

## Introduction:

La licence GPL de GNU

Le Code Source de **LINUX** est accessible gratuitement.

Qui protège LINUX ?

– la fondation FSF (Free Software Foundation) a développé un projet GNU pour permettre la distribution de programmes exempts de droits.

• Comment protéger LINUX ?

=> les utilisateurs GNU sont soumis à une licence d'utilisation : GPL (General Public Licence) qui spécifie les dispositions légales vis-à-vis de l'utilisation, la distribution et les modifications.

– ex : toute modification doit être rendue publique.

Linux est composé de

- le noyau (Kernel) : partie centrale d'Unix
- un serveur X : interface graphique permettant de créer des applications graphiques.
- un environnement graphique, comme : KDE, Gnome,...

Le **Noyau** permet de faire le lien entre les programmes et le matériel, en mettant à disposition des procédures appelées primitives ex: demande de plus de mémoire vive. Il partage les ressources rendant le système multitâche.

Il se charge des tâches essentielles au bon fonctionnement du système:

– gestion du système de fichier

– du processeur

– de la mémoire centrale

– des échanges entre l'unité centrale et les périphériques (entrées, sorties)

le **Shell** (console, terminal), de point de vue visuel est un écran noir avec en début de ligne un prompt. On peut taper à la suite des lignes de commandes.

Mais de point de vue informatique représente la liaison la plus élémentaire entre l'utilisateur et le système d'exploitation. Il permet d'interpréter les

commandes que l'on entre et d'exécuter des applications.

le prompt contient le nom de l'utilisateur courant suivi du nom de la machine et du nom du répertoire courant.

ex: info@debian:~\$

**syntaxe générale des commandes Unix:** info@debian:~\$ commande -options arguments

Les arguments indiquent sur quels objets la commande va agir et les options des variantes dans l'exécution de la commande.

– Les options sont précédées d'un tiret - L'ordre des options est indifférent.

– Si pas d'arguments, argument par défaut est pris en considération

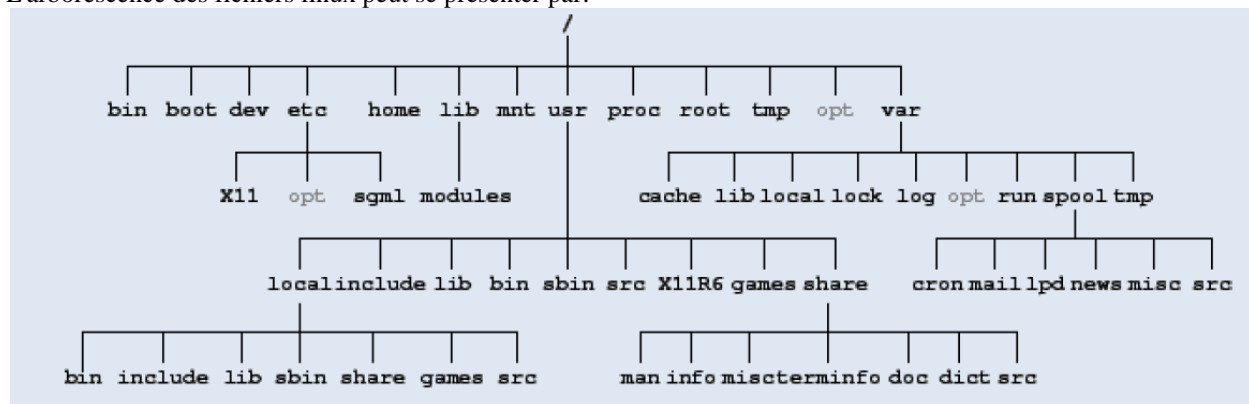
info@debian:~\$ ls -l                      info@debian:~\$ ls -la                      info@debian:~\$ ls debian.dpf

info@debian:~\$ ls -i debian.dpf

Pour connaître l'utilisation d'une commande : commande help ou utiliser le manuel : man commande

## Commandes de base, navigation dans les fichiers

L'arborescence des fichiers linux peut se présenter par:



(on trouve les fichiers ordinaires, les dossiers, les fichiers spéciaux ou device comme usb, cd-rom; tous sont fichiers dans linux)

Pour assurer la compatibilité et la portabilité, les systèmes Linux respectent l'unique norme FHS (**File Hierarchy Standard**). La hiérarchie de base est la suivante:

/	la racine, elle contient les répertoires principaux
/bin	contient des exécutables essentiels au système,
/boot	contient les fichiers permettant à Linux de démarrer
/dev	contient les points d'entrée des périphériques
/etc	contient les commandes et les fichiers nécessaires à l'administrateur du système
/etc/X11	contient les fichiers spécifiques à la configuration de X
/etc/opt	contient les fichiers de configuration des applications installés dans /opt
/home	répertoire personnel des utilisateurs
/lib	contient des bibliothèques partagées essentielles au système lors du démarrage
/media	contient les points de montage des partitions (Ubuntu)
/mnt	contient les points de montage des partitions temporaires (cd-rom, disquette, ...)
/opt	contient des packages d'applications supplémentaires
/root	répertoire de l'administrateur root
/sbin	contient les binaires système essentiels
/tmp	contient les fichiers temporaires
/usr	Hiérarchie secondaire
/usr/X11R6	ce répertoire est réservé au système X version 11 release 6
/usr/X386	utilisé avant par X version 5, c'est un lien symbolique vers /usr/X11R6
/usr/bin	contient la majorité des fichiers binaires et commandes utilisateurs
/usr/include	contient les fichiers d'en-tête pour les programmes C et C++
/usr/lib	contient la plupart des bibliothèques partagées du système
/usr/local	contient les données relatives aux programmes installés par le root
/usr/local/bin	binaires des programmes locaux
/usr/local/games	binaires des jeux locaux
/usr/local/include	fichiers d'en-tête C et C++ locaux
/usr/local/lib	Bibliothèques partagées locales
/usr/local/sbin	binaires système locaux
/usr/local/share	hiérarchie indépendante
/usr/local/src	fichiers sources locaux
/usr/sbin	contient les fichiers binaires non essentiels au système réservés à l'administrateur
/usr/share	réserve aux données non dépendantes de l'architecture
/usr/src	contient des fichiers de code source
/var	contient des données variables

### 1) ls

Liste les fichiers d'un dossier avec les Options :

- a pour les fichiers cachés,
- l pour la liste détaillée,
- h pour les tailles en unités,
- R permet de visualiser les sous-dossiers

### 2) cd

change **directory**, la commande permet de naviguer dans l'arborescence. Par exemple `cd /home/info` va dans le dossier de l'utilisateur info, quel que soit l'endroit où l'on se trouve puisqu'on a mis le chemin absolu. En revanche, `cd sampleC/code` va dans le répertoire code de sampleC. Comme on ne met pas de slash (/) de début, il s'agit d'une adresse relative, on ajoute donc ce chemin à celui dans lequel on se trouve déjà.

Deux raccourcis très utiles sont à connaître. `cd ~` mène au répertoire de l'utilisateur courant (`/home/info/` la plupart du temps ou `/root/` si vous êtes en root) et `cd ..` retourne sur le dossier père

### 3) du

**disk usage**, précise l'espace disque que prend chaque fichier ou dossier (l'option `-h` permet d'obtenir les tailles, tandis que l'option `--max-depth=x` (ou `-d`), permet de limiter le détail à un niveau de sous-dossier. `--max-depth=1` retournera que la taille des répertoires courants.

### 4) pwd

**print working directory**, cette commande affiche tout simplement le chemin absolu du dossier courant

### 5)ssh

secure shell. Permet de se connecter au shell d'un ordinateur distant et d'y exécuter des commandes.

ex: info@debian:~\$ ssh login@ip ou nom\_hôte

Une option de ssh est: tunneling qui permet par exemple de déjouer les pare-feux par la mise en place d'un proxy socks. Pour se diriger vers un port local (port 2013) à travers la connexion SSH:

ex: info@debian:~\$ ssh -D 2013 login@ip\_serveur\_distant

### 6)clear

nettoie l'écran de terminal. Le raccourci clavier ctrl + l fait la même chose

### 7)stty

set teletype affiche ou change les caractéristiques d'un terminal avec l'option -a la combinaison des touches d'interruption d'une commande. ex:Ctrl+c arrêt de la commande en cours; ctrl + d déconnecte proprement une session ou un terminal; ctrl + r permet de faire une recherche dans l'historique des commandes,...

Pour travailler en mode plein terminal les touches Ctrl+Alt+Fn(n:1,...,6) pour revenir en mode graphique Ctrl+Alt+F7 ou Ctrl+Alt+F8

### 8)!!

Lance à nouveau la dernière commande

### 9) date

affiche l'heure et la date du système

### 10)who

indique les utilisateurs connectés au système

### 11) whoami

affiche le nom d'utilisateur courant

### 12) cal

donne l'affichage d'un calendrier

### 13) man

affichage de la documentation , suivi d'une commande affiche la documentation sur cette commande

### 14) passwd

définir un mot de passe

### 15) exit

terminer une session

pour utiliser plusieurs commandes sur la même ligne, il suffit de les séparer par ;

ex: : info@debian:~\$ date;who;cal

### 16)wc

donne le nombre de lignes, les mots et le nombre de bytes

### 17)wich

renvoie le chemin d'un fichier

## **Manipulation**

1) définir un nouveau mot de passe

2)Essayer deux possibilités pour quitter une session

3)Trouver avec man d'autres options de la commande who

4)Donner le nom du jour de votre date de naissance

5) Est il possible la date système

6) Quels sont les répertoires placés directement dans la racine

7)Essayer de remonter à partir de la racine

8)En quoi le répertoire /dev se distingue

9) Afficher les fichiers cachés en couleur

10)Lister le contenu de répertoire /etc

11)Quels sont les utilisateurs en cours du système?

12)Fermer ce terminal.

13) afficher uniquement le nombre de bytes puis le nombre des mots et puis le nombre de lignes d'un fichier

14) Que renvoie la commande: wich ls

**18)cat**

permet de lire le contenu d'un fichier texte (en code ASCII) ex: info@debian:~\$ cat monfichier

**19)less**

permet de lire le contenu d'un fichier texte (en code ASCII) page par page

**20)head**

affiche les 10 premières lignes d'un fichier texte, l'option -n permet de spécifier le nombre de lignes à afficher

**21)tail**

semblable à head, mais concernant la fin du fichier; avec l'option -f(follow), permet de mettre à jour en temps réel l'affichage de la fin du fichier (par exemple suivre l'évolution d'un fichier de logs)

**22) mkdir**

crée un nouveau dossier; ex: info@debian:~\$ mkdir dossier1

**23)cp**

permet de copier un fichier dans un emplacement ; ex: info@debian:~\$ cp fichier1 dossier1/fichier-→ donne la copie de fichier1 qui est fichier2 dans le répertoire dossier1

**24) mv**

Permet de déplacer des fichiers/dossiers dans un autre emplacement . Si les deux fichiers font référence au même répertoire, le fichier source sera renommé.

**25)rm**

supprime des fichiers; ex: info@debian:~\$ rm fichier1. L'option -f force la suppression, l'option -i demande une confirmation avant suppression, enfin Attention!!!! avec l'option -r permet la suppression récursive des dossiers

**26)ln**

link, crée un lien entre deux fichiers. L'option -s permet de créer un lien symbolique

**27) sort**

trie un fichier texte par ordre alphabétique. L'option -r permet d'effectuer un tri inverse, l'option -R permet un tri aléatoire, l'option -u permet d'éliminer les doublons et l'option -o permet de créer un nouveau fichier avec les résultats triés.

**28) uniq**

permet de dédoublonner un fichier; ex : info@debian:~\$ uniq doublons.txt no-doublons.txt

**29)cut**

coupe dans un fichier texte. Pour couper toutes les lignes selon un nombre donné de caractères, on utilisera l'option cut -c 2 conservera seulement les deux premiers caractères. On peut aussi donner un intervalle : cut -c 2-4, alors on conservera uniquement les caractères deux à quatre.

**30) tar**

tar est l'utilitaire d'archivage. Il permet de regrouper des fichiers et des dossiers dans une seule archive. tar -cvf (create, verbose, file) permet de créer une archive, d'afficher tous les détails du processus (mode verbeux) et de tout mettre dans un dossier.  
exp:info@debian:~\$ tar -cvf new\_archive.tar dossier\_archiver

**31)gzip**

permet de comprimer les fichiers au format zip, pour les extraire utiliser la commande gunzip

**32) iconv**

permet de changer l'encodage d'un fichier. option -f pour préciser l'encodage d'origine et option -t pour celui de destination

**33) wget**

copie un fichier distant sur l'ordinateur. ex: info@debian:~\$ wget http://www.fst.ma/info/01/fichier.txt

**34) scp**

visé à remplir la même fonction que la commande de copie cp, mais elle permet de copier les fichiers de manière sécurisée à travers le réseau; c'est-à-dire entre hôtes distants. De même qu'avec cp, l'option -r permet de copier un répertoire entier. scp fichier\_exemple login@ip\_ou\_adresse:adresse\_de\_destination  
ex: info@debian:~\$ scp test.txt fstt@192.168.0.128:~/transfert  
ou pour récupérer un fichier d'un hôte distant;ex: info@debian:~\$ scp fstt@192.168.0.128:~/movie.avi ~/

### 35)file

détermine le type d'un fichier indépendamment de son extension. Il suffit de lui fournir en paramètre le fichier à évaluer

### 36)split

coupe un fichier en fichiers plus petits (-l préciser un nombre de lignes, -b préciser une taille en bytes (suivre la taille de K, M, G, T pour définir une unité différente). Pour créer des fichiers de 300 lignes  
ex: info@debian:~\$ split -l 300 fichier1,  
ou des fichiers de 1 MB ex: info@debian:~\$ split -b 1000000 fichier1.

### 37)sed

sed est un éditeur non interactif.

Cette commande permet d'appliquer un certain nombre de commandes sur un fichier puis d'en afficher le résultat (sans modification du fichier de départ) sur la sortie standard.

Syntaxe

sed [-n] [-e commande] [-f fichier de commandes] [fichier]

-n: écrit seulement les lignes spécifiées (par l'option /p) sur la sortie standard

-e: permet de spécifier les commandes à appliquer sur le fichier. Cette option est utile lorsque vous appliquez plusieurs commandes. Afin d'éviter que le shell interprète certains caractères, il faut mieux encadrer la commande avec des ' ou des " .

-f: les commandes sont lu à partir d'un fichier.

