fichiers /var/log

Le dossier /var/log regroupe généralement les logs applicatifs et systèmes. Il ne faut pas hésiter aller chercher dedans les informations nécessaires.

```
root@sli-haproxy-1 ~:# ll /var/log/
total 2400
drwxr-xr-x  6 root root   4096 avril  5 00:00 .
drwxr-xr-x 11 root root   4096 nov.  23  2020 ..
-rw-r--r--  1 root root      0 avril  1 00:00 alternatives.log
-rw-r--r--  1 root root    482 mars  28 14:41 alternatives.log.1
-rw-r--r--  1 root root    214 mars  21 10:37 alternatives.log.2.gz
-rw-r--r--  1 root root    154 avril 19  2021 alternatives.log.3.gz
-rw-r--r--  1 root root   2309 nov.  23  2020 alternatives.log.4.gz
drwxr-xr-x  2 root root   4096 avril  1 00:00 apt
-rw-r-----  1 root adm  206678 avril  5 17:25 auth.log
-rw-r-----  1 root adm  531964 avril  3 00:00 auth.log.1
-rw-r-----  1 root adm   23075 mars  27 00:00 auth.log.2.gz
-rw-r-----  1 root adm   19587 mars  20 00:00 auth.log.3.gz
-rw-r-----  1 root adm   19354 mars  13 00:00 auth.log.4.gz
-rw-rw----  1 root utmp      0 avril  1 00:00 btmp
-rw-rw----  1 root utmp      0 mars   1 00:00 btmp.1
-rw-r-----  1 root adm   52494 avril  5 17:18 daemon.log
-rw-r-----  1 root adm  133308 avril  3 00:00 daemon.log.1
-rw-r-----  1 root adm    5071 mars  27 00:00 daemon.log.2.gz
-rw-r-----  1 root adm     860 mars  20 00:00 daemon.log.3.gz
-rw-r-----  1 root adm    2189 mars  13 00:00 daemon.log.4.gz
-rw-r-----  1 root adm       0 mars  22 00:00 debug
-rw-r-----  1 root adm    3780 mars  21 10:37 debug.1
-rw-r-----  1 root adm    4548 mars  12 15:18 debug.2.gz
```

Les commandes infos

- cat /proc/cpuinfo
- cat /proc/meminfo
- cat /proc/acpi/thermal_zone/THRM/temperature
- top ou htop

```
top - 18:05:54 up 24 days,  1:47,  1 user,  load average: 0,00, 0,00, 0,00
Tasks: 103 total,   1 running, 102 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0,0 us,  6,2 sy,  0,0 ni, 93,8 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
MiB Mem :   1994,3 total,   1333,8 free,     89,2 used,    571,3 buff/cache
MiB Swap:   2048,0 total,   2048,0 free,      0,0 used.   1708,3 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    1 root      20   0  105104  10456   7872 S   0,0   0,5   0:22.20 systemd
    2 root      20   0       0      0      0 S   0,0   0,0   0:00.33 kthreadd
    3 root       0 -20       0      0      0 I   0,0   0,0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0,0   0,0   0:00.00 rcu_par_gp
    6 root       0 -20       0      0      0 I   0,0   0,0   0:00.00 kworker/0:0H-kblockd
    8 root       0 -20       0      0      0 I   0,0   0,0   0:00.00 mm_percpu_wq
    9 root      20   0       0      0      0 S   0,0   0,0   0:03.90 ksoftirqd/0
   10 root      20   0       0      0      0 I   0,0   0,0   1:58.41 rcu_sched
```