Principe de fonctionnement

Pour chaque ligne , on applique la commande (si cela est possible) puis on affiche sur la sortie standard la ligne modifiée ou non.

la syntaxe générale des commandes est de la forme caractérisation\_des\_adresses commandes avec caractérisation\_des\_adresses de la forme :

toutes les lignes

num: la ligne num (la dernière ligne est référencée par \$)

num1,num2: les lignes entre les lignes num1 et num2

RE: les lignes correspondant à l'expression régulière RE

RE1,RE2: les lignes entre la première ligne correspondant à l'expression régulière RE1 et la première ligne correspondant à l'expression régulière RE2

Les commandes sed

La commande de substitution s

syntaxe :

ad1,ad2s/RE/remplacement/flags

Remplace les expressions régulières RE par la chaîne de remplacement entre les lignes ad1 à ad2  
flags

g: global, c'est à dire toutes les occurrences de la chaîne RE (par défaut seule la première occurrence est remplacée)

p: imprime la ligne (utile avec l'option -n)

w fichier: écrit la ligne dans le fichier spécifié en plus de la sortie standard.

exemple :

info@debian:~\$ sed "s/[Cc]omputer/COMPUTER/g" fichier

info@debian:~\$ sed -e "s/\([0-9]\)[0-9]\*\)/\*\*\1\*\*/"

fichier : encadre le premier nombre de la ligne avec des \*\*

La négation !

syntaxe :

ad1,ad2 !fonction argument

La fonction est appliquée à toutes les lignes qui ne correspondent pas à la caractérisation.

La commande de suppression d

efface les lignes (au niveau de la sortie, le fichier d'origine n'est pas modifié)

exemple:

info@debian:~\$ sed "1,10d" fichier : sortie du fichier à partir de la onzième ligne

info@debian:~\$ sed "/^From/d" fichier : On n'efface tout sauf les lignes commençant par From , donc on imprime les lignes commençant par From.

Remarque :

Il est parfois plus facile de caractériser la négation de ce que l'on veut (voir exemple précédent) . Par exemple plutôt de récupérer ce que je veux, j'efface ce qui ne m'intéresse pas <p>

Les commande d'insertions a,i

a\; texte : écrit le texte après la ligne

i\; texte : écrit le texte avant la ligne

exemple :

fichier de commandes:

1i\

\

-----\  
LOGIN USER\  
-----

s:/!/

s:/-/

s:/-/

s:/!/

s/!.\*/ /

s/:.\*/

commande:

info@debian:~\$ sed -f fich\_commandes /etc/passwd

resultat :

-----  
LOGIN

USER  
-----

root

Operator

info

genieInfo

S5

Section 5

Les autres commandes: q, = et w

q: quitte

=: écrit les numéro de ligne

w fichier:écrit dans un fichier

exemple :

fichier d'entrée :

Un,

deux.

Trois,

quatre.

commande :

```
info@debian:~$ sed -e "q" fichier
```

résultat à l'écran :

Un,

commande :

```
info@debian:~$ sed -e "\./=/" -e "[A-Z]/w capitale" fichier
```

résultat à l'écran :

Un,

2

deux.

Trois,

4

quatre.

résultat dans le fichier capitale :

Un,

Trois,

### 38)awk

Cette commande permet d'appliquer un certain nombre d'actions sur un fichier. La syntaxe est inspirée du C syntaxe

```
awk [-Fs] [-v variable] [-f fichier de commandes] 'program' fichier
```

-F        Spécifie les séparateurs de champs

-v        Définit une variable utilisée à l'intérieur du programme.

-f        Les commandes sont lu à partir d'un fichier.

principe de fonctionnement

Le programme awk est une suite d'action de la forme : motif { action } , le motif permet de déterminer sur quels enregistrements est appliquée l'action.

Un enregistrement est :

une chaîne de caractères séparée par un retour chariot, en général une ligne.

Un champ est :

une chaîne de caractères séparées par un espace (ou par le caractère spécifié par l'option -F), en générale un mot. On accède à chaque champs de l'enregistrement courant par la variable \$1, \$2, ... \$NF. \$0 correspond à l'enregistrement complet. La variable NF contient le nombre de champs de l'enregistrement courant, la variable \$NF correspond donc au dernier champs.

Exemples

```
info@debian:~$ awk -F ":" '{ $2 = "" ; print $0 }' /etc/passwd  imprime chaque ligne du fichier /etc/passwd après avoir effacé le deuxième champs
```

```
info@debian:~$ awk 'END {print NR}' fichier            imprime le nombre total de lignes du fichiers
```

```
info@debian:~$ awk '{print $NF}' fichier    imprime le dernier champs de chaque ligne
```

```
info@debian:~$ who | awk '{print $1,$5}'    imprime le login et le temps de connexion.
```

```
info@debian:~$ awk 'length($0)>75 {print}' fichier  imprime les lignes de plus de 75 caractères. (print équivaut à print $0)
```

```
info@debian:~$ cat >> myfile    (contenant: Ceci est un test. Ceci un deuxième test. Ceci un troisième test. Ceci un quatrième test
```

```
info@debian:~$ awk '{print $1}' myfile ;
```

```
info@debian:~$ awk -F: '{print $1}' /etc/passwd le séparateur n'est un espace mais : (spécifié par -F":")
```

```
info@debian:~$ awk '{ print $1 " moyenne " ($2 +$3)/2 }' myfile
```

le fichier myfile

B1	18 5
B2	6 11
B3	8 4
B4	16 8
B5	16 18

### 39)locate

cette commande permet de localiser un fichier sur le disque dur. ex: info@debian:~\$ locate monfichier.txt.  
La commande locate est rapide, car elle retrouve le fichier en consultant une base de données. Elle ne parcourt pas directement le disque dur à la recherche du fichier en question. L'inconvénient de ce procédé est que si le fichier est tout récent, il risque de ne pas encore être indexé, et locate ne peut le trouver. On peut forcer la mise à jour de la base de données avec la commande sudo updatedb.  
On peut aussi utiliser la commande find

### 40)find

**Les méta caractères** sont des caractères génériques permettant de désigner un ensemble d'objets et s'appliquent aux arguments des commandes qui désignent des noms de fichiers.

\*: toutes chaînes de caractères, y compris la chaîne vide

ex : a\*b tous les noms de fichiers commençant par a et finissant par b

?: caractère quelconque

ex : a?b tous les noms de fichier commençant par a, suivi d'un caractère et finissant par b

[. . .]: un caractère quelconque ∈ à la liste donnée entre crochets. Le - permet de représenter un intervalle.

ex : a[a-z0-9A-Z]b désigne tous les noms de fichiers commençant par a suivi d'un caractère alphanumérique et finissant par b

[! . . .] une liste de caractères à exclure

ex : a[!a-z]b tous les noms de fichiers commençant par a suivi d'un caractère autre qu'un caractère alphabétique en minuscule et finissant par b

**Les expressions régulières** sont aussi des suites de caractères permettant de faire des sélections.

Les différentes expressions régulières sont :

^ début de ligne

. un caractère quelconque

\$ fin de ligne

x\* zéro ou plus d'occurrences du caractère x

x+ une ou plus occurrences du caractère x

x? une occurrence unique du caractère x

[...] plage de caractères permis

[^...] plage de caractères interdits

\ {n\} pour définir le nombre de répétition n du caractère placé devant

Exemple l'expression [a-z][a-z] \* cherche les lignes contenant au minimum un caractère en minuscule. [a-z] caractère permis, [a-z]\* recherche d'occurrence des lettres permises.

L'expression ^[0-9]\ {4\}\$ a pour signification, du début à la fin du fichier \$, recherche les nombres[0-9] de 4 chiffres \ {4\}.

**La commande find** permet de retrouver des fichiers à partir de critères de recherche. La syntaxe est:

find <répertoire de recherche> <critères de recherche>

Les critères de recherche sont les suivants :

-name recherche sur le nom du fichier,

-perm recherche sur les droits d'accès du fichier,

-links recherche sur le nombre de liens du fichier,

-user recherche sur le propriétaire du fichier,



- group recherche sur le groupe auquel appartient le fichier,
- type recherche sur le type (d=répertoire, c=caractère, f=fichier normal),
- size recherche sur la taille du fichier en nombre de blocs (1 bloc=512octets),
- atime recherche par date de dernier accès en lecture du fichier,
- mtime recherche par date de dernière modification du fichier,
- ctime recherche par date de création du fichier.

On peut combiner les critères avec des opérateurs logiques :  
critère1 critère2 ou critère1 -a critère2 correspond au et logique,  
!critère non logique,  
(critère1 -o critère2) ou logique,

La commande find doit être utilisé avec l'option -print. Sans l'utilisation de cette option, même en cas de réussite dans la recherche, find n'affiche rien à la sortie standard (l'écran, plus précisément le shell).

La commande find est réursive, c'est à dire où que vous tapiez, il va aller scruter dans les répertoires, et les sous répertoires qu'il contient, et ainsi de suite.

Pour chercher un fichier dont le nom contient la chaîne de caractères file1 à partir du répertoire /usr:

```
info@debian:~$ find /usr -name file1 -print
```

En cas de réussite, si le(s) fichier(s) existe(nt), vous aurez comme sortie :

```
file1
```

En cas d'échec, vous n'avez rien.

Pour rechercher tous les fichiers se terminant par .c dans le répertoire /usr :

```
info@debian:~$ find /usr -name " *.c " -print
```

on obtient toute la liste des fichiers se terminant par .c sous les répertoires contenus dans /usr (et dans /usr lui même).

Pour connaître les derniers fichiers modifiés dans les 3 derniers jours dans toute l'arborescence (/):

```
info@debian:~$ find / -mtime 3 -print
```

Pour connaître dans toute l'arborescence, les fichiers dont la taille dépasse 1Mo (2000 blocs de 512Ko):

```
info@debian:~$ find / -size 2000 -print
```

Vous pouvez chercher dans toute l'arborescence, les fichiers ordinaires appartenant à olivier, dont la permission est fixée à 755, on obtient :

```
info@debian:~$ find / -type f -user info -perm 755 -print
```

La commande find peut générer beaucoup de messages d'erreur (du genre permission denied. Pour éviter ceci, on peut rediriger les messages d'erreur dans un fichier poubelle (comme /dev/null), les messages d'erreur sont alors perdus (rien n'empêche de les sauvegarder dans un fichier, mais ça n'a aucune utilité avec la commande find).

```
info@debian:~$ find . -name bobo -print
```

Si on veut connaître les fichiers n'appartenant pas à l'utilisateur info:

```
info@debian:~$ find . ! -user info -print
```

! -user info, est la négation de -user info, c'est à dire c'est tous les utilisateurs sauf info.

Recherche des fichiers qui ont pour nom a.out et des fichiers se terminant par .c. On tape :

```
info@debian:~$ find . \ ( -name a.out -o -name " *.c " \ ) -print
```

On recherche donc les fichiers dont le nom est a.out ou les fichiers se terminant par \*.c, une condition ou l'autre.

Recherche des fichiers qui obéissent à la fois à la condition a pour nom core et à la condition a une taille supérieure à 1Mo.

```
info@debian:~$ find . \ ( -name core -a size +2000 \ ) -print
```

L'option -print est une commande que l'on passe à find pour afficher les résultats à la sortie standard. En dehors de print, on dispose de l'option -exec. find couplé avec exec permet d'exécuter une commande sur

les fichiers trouvés d'après les critères de recherche fixés. Cette option attend comme argument une commande, celle ci doit être suivi de {} \ ;.

Exemple recherche des fichiers ayant pour nom core, suivi de l'effacement de ces fichiers.

```
info@debian:~$ find . -name core -exec rm {} \ ;
```

Tous les fichiers ayant pour nom core seront détruits, pour avoir une demande de confirmation avant l'exécution de rm:

```
info@debian:~$ find . -name core -ok rm {} \ ;
```

Une fonction intéressante de find est de pouvoir être utilisé avec d'autres commandes UNIX. Par exemple:

```
info@debian:~$ find . -type f -print | xargs grep file1
```

En tapant cette commande vous allez rechercher dans le répertoire courant tous les fichiers normaux (sans les répertoires, fichiers spéciaux), et rechercher dans ces fichiers tous ceux contenant la chaîne file1.

#### 41)grep

La commande grep permet de rechercher une chaîne de caractères dans un fichier. Les options sont les suivantes :

-v affiche les lignes ne contenant pas la chaîne

-c compte le nombre de lignes contenant la chaîne

-n chaque ligne contenant la chaîne est numérotée

-x ligne correspondant exactement à la chaîne

-l affiche le nom des fichiers qui contiennent la chaîne

Exemple avec le fichier carnet-adresse : info@debian:~\$ cat carnet-adresse

```
ahmed:29:0298333242:Rabat
```

```
mohamed:13:0466342233:Tanger
```

```
sara:30:0434214452:Nador
```

```
nada:92:013344433:Tantan
```

On peut utiliser les expressions régulières avec grep. Si on tape la commande :

```
info@debian:~$ grep ^[a-d] carnet-adresse
```

On va obtenir tous les lignes commençant par les caractères compris entre a et d. Dans notre exemple, on n'en a pas, d'où l'absence de sortie.

```
info@debian:~$ grep Tanger carnet-adresse
```

Permet d'obtenir les lignes contenant la chaîne de caractère Tanger, soit :

```
mohamed:13:0466342233:Tanger
```

Il existe aussi les commandes fgrep et egrep équivalentes

#### Manipulation

1) donner l'évolution d'un fichier logs

2) Créer un nouveau fichier myfile

3)utiliser la commande touch avec un fichier qui existe déjà ?

4) utiliser cat sans nom de fichier ?

5) Que fait l'option -R dans la commande cp

6) afficher uniquement les 100 bytes du fichier fichier1

7)Quelle est la distribution installée?

8)utiliser man pour connaître la commande od

9) Lister le contenu du fichier /etc/passwd et /etc/shadow

```
[username]:[x]:[UID]:[GID]:[Comment]:[Home directory]:[Default shell]
```

le fichier /etc/group:

```
[Group name]:[Group password]:[GID]:[Group members]
```

Le fichier / etc / shadow stocke le mot de passe actuel au format crypté (plus similaire au hachage du mot de passe) du compte de l'utilisateur avec des propriétés supplémentaires associées au mot de passe de l'utilisateur.

Fondamentalement, il stocke des informations de compte d'utilisateur sécurisées. Tous les champs sont séparés par un signe deux-points (:). Il contient une entrée par ligne pour chaque utilisateur répertorié dans le fichier / etc / passwd. En général, l'entrée dans le fichier shadow ressemble à ce qui suit:

vivek:\$1\$infffc\$P\$GteyHdicpGOffXX4ow#5:13064:0:99999:7:::

- 10) Donner la différence entre la commande ln et cp
- 11) créer un fichier qui contient le nom des étudiants d'une classe, mettre le résultat d'un tri croissant dans un autre fichier
- 12) Déterminer le répertoire par défaut dans la hiérarchie des répertoires?
- 13) Y a-t-il des fichiers, des répertoires dans ce répertoire?
- 14) Entrer du texte dans un fichier nommé "Fichier1".
- 15) Lister le contenu de "Fichier1".
- 16) Lister le répertoire courant.
- 17) Lister les répertoires /bin et /dev.
- 18) Créer sous votre répertoire deux sous-répertoires : "Source" et "Data".
- 19) Se positionner sous "Source".
- 20) Lister le répertoire courant.
- 21) Revenir sous le répertoire de départ et détruire "Source".
- 22) Créer un deuxième fichier nommé "Fichier2".
- 23) Copier chaque fichier en nom\_de\_fichier.old.
- 24) Créer un répertoire "Old".
- 25) Déplacer les fichiers avec l'extension old vers le répertoire "Old".
- 26) Copiez les fichiers sans extension dans le répertoire "Data".
- 27) Sous votre répertoire de départ, créez un lien matériel "lien1" équivalent à "Fichier2".
- 28) Lister les deux fichiers "lien1" et "Fichier2" en affichant leur numéro d'inode. Que remarquez-vous?
- 29) Supprimer "lien1". "Fichier2" a-t-il disparu?
- 30) Sous votre répertoire de départ, créez un lien symbolique "lien2" sur "Fichier2".
- 31) Lister les deux fichiers "lien2" et "Fichier2". Que remarquez-vous?
- 32) Supprimer "Fichier2". "lien2" a-t-il disparu?
- 33) Quelle est la taille totale des fichiers contenus dans votre répertoire?
- 34) Renommer les fichiers créés
- 35) Effacer tous les fichiers créés.
- 36) utiliser la commande cut avec ses différentes options dans un fichier texte que vous venez de créer
- 37) Comment chercher tous les fichiers dont les noms commençant par un «a» majuscule ou une minuscule, suivi d'éventuellement quelques lettres ou chiffres, et se terminent par un chiffre entre 3 et 6 ?  
find -name '[aA]\*[3-6]' -print
- 38) Comment fait-on pour indiquer que le fichier recherché a été modifié il y a plus de 30 jours ? Il y a 30 jours ? Il y a moins de 30 jours ? avec l'option -mtime
- 39) chercher un fichier .tex modifié plus récemment que fichier1.tex , il y'a plus que 5 jours  
find . -newer bofichier.tex -mtime +5 -name '\*.tex'
- 40) Comment examiner plusieurs fichiers à la suite ? Comment passe-t-on au suivant ? Comment revient-on au précédent ?  
( less file1 file2 file3 Pour passer au fichier suivant, on tape :n (next); pour revenir au fichier précédent, on tape :p (previous). :x permet de revenir au premier fichier spécifique)
- 41) Comment ouvrir le fichier file1 en se plaçant à la ligne 13 ?  
(less +13 file1)
- 42) Comment ouvrir le fichier file2 en se plaçant dans la ligne où apparaît le mot "test"  
(less -p test file1 ou encore less +/test file1)
- 43) même que 42 avec le fait d'ignorer la casse des caractères  
(less -ip TEST file1  
less -ip test file1  
less -i +/test file1  
less -i +/TEST file1)
- 44) Comment chercher dans un fichier le caractère /  
(essayez // ou /\ /)
- 45) copier le répertoire Rapport et ses sous-répertoires dans le répertoire dossier1  
(cp -r Rapport/ dossier1/ ou cp -R Rapport/ dossier1/)

46) on veut faire une copie appelée tentative/ de son répertoire essai/, et que le répertoire tentative/ existe déjà ?  
(on obtient une copie de essai/, également appelée essai/, placée dans le répertoire tentative/)

47) Quel est l'effet de cette commande `cp fichier1 fichier2 fichier3 repertoire/` ?

Les fichiers fichier1, fichier2 et fichier3 sont copiés sous leur nom dans le répertoire, qui doit exister.

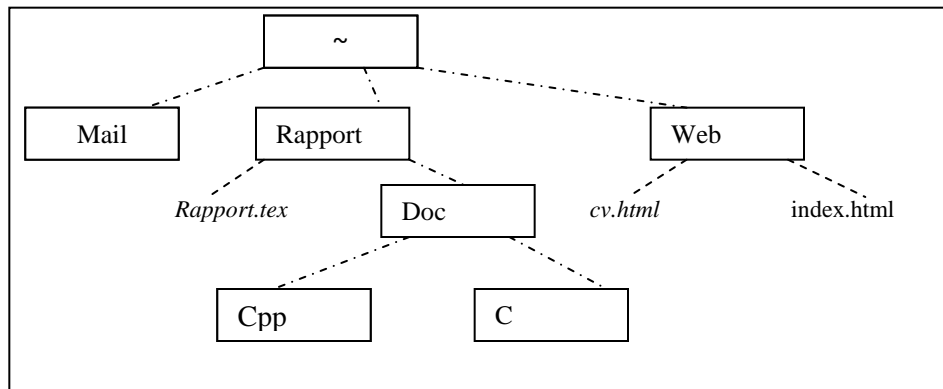
48) soient les commandes: `info@debian:~$ ls`

l'affichage à l'écran est: `rapport.tex notes etudiants`

`info@debian:~$ mv notes etudiant`

que se passe t-il après cette commande?( si on a un alias de mv sur mv -i alors le système remplace le contenu de fichier etudiant par celui du fichier de note)

49)soit l'arborescence suivante:



50) On est dans ~/Rapport/docs/ et on veut y déplacer le fichier Rapport.tex ? (`mv ../Rapport.tex .`)

51) Le faire revenir à son premier emplacement ? (`mv Rapport.tex ..`)

52) Comment peut-on déplacer le fichier Rapport.tex stage et changer son nom par "stage" ?

53) Expliquer ces commandes

`info@debian:~$ ls`

`file1 file2 Rapport/`

`info@debian:~$ alias mv`

`mv='mv -i'`

`info@debian:~$ mv file1 file2`

`mv: overwrite file2 (yes/no)? n`

`info@debian:~$ mv -f file1 file2`

`info@debian:~$ ls`

`file2 Rapport/`

54) Quelles sont les options de grep qui permettent d'obtenir des lignes de contexte (qui précèdent et/ou suivent la ligne où figure le mot) ?

(-num : le numéro indique le nombre de lignes de contexte que l'on veut voir figurer avant et après la ligne où figure le mot recherché. Par exemple, si on veut trois lignes de contexte, avant et après la mot (soit sept lignes au total), on tape :

`grep -3 ...`

-A num (after) : le numéro indique le nombre de lignes qui doivent suivre la ligne où figure le mot. Si on en veut quatre, on tapera :

`grep -A 4 ...`

-B num (before) : le numéro indique le nombre de lignes qui doivent précéder la ligne où figure le mot. Si on en veut dix, on tape :

`grep -B 10 ...`

-C num (context) : équivalent à -num. En fait, les trois lignes suivantes sont strictement équivalentes :

`grep -2 ...`

`grep -C 2 ...`

`grep -A 2 -B 2 )`

55)Comment faire apparaître le numéro de la ligne où figure le mot recherché ?

C'est avec l'option -n (number); le numéro figure tout au début de la ligne, suivi d'un deux-points (:) et du texte. Par exemple :

```
info@debian:~$ grep -n violon verlaine.tex
```

```
12:des violons de l'automne
```

Quand on fait une recherche dans plusieurs fichiers, le nom du fichier figure d'abord, puis le numéro de la ligne, et enfin le texte, le tout séparé par des deux-points. Par exemple :

```
info@debian:~$ grep -n violon *
```

```
verlaine.tex:12:des violons de l'automne
```

```
orchestre:45:Cordes : contrebasse, violoncelle, alto, violons.
```

56) Que se passe-t-il quand on demande également des lignes de contexte ?

La disposition générale ne change pas, par contre, le signe utilisé pour séparer la ligne de son numéro est un tiret (-) quand il s'agit des lignes de contexte, et un deux-points quand il s'agit de la ligne voulue. Par exemple :

```
info@debian:~$ grep -nC violon verlaine.tex
```

```
10-
```

```
11-Les sanglots longs
```

```
12:des violons de l'automne
```

```
13-bercent mon coeur
```

```
14-d'une langueur monotone
```

57) Comment faire pour afficher le nombre d'occurrences du mot recherché ?

On utilise l'option -c (count) :

```
info@debian:~$ grep -c violon *
```

```
verlaine.tex:1
```

```
orchestre:1
```

58) Comment faire pour que grep ignore la casse des caractères (différence entre majuscules et minuscules) dans sa recherche ?

Par défaut, grep fait la différence entre les majuscules et les minuscules; pour invalider ce comportement, on utilise l'option -i (ignorecase).

59) Comment faire pour faire apparaître non pas les lignes où figurent le mot, mais les noms des fichiers ?

C'est l'option -l qui permet de faire cela : afficher les noms des fichiers où figure au moins une fois la chaîne de caractères recherchée.

60) Comment faire apparaître les lignes où ne figurent pas le mot recherché ?

On veut en fait inverser le sens de la recherche : c'est l'option -v qui fait cela.

61) Comment faire apparaître les noms des fichiers ne contenant pas le mot recherché ?

On utilise l'option -L, qui affiche les noms de fichiers où ne figurent pas la chaîne de caractères recherchée. Il ne faut bien sûr pas confondre les options -l et -L...

62) Comment faire pour que grep ne recherche que les lignes où figure le mot tel quel, et non pas ses variantes ?

C'est l'option -w (comme word) qui sert à cela : un mot complet est délimité comme suit :

Début : la chaîne de caractères est placée au début d'une ligne, ou précédée d'un blanc, d'une tabulation ou d'une ponctuation.

Fin : la chaîne de caractère est placée en fin de ligne, ou suivie d'un blanc, d'une tabulation ou d'une ponctuation.

Si donc on veut chercher «travail» et aucune forme dérivée de ce mot, on écrit :

```
grep -w travail mon-fichier
```

63) Comment faire pour chercher plusieurs mots à la fois en faisant apparaître les numéros des lignes ?

On veut chercher toutes les occurrences des mots «terre» et «ciel» dans les deux premiers chapitres de la première partie de Germinal, avec les numéros des lignes. On propose deux solutions, la première utilisant les ressources de la syntaxe de grep, la seconde utilisant l'option -f avec un fichier.

Syntaxe de grep : La structure `\(mot1\|mot2\)` permet de chercher plusieurs mots. Ici, on tape la ligne suivante :

```
grep \"(ciel\\|terre\\)\" fichier
```

On met des apostrophes de part et d'autre de l'expression pour la protéger contre le shell, c'est-à-dire pour que le shell ne cherche pas à interpréter l'expression.

Option «-f fichier» : dans un fichier quelconque, que nous appellerons liste, on indique les mots que l'on recherche : «ciel» et «terre». Chaque ligne correspond à un mot recherché. Il ne faut donc pas mettre de ligne comme terre ciel

car le programme chercherait la chaîne de caractères «terre ciel», qui est assez improbable en français. Il ne faut pas non plus laisser de ligne blanche : le programme afficherait l'ensemble du texte.

64)Quelle que soit la solution retenue, on veut ensuite afficher le numéro des lignes (option -n); d'autre part, pour que la recherche soit exhaustive, il vaut mieux que grep ne fasse pas de différence entre les majuscules et les minuscules, avec l'option -i (ignore case, ignorer la casse des caractères). Il faut aussi décider si on cherche les mots tels quels, sans leurs variantes (comme «terre» au pluriel), ou si on accepte ces variantes. Si on ne veut que le mot sans ses dérivés, on utilise l'option -w.

Pour désigner les deux fichiers où faire la recherche, on peut les écrire littéralement :

zola1.txt zola2.txt

ou, mieux, utiliser les joker du shell :

zola[12].txt

[12] signifie «le caractère 1 ou le caractère 2».

Finalement, on peut taper, au choix :

```
grep -inw -f liste zola1.txt zola2.txt
```

```
grep -inw -f liste zola[12].txt
```

```
grep -inw \"(ciel\\|terre)\" zola1.txt zola2.txt
```

```
grep -inw \"(ciel\\|terre)\" zola[12].txt
```

Et on obtient :

zola1.txt:13:ciel, le pavé se déroulait avec la rectitude d'une jetée, au milieu de

zola1.txt:36:brûlaient si haut dans le ciel mort, pareils à des lunes fumeuses.

Mais, au

zola1.txt:50:besogne. Les ouvriers de la coupe à terre avaient dû travailler tard, on

zola1.txt:124:terre, lorsqu'un accès de toux annonça le retour du charretier. Le ntément,

zola1.txt:191:bleues en plein ciel, comme des torches géantes. C'était d'une tristesse

zola1.txt:207: Le manœuvre, après avoir vidé les berlines, s'était assis à terre,

zola1.txt:222:fois avec tout le poil roussi, une autre avec de la terre jusque dans le

(...)

Le résultat est un peu différent quand on n'utilise pas l'option -w.

grep recherche des chaînes de caractères, qui peuvent être un mot complet («terre»), une suite de lettres («tre»), ou une expression régulière. Les expressions régulières sont des formules qui représentent des chaînes de caractères. On cherche alors non pas un mot précis, mais des suites de caractères correspondant aux critères demandés. Elles sont d'un usage fréquent avec grep bien sûr, mais aussi avec des commandes comme less, ou encore au sein d'un éditeur.

«Expressions régulières» (Regular expressions en anglais) se traduisent en bon français par «expressions rationnelles», mais l'usage est de dire «régulières».

Ces exercices n'entendent pas remplacer un cours sur les expressions régulières, ni faire le tour de leurs possibilités.

65) Chercher toutes les lignes commençant par «a» ou «A».

Il faut indiquer que l'on veut le début de la ligne, avec le chapeau (caret en anglais). Ensuite, on veut préciser que la ligne commence par un «a» minuscule ou majuscule. Il y a deux façons de le faire :

Utiliser l'option -i qui fait ignorer la différence entre les majuscules et le minuscules.

Dire que l'on cherche un «a» ou un «A». C'est à cela que servent les crochets : [abc] signifie «a ou b ou c». Ici, ce sera [aA].

Enfin, il faut protéger les signes contre le shell, pour qu'il ne les interprète pas; on met donc l'expression entre apostrophes. Remarque : la protection des expressions régulières contre le shell est une question complexe....

Il faut donc écrire :

```
grep -i '^a' fichier  
ou
```

```
grep '^[aA]' fichier
```

66) Chercher toutes les lignes finissant par «rs»

C'est le dollar (\$) qui représente la fin de la ligne. Il faut donc écrire :

```
grep 'rs$' fichier
```

67) Chercher toutes les lignes contenant au moins un chiffre

Pour désigner un chiffre, on peut en indiquer une liste entre crochets : [0123456789]. Il est plus simple d'utiliser une classe de caractères : [0-9] qui désigne, comme la solution précédente, n'importe quel chiffre de zéro à neuf.

Il faut donc taper :

```
grep '[0-9]' fichier
```

68) Chercher toutes les lignes commençant par une majuscule

Comme on l'a vu, c'est le chapeau qui indique le début de la ligne. Pour indiquer que l'on cherche une majuscule, on peut soit en donner une liste ([ABCDEFGHJKLMNPQRSTUVWXYZ]), soit utiliser une classe de caractères : [A-Z], la seconde solution étant, de loin, préférable...

Il faut donc taper :

```
grep '^[A-Z]' fichier
```

69) Chercher toutes les lignes commençant par «B», «E» ou «Q»

Il faut indiquer entre crochets les trois lettres recherchées :

```
grep '^[BEQ]' fichier
```

Chercher toutes les lignes finissant par un point d'exclamation

Le point d'exclamation n'a pas de signification particulière avec grep, on peut donc le mettre tel quel :

```
grep '!$' fichier
```

70) Chercher toutes les lignes ne finissant pas par un signe de ponctuation (point, virgule, point-virgule, deux-points, point d'interrogation, point d'exclamation)

Il faut donner une liste de caractères, que l'on ne veut pas voir figurer; la liste sera entre crochets, comme on l'a déjà vu, et c'est le chapeau qui signifiera, dans ce contexte, «sauf». Par exemple, si on cherche tous les «a», sauf ceux suivi de «b», «c» ou «t», on écrit :

```
grep 'a[^bct]'
```

Il y a une seconde difficulté, qui vient de ce que certains caractères sont spéciaux avec grep. Vous vous doutez que le chapeau est spécial quand il est placé au début de l'expression, et que le dollar l'est quand il est placé en fin d'expression. Dans notre cas :

Le point désigne n'importe quel caractère.

Le point d'interrogation signifie «le caractère qui précède apparaît 0 ou 1 fois». Avec egrep, il fonctionne tout seul, avec grep, il faut le faire précéder d'un backslash pour qu'il fonctionne; par exemple (avec grep), pour chercher «charbon» ou «vagabond», on écrit :

```
grep 'ar\?bo' fichier
```

(chercher la suite de lettre «abo» avec un «r» facultatif entre le «a» et le «b»).

Pour que grep interprète littéralement ces caractères, et ne les considère plus comme spéciaux, il faut les faire précéder d'un backslash (\). Si par exemple vous cherchez toutes les lignes qui se terminent par un point, il faut taper :

```
grep '\.$' fichier
```

Dans notre cas cependant, ces caractères sont protégés par les crochets. On peut donc écrire :

```
grep '[^.,;?!]$\$' fichier
```

On peut aussi utiliser l'option -v, qui prend toutes les lignes où ne figure pas une chaîne de caractères donnée; dans ce cas, on tape :

```
grep -v '[^.,;?!]$\$' fichier
```

71) Comment chercher tous les mots contenant un «r» précédé de n'importe quelle lettre majuscule ou minuscule ?

On cherche une chaîne de caractères qui soit indifféremment au début ou au milieu d'un mot. N'importe quelle lettre, ce sont les classes de caractères [a-zA-Z] ou [:alpha:], qui sont équivalentes.

Il y a une petite subtilité avec l'emploi de classes du second type; elles désignent un groupe de caractères, et il faut mettre une seconde paire de crochets pour dire «n'importe quel caractère de cette classe prédéfinie». On tape donc au choix :

```
grep '[a-zA-Z]r' fichier
```

ou

```
grep '[:alpha:]r' fichier
```

Attention, dans ces listes ne sont pas compris les caractères accentués...

72) Chercher tous les mots dont la seconde lettre est un «r».

C'est le symbole \< qui désigne un début de mot. La première lettre du mot est indifférente, la seconde est un «r».

On écrit donc :

```
grep '\<.r' fichier
```

Il y a cependant un problème avec les caractères accentués, que grep considère comme des blancs. Dans ce cas, il vaut mieux procéder autrement : un mot est précédé d'un début de ligne, ou d'un blanc ou d'une tabulation. Un début de ligne, c'est le chapeau, un blanc ou une tabulation, c'est la classe de caractères [:space:].

On va se servir du pipe (|) qui signifie «ou». Avec grep, il faut backslasher le pipe, avec egrep ce n'est pas nécessaire. On écrit donc (avec grep) :

```
grep '^r\|[:space:]r' fichier
```

Ce n'est quand même pas si simple; les mots peuvent être précédés d'un tiret (mots composés), d'une apostrophe, de guillemets divers (` , " , « , << ),



La redirection d'entrées/sorties est une caractéristique de shell. Les commandes vont soit lire des données (entrée) soit les transmettre (sortie)

42) >

renvoie le résultat dans un fichier (si celui-ci existe, il sera écrasé).

ex: info@debian:~\$ ls > file1 la sortie de ls est dirigé vers le fichier file1; désormais file1 contient le résultat de la commande ls

43) >>

renvoie le résultat dans un fichier (si celui-ci existe déjà, ajoute le résultat à la fin) il existe deux sorties : 1 la sortie normale (écran), 2 la sortie d'erreurs

ex: info@debian:~\$ who >> file2

info@debian:~\$ date >> file2

l'affichage de la date est mis à la fin de file2

43) 2>

crée un fichier pour les erreurs.

info@debian:~\$ rmdir dossier1

supposons que dossier1 n'est pas vide un message d'erreur sera affiché

info@debian:~\$ rmdir 2> file1

le fichier file1 contiendra le message d'erreur, si file1 existe, il sera écraser

44) 2>>

crée un fichier pour les erreurs. (si le fichier existe déjà, la commande ajoute le résultat à la fin)

45) 2>&1

fusionner les sorties dans un seul et même fichier, rediriger la sortie d'erreur (2: stderr) au même emplacement redirigé par la sortie standard (1: stdout)

supposons que le fichier test.txt n'existe pas

info@debian:~\$ cat test.txt > err.txt 2>&1

va envoyer les erreurs dans le même fichier et de la même façon que la sortie standard

46)<

prend un fichier en entrée

47)<<

prend en entrée le clavier au fur et à mesure. Ceci nous permet de passer des données directement à une commande sans avoir besoin de créer de fichier. Mettons que nous voulions trier des prénoms par ordre alphabétique. Nous allons pour cela invoquer la commande sort, mais au lieu de créer un fichier texte puis de le faire réorganiser par sort, nous allons directement lui soumettre les noms en les entrant au clavier

info@debian:~\$ sort nom\_trie

khalid

ahmed

mohamed

info@debian:~\$ nom\_trie

ahmed

khalid

mohamed

## Manipulation

1) Essayer d'utiliser 2>>&1

2) Compter les fichiers d'un répertoire (ls et wc)

3) Enregistrer dans un fichier les noms, terminaux et heures de connexion de tous les utilisateurs connectés

4) Rajouter la date du jour et le résultat de la commande whoami

5) Faites afficher les fichiers du répertoire /bin page par page

6) si on tape uniquement cat ?

7) quel est le résultat de : cat xyz > xyz

8) Que se passe-t-il si on utilise >> au lieu de >

9) Comment peut-on utiliser cat pour placer un texte dans un fichier?

10) La séquence de commandes précédentes est-elle correcte sur le plan de la syntaxe? Quel est son effet?

- 11) Recherchez dans un fichier /etc/passwd une ligne contenant votre nom
- 12) Ecrire le message "Bonjour" dans un fichier appelé "test" en redirigeant la sortie de la commande echo.
- 13) Ecrire le message " au revoir " dans le même fichier " test " en redirigeant la sortie de la commande echo et sans écraser le contenu de " test " vérifier avec cat
- 14) Exécuter la commande find /etc -name hosts, y a-t-il des messages d'erreurs qui sont affichés? rediriger les messages d'erreur de la commande précédente vers le fichier "err.txt "
- 15) Rediriger maintenant la sortie standard et la sortie d'erreur de la commande find /etc -name hosts vers deux fichiers différents (std.out et std.err)
- 16) Exécuter les deux commandes suivantes:  
info@debian:~\$ find /etc -name hosts >find1.out 2>&1  
info@debian:~\$ find /etc -name hosts 2>find2.out >&2  
(2>&1 signifie rediriger la sortie d'erreur vers le même emplacement que la sortie standard)  
comparer le contenu des deux fichiers find1.out et find2.out, quelles sont vos remarques
- 17) Exécuter la commande tr 'a-z' 'A-Z', une fois exécutée la commande tr attend du texte à partir de l'entrée standard saisissez 3 lignes de textes à faire passer comme argument à tr; pour arrêter la saisie pressez ctrl+d
- 18) Rediriger l'entrée de tr 'a-z' 'A-Z' depuis le fichier " test " de la question 13.
- 19) En utilisant la commande tr 'a-z' 'A-Z' convertir le contenu de « find1.out » en majuscule puis le placer dans un fichier « FIND1.out », vérifier avec cat.
- 20) Rediriger la sortie de la commande ls -l /etc vers un fichier file1 avec la commande more.
- 21) le fichier /etc/passwd contient les informations sur tous les comptes utilisateurs du système afficher le contenu de /etc/passwd avec more.  
/etc/passwd contient plusieurs champs, le premier champ représente le nom de l'utilisateur, quel est le délimiteur qui sépare ces champs
- 22) En utilisant la commande cut extraire depuis /etc/passwd les noms des utilisateurs
- 23) Trier la sortie de la commande cut précédente par ordre alphabétique puis afficher le résultat avec more
- 24) Numéroté les lignes de /etc/passwd avec nl
- 25) En utilisant la commande nl précédente et les commande head et tail afficher les lignes de /etc/passwd comprises entre la ligne 7 et la ligne 12
- 26) (a) Exécuter ls dans un terminal. Taper ensuite ls > sorties, puis lister le contenu du répertoire courant. Un nouveau fichier appelé sorties a été créé. Lire son contenu.  
(b) Exécuter maintenant la commande ls -l > sorties. Consulter à nouveau le contenu de sorties.  
(c) Concaténer le résultat de la commande ls >> sorties. à la fin du fichier sorties.  
(d) Comparer les résultats de wc < sorties et wc sorties. Expliquer la différence.  
(e) Ajouter le résultat de wc < sorties dans le fichier sortie.  
(f) Exécuter la commande wc test1. Comme le fichier test1 n'existe pas, wc vous renvoie une erreur.  
(g) Exécuter maintenant wc test1 > sortie. Regarder le contenu de sortie.  
(h) Faire en sorte que le message d'erreur soit écrit dans un fichier nommé erreur  
i) Dans le terminal, taper la commande cat puis saisir le texte suivant :  
"Aujourd'hui, on étudie les redirections." (valider avec la touche enter)  
j) La saisie est affichée deux fois. Expliquer.  
k) Comment faire pour terminer cette commande?  
l) Taper la commande suivante : cat > test et saisir le texte suivant :  
"Un seul chevron, et le texte précédent disparaît. (enter) Mais avec deux, on écrit à la fin du fichier."  
m) Le texte ne s'affiche plus qu'une fois. Pourquoi ?  
n) Utiliser la commande cat pour copier le fichier sorties dans le fichier test.  
o) Utiliser la commande cat pour concaténer les fichiers sorties et erreur dans le fichier exercice.  
p) Ajouter à la fin du fichier sortie, le contenu du fichier erreur  
q) En une commande, ajouter à la fin de sortie la phrase suivante : "Mais ces redirections, sont-elles assistées ?"  
r) On veut maintenant ajouter au début du fichier sortie le texte suivant : "Les redirections, ça va dans tous les sens."  
s) A l'aide de cat et d'une redirection vers le fichier tmp1, stocker ce qui aura été saisi sur l'entrée standard.  
t) Stocker dans tmp2 la concaténation de tmp1 et sortie.  
u) Renommer le fichier tmp2 en sortie, puis détruire le fichier tmp1.

met & à la fin d'une commande permet d'en lancer une autre sans attendre la fin de la première  
ex. : info@debian:~\$ cp video.avi /home/info/copie-video.avi &  
Elle permet aussi de passer la commande en arrière-plan et redonne immédiatement accès au shell.

#### 49)nohup

lance le programme et le maintient même une fois la console fermée. Les sorties 1 et 2 sont redirigées vers nohup.out.

ex: info@debian:~\$ nohup cp video.avi copie\_video.avi

nohup: ajout à la sortie de `nohup.out`

la commande continue de fonctionner sauf si on utilise la commande kill

#### 50)Ctrl+z

met en pause le processus en cours

#### 51) bg

passé le processus qui est en pause en arrière-plan.

#### 52)fg

reprend un processus en premier plan (si plusieurs tournent en même temps, fg %n°).

#### 53)at

programme une tâche à exécuter à une heure ultérieure ex. : info@debian:~\$ at 18:22

info@debian:~\$ at now + 5hours puis ctrl + d.

#### 54) atq

liste les jobs en attente.

#### 55)atrm

supprime des jobs.

#### 56)sleep

cette commande permet de faire une pause entre l'exécution de deux commandes. Exemple : touch gt.txt && sleep 10 && rm gt.txt. La pause est exprimée en secondes par défaut, il est cependant possible de changer cela en faisant suivre le nombre d'une unité : m, h, ou d pour respectivement les minutes, heures et jours.

#### 57)crontab

crontab est en fait une commande qui permet de lire et de modifier un fichier appelé la " crontab ".

e

modifie la crontab.

l

affiche la crontab actuelle.

r

supprime votre crontab. Attention, la suppression est immédiate et sans confirmation .

#### 58)screen

multiplexeur de terminaux. Sous ce terme un peu barbare se cache en fait une sorte de terminal virtuel.

Vous êtes au boulot, vous ouvrez un terminal et vous le nommez, vous lancez une tâche (un script qui va ré-encoder plusieurs GB de vidéos par ex., ce qui prend du temps), et vous vous déconnectez, vous arrivez chez vous, vous rouvrez votre terminal et retrouvez votre tâche comme si tout était resté ouvert en face de vous.

### Manipulation

1) Quelle est la syntaxe utilisée par Linux pour exécuter un processus en arrière-plan?

2) Quel est le terme qui décrit un processus lancé dans un terminal shell?

3) Une commande est exécutée en arrière-plan, mais on veut que le processus soit exécuté au premier plan:

info@debian:~\$ sleep 100 &

[1] 3180

info@debian:~\$ jobs

[1]+ running sleep 100 &

4) Le processus est maintenant exécuté au premier plan. quelle est l'interruption utilisée pour l'arrêter (et non le tuer)?

5) Le processus est maintenant interrompu. Comment reprendriez-vous l'exécution en arrière-plan?

6) Quelles touches utilisées pour envoyer un SIGINT à un processus au premier plan?

7) On exécute les commandes suivantes dans un terminal:

```
info@debian:~$ ./script &
```

```
[1] 3450
```

L'exécution du processus prend beaucoup de temps, par conséquent on quitte le terminal actuel. Qu'est-il arrivé?

8) Quelles solutions pouvez-vous donner pour éviter ce qui vient de se passer?

(on utilise `info@debian:~$ nohup ./script &`)

9) Essayez la commande `sleep 5`. Que fait cette commande?

10) Exécutez la commande en arrière-plan en utilisant `&`.

11) Exécutez `sleep 15` au premier plan, suspendez-le avec `Ctrl-z` puis placez-le à l'arrière-plan avec `bg`. Tapez `jobs`. Tapez `ps`. Ramenez le travail au premier plan avec `fg`.

12) Exécutez `sleep 15` en arrière-plan avec `&`, puis utilisez `kill` pour terminer le processus par son numéro de job. Répéter, sauf que cette fois tuer le processus en spécifiant son PID.

13) Exécutez `sleep 15` en arrière-plan à l'aide de `&`, puis utilisez `kill` pour suspendre le processus. Utilisez `bg` pour continuer à exécuter le processus.

14) Démarrez un certain nombre de processus `sleep 60` en arrière-plan et terminez-les tous en même temps à l'aide de la commande `pkill`.

15) Utilisez `ps`, `w` et `top` pour afficher tous les processus en cours d'exécution.

16) Utilisez `ps -aeH` pour afficher la hiérarchie des processus. Recherchez le processus `init`. Voyez si vous pouvez identifier des importants démons système. Pouvez-vous également identifier votre shell et ses sous-processus?

17) Combinez `ps -fae` avec `grep` pour afficher tous les processus que vous exécutez, à l'exception des commandes `ps` `-fae` et `grep`.

18) Démarrez un processus `sleep 300` exécuté en arrière-plan. Déconnectez le serveur et reconnectez-vous.

19) Répertoirez tous les processus que vous exécutez. Qu'est-il arrivé à votre processus `sleep`? Répétez maintenant, sauf que cette fois, commencez par exécuter `nohup sleep 300`.

## Droits , groupes et utilisateurs

### 59)sudo

exécute une commande en tant que root.

### 60)sudo su

passer root (comme su)

### 61)chmod

change les droits sur un fichier ou dossier (option `-R` pour la récursivité dans tous les fichiers et sous-dossiers du dossier sur lequel on l'applique).

si on exécute: `info@debian:~$ ls -l` le système affiche les fichiers et dossiers avec le droit d'accès

La commande reconnaît :

user (u), l'utilisateur propriétaire (le créateur du fichier ou du répertoire);

group (g), le groupe de l'utilisateur propriétaire (groupe propriétaire);

other (o), les autres, sauf l'utilisateur propriétaire et les membres de son groupe;

all (a), l'union des ensembles précédents

La commande permet 3 types d'accès aux fichiers (et répertoires) :

read (r), lecture;

write (w), écriture;

execute (x), exécution.

ex: `info@debian:~$ chmod a-wx fichier.c` (retire à tous les utilisateurs le droit d'écrire(w)et d'exécuter(x) le fichier.c;)

ex: `info@debian:~$ chmod o+rw test.txt`

Ajoute la permission aux utilisateurs "autres (others)" de lire et modifier test.txt

ex: `info@debian:~$ chmod u=r final.txt`

L'utilisateur propriétaire n'est autorisé qu'à lire le fichier final.txt.

En bref, le symbole " = " sert à fixer le type d'accès d'un ensemble d'utilisateurs alors que les symboles " + " et " - " servent, respectivement, à ajouter et à retirer un accès.

La syntaxe chiffrée peut être utilisée pour régler d'un trait l'accès de tous les ensembles d'utilisateurs:

Le premier chiffre de la valeur numérique fixe l'accès de l'utilisateur propriétaire, le deuxième fixe l'accès du groupe propriétaire et le dernier fixe l'accès des " autres ". Chaque chiffre est la somme d'un ou plusieurs chiffres correspondant à des accès différents :

4 : lecture

2 : écriture

1 : exécution

0 : aucun

ex: info@debian:~\$ chmod 764 fichier.txt

(7=4+2+1 donc l'utilisateur a le droit de lecture, écriture et exécution, 6=4+2 donc le groupe a le droit de lecture et d'écriture, 4=4 donc les autres ont le droit de lecture)

est équivalente à

info@debian:~\$ chmod ug=rw fichier.txt;chmod o=r fichier.txt

une autre façon de voir les choses

u			g			o		
r	w	x	r	w	x	r	w	x
1	1	1	1	1	0	1	0	0
111 représente 7 (décimal)			110 représente 6			100 représente 4		

## 62)adduser

ajoute un utilisateur. (vérifier aussi useradd)

ex: info@debian:~\$ sudo adduser info99

## 63)passwd

change le mot de passe d'un user | ex. : info@debian:~\$ passwd info

Pour forcer l'utilisateur à changer de mot de passe lors de sa première connexion

ex. : info@debian:~\$ sudo passwd -e info99

## 64)deluser

supprime un user (option --remove-home pour supprimer tous ses fichiers; vérifier aussi:

deluser --remove-all-files nom-utilisateur).

info@debian:~\$ sudo deluser info99

Malgré la suppression du compte, le répertoire personnel est conservé. L'ensemble des données qui y sont stockées est donc conservé. Par conséquent n'importe quel nouvel utilisateur créé ou ajouté ultérieurement avec le même UID/GID que le précédent propriétaire aura alors accès à ce répertoire et aux fichiers qui y sont stockés. La suppression du répertoire personnel doit donc être effectuée manuellement. Si le répertoire est /home/info99, la commande suivante fera le travail :

sudo rm -R /home/info99

## 65)addgroup

crée un groupe. (vérifier aussi groupadd)

ex: info@debian:~\$ sudo addgroup ami

## 66)usermod

modifie un utilisateur (options : -l pour changer le nom, -g pour lui assigner un groupe, -G pour lui assigner plusieurs groupes (séparés par des virgules), -a en complément de -g ou -G, ajouter des nouveaux groupes au lieu de tout redéfinir) ex., ajouter le groupe OS à l'utilisateur info, sans supprimer les groupes auxquels il appartenait avant : info@debian:~\$ usermod -aG OS info.

## 67)delgroup

supprime un groupe. info@debian:~\$ delgroup ami

## 68)groups

vérifie dans quels groupes est un utilisateur ex:info@debian:~\$ groups info.

#### 69)chown

change le propriétaire d'un fichier/dossier (ne peut s'utiliser qu'en root).

ex: chown info99 test.txt pour changer le groupe1 en groupe2 ex: chown info99:groupe2 test.txt

Le paramètre -h modifie uniquement les liens symboliques et pas leur cible.

Le paramètre -R modifie tous ses sous-répertoires et ses sous-fichiers d'une manière récursive.

#### 70)chgrp

change le groupe propriétaire d'un fichier (équivalent à chown user:group).

Donner le groupe "audio" au fichier "fichier"

ex: \$ chgrp audio fichier

Changer récursivement l'appartenance à un groupe. Le répertoire et les fichiers contenus dans ce répertoire changeront de groupe.

ex: \$ chgrp -R audio /home/info/repertoire

#### 71)fichier passwd

Bien qu'il ne s'agisse pas, techniquement, d'une commande, il est important de connaître la structure du fichier /etc/passwd. Lequel regroupe l'ensemble des utilisateurs du système et de leurs informations.

- 1 Nom de l'utilisateur ;
- 2 Mot de passe (x signifie que le mdp est chiffré dans le fichier /etc/shadow ;
- 3 L'id de l'utilisateur (0 est pour root et les id de 1 à 99 sont réservés pour les comptes prédéfinis) ;
- 4 ILid du groupe tel que défini dans /etc/group ;
- 5 Champ de commentaire ;
- 6 Répertoire « home » de l'utilisateur ;
- 7 Le shell par défaut (/bin/false et /usr/sbin/nologin signifient que l'utilisateur n'a pas de shell).

#### 73)stat

La commande **stat** permet d'obtenir une information plus poussée sur un fichier.

stat /etc/rc.d/rc.sysinit

stat /etc/fstab

Le fichier de démarrage /etc/rc.d/rc.sysinit possède les droits rwx r-x r-x Tous les utilisateurs ont donc le droit de lire et d'exécuter ce fichier (ce qui est à éviter); seul root peut le modifier

La table de montage /etc/fstab : rw-r--r-- peut être lue par tous, modifiée uniquement par root

#### 74) Droits supplémentaires

Parallèlement aux droits standards, des droits étendus existent et répondent à des besoins spécifiques :

- a) Pour un répertoire, restreindre la suppression au seul propriétaire : droit sticky bit.
- b) Pour un fichier binaire, exécution sous l'identité du propriétaire : droit SUID (setuid) ou sous l'identité du groupe : droit SGID (setgid).

Ce dernier point permet d'exécuter un programme (normalement un binaire et non un fichier de commandes) à partir d'un utilisateur quelconque nécessitant les droits d'un autre.

Pour affecter ou retirer ces droits, on utilise la commande chmod avec cette fois-ci sur quatre bits, le premier indiquant la nature du droit ou par une lettre spécifique en notation symbolique:

Droit	En octal	En symbolique
Sticky bit	1000	o+t
SUID	4000	u+s
SGID	2000	g+s

Le champ d'application du droit par la notation symbolique se fait clairement : u pour le SUID et g pour le SGID.

Dans le cas du sticky bit, l'application porte sur le "reste du monde" soit o, partant du principe qu'un fichier sous Ubuntu appartient à un groupe du même nom que le propriétaire.

Exemples de commandes :

chmod o+t /home/info/programmes/

chmod 4755 liste.sh

ls -l /usr/bin/chage

La première commande positionne le droit sticky bit pour le répertoire programmes situé dans le répertoire de l'utilisateur info. La deuxième commande met les droits en exécution pour le fichier comprenant des commandes shell et positionne en même temps le droit SUID. Enfin la dernière commande montre que le droit SGID est mis pour la commande chage (modifie les informations de validité d'un mot de passe) pour le groupe shadow (note : le fichier /etc/shadow a pour propriétaire le root et comme groupe shadow).

### Manipulation

1) Créez un fichier "Droit". Placez sur les valeurs suivantes les droits d'accès de votre fichier "Droit" :

a) rw-rw-r--      b) rw-----      c) rwxr-x--x      d) r-x-w---x      e) --x--xr--

puis, utilisez la forme octale

2) Créez un nouveau répertoire "RepCat", dans ce répertoire créez un fichier "FileDat" modifiez les droits d'accès avec : a) rwx    b) rw-    c) r-x    d) --x    e) -wx pour chaque lancer la commande ls et un éditeur de texte

3) Un fichier peut-il appartenir à un groupe autre que celui dont fait partie le propriétaire? Est-il possible que le numéro de groupe soit différent du numéro de groupe du propriétaire?

4) Rendre tous les nouveaux fichiers auront le droit rw-r----- et tous les répertoires rwxr-x--

5) Quelle est la différence entre :

cp fichier1 fichier2      et    cat < fichier1 > fichier2

N.B: fichier2 n'existe pas

6) a) Est-il possible de faire une copie du fichier /etc/passwd? Est-il possible de supprimer ou de modifier le fichier /etc/passwd?

b) Répéter ces tentatives en tant qu'utilisateur root.

c) A l'aide de la commande id, vérifiez votre identité et le(s) groupe(s) au(x)quel(s) vous appartenez.

Créer un fichier texte (de contenu quelconque), qui soit lisible par tout le monde, mais non modifiable (même pas par vous).

d) Créer un répertoire nommé "Secret", dont le contenu est visible uniquement par vous.

Les fichiers placés dans ce répertoire sont-ils lisibles par d'autres membres de votre groupe?

e) Créer un répertoire nommé "Connaissances" tel que les autres utilisateurs ne puissent pas lister son contenu mais puissent lire les fichiers qui y sont placés.

On obtiendra:

ls Connaissances

ls : Connaissances: Permission denied

cat Connaissances/fichier1

<...le contenu du fichier fichier1 (s'il existe)...>

7) Vérifiez l'endroit où vous êtes, votre identité de compte avec notamment l'UID et le GID par la commande id.

- Quelles sont les permissions en octal de votre répertoire personnel ? et celles du répertoire parent ?
- À quel groupe appartenez-vous ?
- Créez un fichier par la commande touch liste.sh.
- À qui appartient-il ? À quel groupe ?
- Quels sont les droits de ce fichier en octal ?
- Ajoutez le droit en écriture pour le groupe par la méthode octale.
- Enlevez le droit en écriture pour le groupe par la méthode symbolique.
- Le fichier liste.sh contiendra des commandes shell. Aussi, donnez-lui les permissions que l'on donne généralement à un programme, c'est-à-dire 755 et vérifiez-le.

8) Votre compte d'utilisateur est-il défini dans le fichier /etc/passwd? Pourquoi? Il y a-t-il d'autres alternatives?

- Quel est le répertoire de connexion de l'utilisateur root?
- Quel est le shell de l'utilisateur root?

- Quelle est la particularité de l'utilisateur nobody? Et de l'utilisateur shutdown?
- Quels sont les utilisateurs définis dans /etc/passwd qui font partie du même groupe que l'administrateur?

9) Un administrateur désire s'assurer chaque matin que tous les fichiers placés sous \$REPERTOIRE sont lisibles par tout le monde, mais non modifiables excepté par leur propriétaire. Quel doit être le mode de ces fichiers et répertoires?

10) Créer avec un dossier appartenant au groupe "omega".

Le dossier est partagé par deux utilisateurs "alfa" et "beta" appartenant au groupe secondaire "omega".

Ce dossier partagé est leur dossier d'accueil et personnel.

Ces utilisateurs peuvent lire le contenu du dossier et ajouter ou modifier des fichiers. Fixer le "sticky bit" et le "SGID" sur ce dossier en démontrant leur utilité.

En options :

Retirer à ces utilisateurs les droits d'accès à une console graphique.

Désactiver le compte de "beta" (plusieurs solutions)

11)

1. Comparer les permissions de /etc/passwd et /etc/shadow. Pourquoi a-t-on nommé ainsi ce dernier fichier ? Info (un utilisateur normal) peut-il le lire ? et voir sa présence ?

2. Par précaution, en faire une copie sous le nom shadow.bak dans /home/temp ! vérifier les droits de /home/temp/shadow.bak

3. Pensez-vous tout de même pouvoir supprimer le fichier précédent ? Concluez !

4. root fait maintenant une copie de shadow dans /home/Info, sous le nom shadow.bak et vous accorde la propriété de la copie .a) Comment fait-il ? b) vérifier le résultat

5. Vous éditez ce fichier avec un éditeur de texte, vous le modifiez, par exemple en supprimant des lignes, et vous faites une mise à jour. Ecrivez le mode opératoire. La mise à jour sera-t-elle réalisée.

5. Vous éditez ce fichier avec un éditeur de texte, vous le modifiez, par exemple en supprimant des lignes, et vous faites une mise à jour. Ecrivez le mode opératoire. La mise à jour sera-t-elle réalisée ? pourquoi ?

6. Pensez-vous que Info puisse supprimer ce fichier ? Essayez et expliquez !

12)

1. En tant que Info, pouvez-vous créer le rép. temporaire /home/temp ? essayez ! pourquoi ?

2. Effectuez cette création comme root (pensez à la commande su).

3. Accorder les permissions maximales sur /home/temp; vérifiez.

4. info99 a un droit d'écriture, dans /home/temp essaie de copier les 2 fichiers système /etc/hosts et /etc/passwd dans /home/temp ? y parviendra-t-il ? pourquoi ? que donne [info99\$] ls -l /home/temp ?

5. info99, essaie maintenant de supprimer ces 2 fichiers de /etc. Réussit-il ?

6. info99 veut que Info devienne propriétaire du fichier copié passwd. Comment s'y prend-t-il ? Réussit-il ? comment auriez-vous fait ?

13)

Il s'agit de créer un rép. partagé par tous les membres Info du groupe stagiaire Normalement, ce groupe a déjà été créé et rempli de comptes Info.

1. Créez dans /home un répertoire appelé rep-stagiaire. Rappelez pourquoi cette tâche relève des prérogatives de root

2. Faites-le appartenir au groupe stagiaire

3. Modifier les permissions sur le rép, pour que tous les membres du groupe stagiaire puissent y écrire et s'y déplacer.

4. En tant que Info, vous créez un fichier et vous le déposez dans /home/rep-stagiaire, mais en attribuant des droits 660

5. Vérifier le bon accès en lecture seulement pour les membres du groupe. Ainsi info99 qui ne fait pas partie de groupe stagiaire ne doit pas pouvoir le lire. A vérifier.

6. Un autre utilisateur de votre groupe tente de supprimer ce fichier ou de le renommer

Y parvient-il ? Essayez ! Pourtant, vérifiez que ce fichier appartient au groupe est-ce pas inquiétant ? Expliquez comment cela est possible.



7. Demandez à root de positionner le "sticky bit" sur le répertoire partagé. Vérifiez bien que le problème est réglé et protège le propriétaire des tentatives de suppression ou de changement de nom de ses fichiers. Prolongement : vérifier que cette protection s'applique aussi à distance? pourquoi ?

14) Quels sont les droits sur les répertoires personnels (par exemple Info) ?

a. Un utilisateur un compte différent info99 peut-il y pénétrer ou seulement lister ses fichiers ? et l'utilisateur Genie, le pourrait-il s'il faisait partie du groupe de Info ?

b. Quelles commandes devrait écrire Info pour accorder le droit de visite de son rép. perso seulement à Genie ?

14)

Création de 2 groupes.

Les premier et deuxième utilisateurs sont membres du premier groupe.

Les troisième et quatrième utilisateurs sont membres du second groupe.

Le deuxième utilisateur est aussi membre du second groupe.

Le quatrième utilisateur est aussi membre du premier groupe.

Outre leur répertoire de travail, les utilisateurs ont accès à un répertoire commun /home/groupe1 et/ou /home/groupe2 suivant leur groupe. Dans ce répertoire, ils peuvent écrire, créer des fichiers mais ne peuvent pas effacer les fichiers.

Détaillez les étapes de création des groupes, des utilisateurs et des répertoires en indiquant quels fichiers vous utilisez et quelles commandes vous utilisez sur ces fichiers.

Modifiez les profils des utilisateurs pour qu'au login, le répertoire d'accueil sur lequel il souhaite travailler lui soit demandé (s'il entre U, il sera sous son répertoire, s'il entre G, il sera sous le répertoire de son groupe).

Parmi ces 3 valeurs: 066, 067, 077, quel umask positionneriez vous? Expliquez pour chaque valeur quelles sont les conséquences de ce choix.

15) Création d'une instance cloud Linux (<https://aws.amazon.com/>)

Comprendre le fonctionnement de PATH

☐ Utilisez la commande echo pour afficher le contenu de votre PATH actuel (rappelez-vous qu'il doit s'appeler \$ PATH lorsque vous l'utilisez).

o Quels répertoires sont recherchés?

☐ Utilisez la commande which pour savoir où le programme figlet est installé sur votre système de fichiers.

☐ Essayez d'utiliser which -a pour voir s'il existe un autre programme appelé figlet dans PATH

☐ mettre dans / home / figlet / figlet . Exécutez ceci en lui fournissant un appel direct (en lui donnant le chemin complet). Vérifiez qu'il fonctionne correctement.

☐ Modifiez votre PATH pour ajouter / home / figlet / figlet avant les répertoires actuels (utilisez la fonction export PATH = ... et assurez-vous que le \$ PATH existant est toujours inclus).

☐ Réexécutez which figlet pour montrer que la version de / opt est maintenant trouvée en premier

☐ Exécutez figlet à partir de la ligne de commande et vérifiez que la version / opt est désormais la version par défaut.

☐ Utilisez nano pour modifier la fin de votre fichier .bashrc afin d'ajouter définitivement / opt / figlet à votre PATH.

Ouvrez un nouveau shell pour vérifier que cela fonctionne toujours

16)

Créez un compte Amazon AWS

☐ Suivez le processus de création et de connexion à une instance linux EC2

o Créez une paire de clés et téléchargez le fichier .pem correspondant.

o Créez une instance EC2 et lancez-la

o Utilisez le fichier de clé pour vous connecter à votre instance.

o Essayez certaines commandes linux pour montrer qu'ils travaillent dans cet environnement.

16) Copier le fichier /etc/passwd dans votre répertoire. Dans ce fichier chercher la ligne commençant par votre nom d'utilisateur; après ce nom on trouve ":"; chercher un nom un nom devant : et ne contenant aucun chiffre; chercher les nom devant :, ces noms doivent être d'une longueur de 4 ou 5 caractères.

17) la commande cut: cut -c Sélection de colonne [fichier]

Sélection de colonne peut représenter: une colonne individuelle (par exemple -c5) ou une plage de colonne -c3-10 ou la combinaison des deux vérifier une autre option qui est cut -dx

18) Comment créer un nouveau utilisateur sans utiliser la commande (adduser ou useradd)

19) Quelle est la permission dans le répertoire de l'utilisateur (home/utilisateur)

20) Quelle est la différence entre .bash\_profile et .bashrc?

21) Quelle est la commande pour créer un utilisateur avec un UID, shell et un répertoire de travail définis

22) Comment changer un groupe pour un utilisateur

23) On a supprimé un utilisateur, comment supprimer son répertoire de travail?

24) Nommer 3 fichiers qui sont créés automatiquement après création d'un nouveau utilisateur?

25) Quelle est la commande pour afficher tous les utilisateurs actuellement connectés?

26) Quelle est la commande pour changer et voir la date d'expiration pour tout utilisateur? (chage)

27) Comment donner à un utilisateur normal (user1) tous les droits de root?

28) Nommer 5 groupes que le super utilisateur est membre par défaut:

29) Comment pouvez-vous donner à sudo accès à n'importe quel utilisateur sans lui demander de fournir un mot de passe chaque fois qu'il exécute une commande?

30) Dans quel cas les permissions d'un fichier à sa création sont-elles différentes des permissions fixées par umask ?

31) Si on change umask le changement s'applique-t-il pour tous les sessions?.

Pour la rendre permanente, on peut intervenir sur un fichier profile : Dans le fichier profil général /etc/profile, on peut modifier la règle habituelle : `if [ $UID == 0 ] ; then umask 022 ; else umask 077 ; fi`

Pour agir au niveau des utilisateurs, ajouter la ligne umask masque dans le fichier de profil personnel

`$HOME/.bash_profile`

32) Comment un membre de votre groupe peut modifier mais pas supprimer et un autre qu'il peut supprimer mais pas modifier?

33) Écrivez un alias imposant la confirmation sur la suppression des fichiers.

Écrivez la commande " psme " permettant d'obtenir la liste de tous les processus vous appartenant

34) Vous souhaitez modifier :

1) l'invite afin qu'elle indique le nom de l'ordinateur sur lequel vous travaillez et votre répertoire courant,

2) la valeur par défaut des protections des fichiers et répertoires que vous allez créer.

De plus vous souhaitez retrouver ces modifications lors de chacune de vos entrées en session.

35) Trouvez à partir de votre répertoire d'accueil, le nombre de fichiers ayant une taille supérieure à 1 Mega-octets et stockez leurs noms dans un fichier (utilisez la commande tee)

36) Il arrive souvent de lancer une commande Linux produisant plusieurs pages d'écran à toute vitesse (par exemple : `ls l` /etc). Il faut alors relancer la même commande, en envoyant sa sortie standard dans `less`, ce qui permet d'examiner le résultat page par page

37) Créez un fichier texte et un lien dur sur ce fichier dans le même répertoire. Vérifiez que les deux noms correspondent au même inode. Changez les permissions de l'un et vérifiez que les permissions de l'autre ont suivi. Modifiez le contenu de l'un et relisez le contenu de l'autre. Supprimez l'un, que devient l'autre ? Essayez de créer un nouveau lien entre un de ces noms et un nouveau nom dans `/tmp`. Expliquez.

38) Créez un fichier texte et un lien symbolique sur ce fichier dans le même répertoire. Quelles sont les permissions du lien symbolique ? Combien de liens possède le fichier texte ? Avec la commande `more`, affichez le contenu du lien symbolique. Effacez le fichier texte. Que devient le lien symbolique ?

## Système

### 75)w

qui est connecté et fait quoi.

### 76)who

qui est connecté.

### 77)date

donne l'heure.

### 78)ntpdate

synchronise l'heure avec un serveur ntp. Il faut préciser le serveur à la commande `ntpdate pool.ntp.org`. Par ailleurs, pensez que le port NTP (123 en UDP) doit être ouvert sur votre machine.

### 79)uptime

temps depuis mise en route + charge (charge moyenne 1 - 5 - 15 mn).

### 80)free

indique l'allocation de la ram et la mémoire libre restante.

### 81)vmstat

info ram, swap, cpu.

### 82)proc/meminfo

le fichier `/proc/meminfo` contient de nombreuses informations sur la mémoire. Il suffit d'en afficher la sortie avec `cat /proc/meminfo`.

### 83)load

affiche la charge CPU sous forme de graphique.

### 84)ps -ef

affiche tous les processus lancés. Alternativement, on peut utiliser la syntaxe BSD : `ps aux`.

### 85)ps -ejH

affiche processus en arbre.

### 86)ps -u

liste les processus lancés par un utilisateur donné ex. : `ps -u info`.

### **87)top**

l'activité du système en temps réel : load, RAM, SWAP processus... top a l'avantage d'être installé presque partout.

### **88)htop**

c'est une version améliorée de top, un peu plus graphiques, les infos y sont plus claires et il est possible de trier/ordonner l'affichage selon certains critères.

### **89)glances**

similaire à top et htop, glances est le tableau de bord de votre machine, car il réunit en un coup d'œil toutes les métriques importantes : cpu, load, ram, swap, i/o disques, remplissage des disques.

### **90)atop**

on a parlé ci-dessus de top et htop, mais il y a aussi atop qui est très utile lorsqu'on doit faire un diagnostic un peu plus poussé. plus de détail dans linuxpedia (<https://www.linuxpedia.fr/>)

### **91)iotop**

dans la lignée des \*top, voici iotop qui, comme son nom le laisse entendre, permet d'avoir un aperçu temps réel de l'I/O disque.

### **92)swapoff**

swapoff -a permet de désactiver le(s) swap(s) tandis que son pendant swapon -a permet d'activer le(s) swap(s). L'exécution de swapoff -a && swapon -a permet donc de forcer la purge du swap.

### **93)kill**

tue un processus (va demander son PID).

Pour forcer un processus à se terminer, on peut envoyer le signal TERM à l'aide de la commande kill. Ce signal est passé au processus et le comportement par défaut est de terminer le processus à la réception du signal. Le signal TERM (SIGTERM) est celui envoyé par défaut par kill, il suffit donc de donner le numéro de processus (son PID) à fermer après le nom de la commande.

N.B: si on récupère le pid d'un processus par exemple firefox

ex: \$ pidof firefox

3143

pour le fermer correctement

\$ kill 3143

Si on arrive pas à fermer un programme qui a par exemple planter, il est intéressant d'utiliser kill avec une option particulière permettant de forcer la fermeture du processus. La commande kill permet d'envoyer des signaux à des processus, le signal utilisé pour forcer la fermeture d'un programme est SIGKILL. Sachez également que les signaux sont numérotés, le signal SIGKILL a le numéro 9.

Pour forcer la fermeture d'un processus, utilisez la commande kill avec l'option -s, cette option permet de spécifier le signal à envoyer.

Dans l'exemple ci-dessous, nous envoyons le signal SIGKILL à un processus ayant le PID 3143. Le processus va être tué immédiatement.

ex: \$kill -s 9 3143

ou bien \$ kill -s SIGKILL 3143

### **94)kill -9**

force à quitter.

### **95)killall**

quitte toutes les occurrences d'un programme.

### **95 a) kill -l**

Donne la liste des signaux

Remarque :

Linux supporte à la fois les signaux POSIX classiques (« signaux standards ») et les signaux POSIX temps-réel.

Dispositions de signaux Chaque signal a une disposition courante qui détermine comment le processus se comporte lorsqu'un signal lui est délivré.

Les symboles de la colonne « Action » des tables ci-dessous spécifient la disposition par défaut pour chaque signal :

Term Par défaut, terminer le processus.

Ign Par défaut, ignorer le signal.

Core Par défaut, terminer le processus et créer un fichier core (voir core(5)).

Stop Par défaut, arrêter le processus.

Cont Par défaut, continuer le processus s'il est arrêté

Signal	Valeur	Action	Commentaire
<b>SIGINT</b>	2	Term	Interruption depuis le clavier.
<b>SIGQUIT</b>	3	Core	Demande « Quitter » depuis le clavier.
<b>SIGILL</b>	4	Core	Instruction illégale.
<b>SIGABRT</b>	6	Core	Signal d'arrêt depuis <b><u>abort</u></b> (3).
<b>SIGFPE</b>	8	Core	Erreur mathématique virgule flottante.
<b>SIGKILL</b>	9	Term	Signal « KILL ».
<b>SIGSEGV</b>	11	Core	Référence mémoire invalide.
<b>SIGPIPE</b>	13	Term	Écriture dans un tube sans lecteur.
<b>SIGALRM</b>	14	Term	Temporisation <b><u>alarm</u></b> (2) écoulée.
<b>SIGTERM</b>	15	Term	Signal de fin.
<b>SIGUSR1</b>	30,10,16	Term	Signal utilisateur 1.
<b>SIGUSR2</b>	31,12,17	Term	Signal utilisateur 2.
<b>SIGCHLD</b>	20,17,18	Ign	Fils arrêté ou terminé.
<b>SIGCONT</b>	19,18,25	Cont	Continuer si arrêté.
<b>SIGSTOP</b>	17,19,23	Stop	Arrêt du processus.
<b>SIGTSTP</b>	18,20,24	Stop	Stop invoqué depuis tty.
<b>SIGTTIN</b>	21,21,26	Stop	Lecture sur tty en arrière-plan.

**SIGTTTOU**                      22,22,27                      Stop                      Écriture sur tty en arrière-plan.

**96)reboot**

redémarre le système d'exploitation.

**97)shutdown**

programme un redémarrage ou un arrêt.

**98)poweroff**

bien qu'assez similaire à shutdown dans la mesure où elle permet d'éteindre le système, poweroff permet aussi selon les arguments qui lui sont passés, de rebooter ou de changer de runlevel.

**99)halt**

permet « l'arrêt » du système. Je mets arrêt entre guillemets, car le système peut rester sous tension avec cette commande (selon les options passées et les paramètres par défaut du système).

**100)last**

historique des connexions.

**101)df**

remplissage des disques (l'option -h permet d'obtenir les tailles en " human readable ").

**102)mount**

permet de monter le périphérique d'un système de fichiers sous un répertoire local. Par exemple, pour monter la partition /dev/sdb1 au point de montage /home : mount /dev/sdb1 /home/.

**103)umount**

permet de démonter le périphérique d'un système de fichiers : umount /dev/sdb1  
Un disque dur est généralement organisé de la façon suivante :



Le Master Boot Record (MBR) est situé dans les premiers secteurs du disque. Le mode de partitionnement MBR est le mode historique des ordinateurs de type IBM PC-compatible et le plus répandu. Dans ce modèle, une zone de 512 octets appelée le Master Boot Record (MBR) est réservée en début de disque pour contenir l'information relative à un maximum de 4 partitions.

Le MBR est constitué de 2 parties :

- ☐ La table de partition ;
- ☐ Le programme d'amorçage qui charge le noyau du système d'exploitation.

Dans un disque dur, plusieurs types de partitions peuvent coexister. On peut mentionner les partitions principales, étendues et logiques.

- ☐ Les partitions principales :
  - Sont au maximum de 4
  - Accepte tout type de système de fichiers.
- ☐ Les partitions étendues :
  - Sont destinées à contenir des partitions logiques et non au système de fichiers,
  - Nécessitent au moins une partition principale.
- ☐ Les partitions logiques :
  - Sont contenues dans une partition étendue,
  - Accepte tout type de systèmes de fichiers. Sous Linux, le pointeur spécial /dev permet l'accès aux disques.

Pour dénommer les disques, un format est adopté : /dev/XYZ où

- ☐ XX désigne le type de bus :

- hd pour les périphériques de type IDE
- sd pour les périphériques de type SATA
- Y désigne la lettre de périphérique :
- "a" pour le maître de la nappe IDE primaire
- "b" pour l'esclave de la nappe IDE primaire
- "c" pour le maître de la nappe IDE secondaire
- "d" pour l'esclave de la nappe IDE secondaire
- Z désigne le numéro de la partition

Exemples :

- /dev/hda1 : partition 1 sur le 1er disque IDE
- /dev/sdb2 : partition 2 sur le 2ème disque SATA

#### 104)fdisk

permet de gérer les partitions. Affiche la table des partitions si on utilise l'option -l. Pour créer ou modifier des partitions, on utilisera le mode interactif fdisk /dev/sdx ou /dev/sdx est le chemin vers votre disque. Ensuite, l'option m listera toutes les commandes possibles.

#### 105)parted

semblable à fdisk, parted supporte les partitions GPT (GUID Partition Table). L'option -l permet d'afficher la table des partitions.

#### 106)fsck

cet utilitaire permet de vérifier et de réparer le système de fichiers.

#### 107)dd

l'outil permet d'effectuer des opérations sur les disques, notamment de les effacer, par exemple dd if=/dev/zero of=/dev/sdc ou de cloner un disque ; ex. : dd if=/dev/sda1 of=/dev/sdb1 bs=64K conv=noerror,sync.

#### 108)ddrescue

lorsqu'on fait face à un disque dur endommagé, il vaut mieux tenter de le cloner avec ddrescue qu'avec dd. Le premier est en effet dédié à cet usage. Il peut effectuer une première passe pour récupérer le maximum de données en ignorant les secteurs endommagés, puis dans une seconde passe il tentera de récupérer les données endommagées. La doc d'Archlinux détaille bien le processus de recovery. (<https://wiki.archlinux.org/>)

#### 109)hdparm

la commande hdparm permet de récupérer des infos, de tester et de modifier la configuration - bas niveau - des disques durs. On peut par exemple modifier la taille des buffers avant écriture, activer ou désactiver ou modifier l'économie d'énergie, modifier la vitesse de mouvement des têtes de lecture  
hdparm -I affiche toutes les infos sur le disque (type de disque, taille des caches, vitesse de rotation, options supportées...)  
hdparm -tT /dev/sda fournit la vitesse de lecture pour les contenus avec et sans cache (à répéter plusieurs fois avec le serveur idle pour des résultats pertinents). Enfin, hdparm s'avère parfois intéressant en cas de secteurs défectueux.

#### 110)smartctl

smartctl permet d'afficher les informations smart d'un disque. smartctl -a /dev/sda affiche toutes les informations à propos de sda. La fiche Wikipedia à propos du SMART fournit de bonnes explications sur les différentes données et leur interprétation.

#### 111)/proc/mdstat/

fichier qui contient les infos sur vos RAID logiciels. On peut afficher ces informations en faisant un cat : cat /proc/mdstat. Lisez l'article sur les différents niveaux de RAID (simulateur intégré) pour trouver le RAID qui vous correspond. Consulter ce wiki [en] pour des infos délivrées par mdadm.

#### 112)mdadm

commande qui permet d'obtenir des infos sur les RAID soft et de les paramétrer (sortir un disque de l'array, en rajouter un, le reconstruire en cas de disque défaillant...). Par exemple mdadm --detail /dev/md0 donnera tous les détails liés à un l'array. Notez que cette commande est assez proche de l'option --examine, mais laquelle s'applique à un disque constitutif d'un raid --examine /dev/sd\* et non au volume raid. Cette commande étant très riche et servant de nombreuses fonctions

#### **113)lsuf**

list open files, dresse la liste des fichiers ouverts. Comme le fait remarquer l'ami MagiCrazy dans les commentaires, cette commande peut s'avérer bien utile pour voir quel fichier bloque le démontage d'un filesystem par exemple.

#### **114)hostname**

affiche le nom d'hôte de la machine conformément à ce qui est écrit dans le fichier /etc/hostname.

#### **115)uname**

infos sur le système et le matos.

#### **116)lsb\_release**

lsb\_release -a donne toutes les infos sur la distrib.

#### **117)lshw**

donne une liste détaillée de l'hardware système, par exemple la configuration ram, la version du firmware, la configuration de la carte mère... Avec l'option -short vous obtiendrez une sortie plus digeste. L'option -c network s'avère aussi bien utile pour connaître le nom d'une interface réseau encore non configurée avec le standard de nommage systemD [en].

#### **118)lsblk**

liste tous les devices de type bloc (disque dur).

#### **119)lspci**

liste tous les périphériques PCI.

#### **120)lsusb**

liste tous les périphériques USB.

#### **121)/proc/version**

fichier qui contient des infos sur le noyau. On peut afficher son contenu avec cat : cat /proc/version.

#### **122)/proc/cpuinfo**

fichier qui contient des infos sur le processeur. On peut afficher son contenu avec cat : cat /proc/cpuinfo.

#### **123)sysctl**

cette commande permet d'afficher et de configurer les paramètres du noyau . Très pratique avec un coup de grep, sysctl -a vous donnera tous les paramètres du noyau. L'option -p est également intéressante, car elle permet de recharger les paramètres soit depuis /etc/sysctl.conf par défaut, soit depuis un fichier passé en paramètre.

#### **124)dmiencode**

lit les infos du bios.

#### **125)dmesg**

affiche les messages du buffer du noyau.

#### **126)apt-cache**

gestion des paquets. Deux options sont très utiles apt-cache search nom\_paquet, permet de chercher un paquet, et apt-cache show, permet d'obtenir des détails sur un paquet.

#### **127)apt-get**

gestion des paquets. Les commandes que l'on utilisera le plus sont update (MAJ des sources de paquets dispos), upgrade (mise à jour du système et autres softs), install (apt-get install truc-à-installer pour installer un nouveau logiciel et ses dépendances), purge (permet de désinstaller un paquet de manière plus " propre " que remove, car cela efface aussi les fichiers de configuration).

#### **128)apt**



apparue assez récemment, apt est décrit dans son man comme le front-end utilisateur pour un usage plus interactif d'autres outils spécialisés tels que apt-get ou apt-cache. Il offre dans l'ensemble les mêmes possibilités que apt-get. Je retiens une commande toute particulière : apt list --upgradable qui permet de lister les packets qui seront mis à jour si l'on fait un upgrade (avec apt ou apt-get).

### 129)aptitude

c'est un autre utilitaire de paquets. Plus récent qu'apt-get, il est installé en parallèle de celui-ci sur Ubuntu et Debian. Préférez-le à apt-get. Il s'utilise dans l'ensemble comme apt-get, mais est plus performant.

### 130)add-apt-repository

add-apt-repository permet d'ajouter des dépôts alternatifs aux dépôts officiels. C'est très utile, car les dépôts officiels ont souvent du retard sur les versions de logiciels que sortent les développeurs et certains logiciels en sont même absents. Ainsi, en ajoutant par exemple les dépôts des développeurs, vous pouvez bénéficier des dernières versions juste en vous servant de apt-get ou aptitude, sans avoir besoin de compiler !

Par exemple, FFMPEG avait été supprimé des dépôts officiels d'Ubuntu (réintégré en version 15.04) au profit de Libav, son fork. Pour profiter des dernières versions de FFMPEG sans avoir à compiler manuellement à chaque fois :

```
sudo add-apt-repository ppa:kirillshkrogalev/ffmpeg-next 2.
```

```
# on met notre liste de packets à jour
```

```
sudo apt-get update 5.
```

```
# si ffmpeg n'est pas installé, on l'installe
```

```
sudo apt-get install ffmpeg 8.
```

```
# s'il est déjà installé,
```

```
# un upgrade se chargera de le mettre à jour
```

```
sudo apt-get upgrade
```

### 131)une combinaison de commandes qui permettent de rechercher un dépôt sur le système :

```
grep ^ /etc/apt/sources.list /etc/apt/sources.list.d/* | grep nom_du_ppa
```

### 132)apt-key

cette commande va bien souvent de pair avec l'ajout de dépôts puisqu'elle permet de gérer les clefs cryptographiques en validant l'authenticité. On utilisera le plus souvent apt-key adv pour ajouter de nouvelles clefs, Ubuntu-fr en détaille très bien l'usage. apt-key list permet de lister toutes les clefs installées et apt-key del permet d'effacer une clef.

### 133)apt-cache madison

autre commande bien pratique, apt-cache madison nom\_du\_packet affiche les différents dépôts liés à un packet donné et la version actuelle de chacun d'entre eux.

### 134)dpkg

info sur les paquets installés (options pour lister tous les paquets, désinstaller, etc.) ex. : liste des paquets installés dpkg --get-selections. L'option -l fournit également une liste exhaustive et avec une petite description de chaque packet, ce qui peut s'avérer très pratique. On constate parfois qu'un grand nombre de paquets sont marqués pour être désinstallés avec le tag deinstall, pour tout enlever d'un coup dpkg --purge `dpkg --get-selections | grep deinstall | cut -f1`

### 135)update kernel

si votre partition /boot est indépendante et qu'elle n'est pas très grande, il est probable qu'après un certain temps, vous deviez faire un peu de ménage, sans quoi l'espace nécessaire à une mise à jour du kernel est insuffisant. On devra donc supprimer les anciens noyaux, pour ce faire :

```
apt-get purge $(dpkg -l linux-{image,headers}-"[0-9]*" | awk '/ii/{print $2}' | grep -ve "$(uname -r | sed -r 's/-[a-z]+//'))
```

Il ne vous reste alors plus que le kernel actuellement utilisé et vous pouvez maintenant effectuer votre mise à jour sans problème. Vous trouverez plus de détails sur cette commande sur le site de askubuntu.

### 136)service

cette commande permet de gérer les services. Lancer, arrêter, lister les services du système, etc. Par exemple, pour relancer nginx après un changement dans le fichier de config, on fera `service nginx restart` (notez que `reload` suffit dans certains cas). La commande `service --status-all`, bien pratique, permet de lister tous les services disponibles sur le système.

Le paramètre `status` s'avère souvent fort utile puisqu'il donne des informations sur un service en particulier, notamment s'il est actif ou non. Exemple `service nginx status`.

On peut également activer ou désactiver le démarrage automatique des services au boot, pour cela, on doit troquer la commande `service` pour la commande native `systemctl`. Prenons `apache2` pour exemple

```
# activer/désactiver un service
systemctl enable apache2
systemctl disable apache2
# vérifier si un service est lancé au boot
systemctl is-enabled
systemctl is-enabled apache2
```

### 137)journalctl

cette commande, qui accompagne `systemd`, permet de visionner et requêter les journaux système. Nous ne pourrions bien entendu pas aborder toutes les options ici, cependant, à vous d'aller fouiller le web pour répondre à vos cas d'usage (requête sur un horaire donné, process, etc.). La commande `journalctl` seule, affiche l'ensemble des logs que `journald` a collectés.

Que s'est-il passé durant le dernier boot, ou le précédent ?

```
# on commence par lister les différents boots présents dans les logs
# (il se peut qu'il n'y en ait qu'un)
journalctl --list-boots
0 3774424e8f5a45cb930b1eef3b8355dd Tue 2017-08-15 14:45:48 CEST—Wed 2017-08-16 12:54:57
CEST
# on demande ensuite à afficher les logs correspondants à ce boot
journalctl -b -0
- Logs begin at Tue 2017-08-15 14:45:48 CEST, end at Wed 2017-08-16 12:56:23 CEST. --
[...]
```

### 138)make

permet de compiler un programme dont on détient les sources. En général on fait tout d'abord `./configure` [à lancer avec `--help` pour voir les différentes options de compilation] (lance le script de configuration qui vérifie la présence de toutes les dépendances, et écrit le fichier `makefile` qui contient les ordres de compilation), `make`, et enfin `make install` (elle installe le logiciel).

### 139)update-rc.d

`update-rc.d` permet de configurer le démarrage ou l'arrêt automatique de service au démarrage de la machine ou selon le runlevel. On donne en argument le nom du service et l'action (`remove` ou `default` pour l'ajout) `update-rc.d -f apache2 remove`; `-f` permet de forcer l'effacement du lien symbolique même si le nom existe encore. On peut aussi placer un script de démarrage dans répertoire `/etc/init.d` ou renseigner le fichier `/etc/rc.local` (qui a lui-même un lien symbolique dans `/etc/init.d`).

### 140)/etc/passwd

fichier qui contient les différents comptes utilisateurs de la machine (ce qui inclut les comptes utilisés par les logiciels ex. : `www-data` pour Apache). On peut afficher les informations avec `cat` : `cat /etc/passwd`.

### 141)cat /etc/group

fichier qui contient les groupes utilisateurs de la machine (ce qui inclut les groupes utilisés par les logiciels ex. : `www-data` pour Apache). On peut afficher les informations avec `cat` : `cat /etc/group`.

### 142)which

localise une commande ex. : `which cat`     `/bin/cat`

### 143)whereis

localise un fichier binaire.

#### 144) nice

permet de changer la priorité d'un programme/ (processus) (Niceness)

nice -PRIORITY COMMAND

ex: info@debian:~\$ ps -lp

```
F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD
```

```
4 S 0 941 1 0 70 0 - 1713 poll_s ? 00:00:00 test
```

on change la priorité de test par la valeur 10

info@debian:~\$ nice -10 ./test

si on veut une valeur négative

info@debian:~\$ nice --10 ./test

#### 145) renice

permet de changer la priorité d'un processus même après son lancement

info@debian:~\$ ps -lp

```
F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD
```

```
4 S 0 941 1 0 70 20 - 1713 poll_s ? 00:00:00 ps
```

info@debian:~\$ renice 30 941

après changement de processus ayant comme PID=941 on a :

info@debian:~\$ ps -lp

```
F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD
```

```
4 S 0 941 1 0 70 30 - 1713 poll_s ? 00:00:00 ps
```

### Manipulation

1) Que se passe-t-il si vous envoyez à votre propre shell le signal 15 ? essayez avec les signaux 1, 2 et 9

2) Lancez la commande suivante avec nohup en tâche de fond:

ls -lR /fichier 2>&1

Notez le numéro de ce processus en tâche de fond. Déconnectez puis reconnectez. Quelle est la valeur de PPID? expliquez.

3) Lancez la commande de l'exercice précédent. Appelez ensuite la commande **wait**. Que se passe-t-il? Comment revenir au prompt de shell? envoyez au processus le signal 9

4) Relancez la commande de l'exercice 2 et mesurez sa durée avec time. Redirigez la sortie de la commande time dans un fichier et essayez d'interpréter les valeurs

5) Afficher les cinq derniers processus lancés (ceux avec les ID de processus les plus élevés) ordonnée, avec l'ID de processus justifié à droite dans le premier champ séparé par des espaces.

ps -A | tac | head -5 | cut -b 0-5

6) Supposons que vous avez un programme en cours d'exécution appelé "myprog", qui est un enfant du shell actuel. Vous souhaitez diminuer l'utilisation du processeur de ce programme. Laquelle des lignes de commande suivantes pouvez-vous utiliser pour que "myprog" génère davantage de ressources CPU?

A. nice +1 myprog

B. ps h -o pid -C myprog | xargs nice +1 -

C. `renice +1 -u `whoami` myprog`

D. `renice +1 -p `ps -a | grep myprog | cut -b 1-6``

7) Analyser correctement un fichier source C nécessite un analyseur complet (tel que celui intégré dans un compilateur C). Néanmoins, les expressions régulières peuvent être utilisées pour fournir une assez bonne description approximative de nombreuses constructions de programme. Laquelle des recherches suivantes localisera au moins la plupart des fonctions C qui acceptent un int comme premier argument et renverra un int.

A. `grep -E "int[ \t]+\w+[ \t]*\([ \t]*int" *.c`

B. `grep -E "^int\w+[A-Za-z_]+\w*\(\w*int" *.c`

C. `grep -E "int.\([ \t]+int.*\) " *.c`

D. `grep -E "int[ \t]+[A-Za-z_][ \t]+\([ \t]*int" *.c`

8) Sur la base du système de nommage des partitions de Linux, lequel des noms de périphérique suivants pointe vers des partitions "logiques" (en supposant que les partitions correspondantes existent sur le système en question)?

A. `/dev/sda3`

B. `/dev/fd0`

C. `/dev/hdb7`

D. `/dev/hda4`

E. `/dev/fd7`

F. `/dev/sdc11`

9) Laquelle des lignes de commande suivantes peut (éventuellement) être utilisée pour formater une partition?

Supposons que les partitions requises existent et que le partitionnement logique est utilisé sur chaque disque dur.

A. `mkfs -t msdos /dev/sda1`

B. `mkfs.ext2 /dev/null`

C. `mkfs -t ext2 /dev/hda4`

D. `mkfs --type=ext2 /dev/hdb7`

10) # jobs -l

[1] 5110 Running kedit &

[2]- 5382 Stopped (signal) pine

[3]+ 5457 Stopped (tty output) vi

Quelle est la commande qui permet de terminer l'application 'vi'?

A. `bg %3`

B. `kill -9 5457`

C. `term -i %3`

D. `fg 5457`

11) La commande qui permet de repairer une partition

A. `chkdsk`

B. `scandisk`

C. `fsck`

D. `fdisk`

12) Laquelle des lignes de commande suivantes produira un rapport sur l'espace disque total utilisé par l'utilisateur actuel?

A. `fsck ~`

B. `df ~/.`

C. `quota --used`

D. `du -hs ~`

13) Quelle commande permet de donner l'espace disponible de disque utilisé

A. `free`

B. `df`

C. `du`

D. `fdisk`

14) Que contient le fichier '/etc/fstab' du système

15) Quelle commande Linux peut être utilisée pour limiter le quota d'utilisation de l'espace disque d'un utilisateur particulier? Supposons pour cette question que les quotas sont activés pour le (s) système (s) de fichiers utilisé (s) sur le système en question.

- A. edquota
- B. setquota
- C. quotaon
- D. repquota

16) Un avantage des liens durs sur les liens symboliques est:

- A. Un lien dur peut s'étendre sur différents systèmes de fichiers
- B. Un lien dur ne devient pas déconnecté du fichier principal si le fichier est déplacé.
- C. Vous pouvez déterminer l'inode utilisé par un lien dur, mais pas pour un lien symbolique.
- D. Un lien dur vous permet de modifier les autorisations sur le fichier principal.

17) Selon le standard de hiérarchie du système de fichiers Linux, lequel des répertoires suivants constitue-t-il un emplacement approprié pour un utilisateur puisse installer une application partagée?

- A. /sbin
- B. /dev/user/bin
- C. /usr/local/bin
- D. /etc/bin

18) Parmi les lignes de commande Linux suivantes, laquelle peut être utilisée pour examiner les messages de démarrage du noyau (kernel) après le temps de bootage?

- A. dmesg | less
- B. less /proc/kmsg
- C. bootlog -v
- D. vi /var/log/messages

19) Comment formater un USB?

20) Lister tous les fichiers cachés du répertoire de connexion sauf les répertoires

21) Lister tous les fichiers ordinaires cachés du répertoire de connexion

22) 1. Ecrire une commande qui donne le nombre de fichiers du répertoire courant

Rep : % `ls -al | wc -l`

2. Ecrire une commande qui donne le nombre de répertoires du répertoire courant

Rep: % `ls -al | grep ^d | wc -l`

3. Ecrire une commande qui donne le nombre de fichiers ordinaires du répertoire courant

Rep : % `ls -al | grep ^- | wc -l`

4. refaire 1 et 2 pour un répertoire donné

Rep: % `ls -al rep | wc -l;`

5. Ecrire une commande qui permet d'afficher le nombre d'utilisateurs utilisant le « bash » comme shell par défaut

Rep: % `grep bash /etc/passwd | wc -l`

6. Ecrire une commande qui permet d'afficher les utilisateurs qui utilisent le « bash » comme shell par défaut.

Rep : % `grep bash /etc/passwd | cut -f 1 -d ":"`

7. En utilisant la commande find « lister » tous les fichiers qui ne sont ni des fichiers ordinaires ni des répertoires. Pour les tests, vous créez un répertoire nommé « rep », ensuite à l'aide de la commande « cat »

ou « touch », créez deux fichiers nommés « fiche1 » et « fiche2 », enfin vous créez deux liens symboliques nommés « lien\_fiche1 » vers le fichier « fiche1 » et « lien\_fiche2 » vers le fichier « fiche2 »

Rep: find . ! \( -type f -o type d \) -ls

8. A l'aide de la commande grep, lister tous les fichiers du répertoire /etc/X11 dont le nom Contient le caractère x.

Rep : ls /etc/X11 | grep x

9. Contient la lettre x ou X.

Rep : ls /etc/X11/ | grep "[x|X]" ou ls /etc/X11/ | grep "[x|X]"

10. Commence par la lettre X ou x et qui contient un ou plusieurs chiffres.

Rep1 : ls /etc/X11/ | grep "^[x|X].\*[0-9]"

Rep2 : ls /etc/X11/ | grep "[x|X]" | grep "[0-9]"

11. Commence par la lettre X ou x et qui finit par un chiffre.

Rep 1 : ls /etc/X11/ | grep "^[x|X].\*[0-9]\$" "

Rep 2 : ls /etc/X11/ | grep "[x|X]" | grep "[0-9]\$" "

12. Ecrire une commande qui donne la liste des fichiers du répertoire « /etc » qui commencent avec la lettre « p » et qui sont triés par type.

Rep : ls -ld /etc/p\* | sort

13. Ecrire une commande qui donne la liste des fichiers du répertoire « /etc » qui commencent avec la lettre « p » triés suivant le nombre de liens (ordre croissant et ordre décroissant).

Rep : Ordre croissant : ls -ld /etc/p\* | sort -n k2,2 ou ls -ld /etc/p\* | sort -k2n,2n

Ordre décroissant : ls -ld /etc/p\* | sort -n -r -k2,2 ou ls -ld /etc/p\* | sort -k2nr,2nr

23) Créer le fichier « resultat.txt » qui a le contenu suivant :

Nom	Résultat
-----	----------

-----

Omar	18
Ali	8
Fatima	15
Mohammed	18
Houcine	16
Zahra	5
Hassane	15

a. Afficher le contenu du fichier résultats sauf les deux lignes titres (les deux premières lignes).

Rep : tail -n +3

b. Trier le fichier par ordre alphabétique (les deux lignes titres ne figurent pas sur la sortie). Rep : tail -n +3 resultat.txt | sort

c. Trier le fichier par ordre croissants suivant le champ « Résultat » (les deux lignes titres ne figurent pas sur la sortie).

Rep :

tail -n +3 resultat.txt | sort -n -k2 ou tail -n +3 resultat.txt | sort -n -k2,2 ou

tail -n +3 resultat.txt | sort -k2n ou tail -n +3 resultat.txt | sort -k2n,2n

Remarque : pour la commande « sort » on doit rajouter l'option « -n », sinon le trie se fait par ordre ASCII, dans ce cas on a  $18 < 5 < 8$ .

d. Trier le fichier par ordre décroissants suivant le champ « Résultat » (Les deux lignes titres ne figurent pas sur la sortie).

Rep : tail -n +3 resultat.txt | sort -n -r -k2,2 ou tail -n +3 resultat.txt | sort -k2nr,2nr

e. Supposons maintenant que le contenu du fichier « resultat.txt » est comme suit : Nom Résultats

-----

Omar	18:10:12
Ali	8:19:0
Fatima	15:16:12
Mohammed	18:10:20
Houcine	16:0:10

Zahra 5:13:10

Hassane 15 :16 :12

f. Trier le fichier par ordre croissant suivant le premier résultat

Rep 1 :

On doit unifier les séparateurs des champs, en transformant tous les séparateurs de champs en « : ». - Si on suppose que les noms sont séparés des résultats par des tabulations alors on transforme les tabulations en « : » avec la commande : `tr -s "$\t" ":"` Pour faire le tri suivant le deuxième champ, on utilise la commande :

```
% tail -n +3 resultat.txt | tr -s "$\t" ":" | sort -n -k2,2 -t :
```

Si on suppose que les noms sont séparés des résultats par des espaces et par des tabulations alors on transforme, tous d'abord, les tabulations en espaces ensuite on transforme les espaces en « : » avec la commande : `tr "$\t" " " | tr -s " " ":"`

Pour faire le tri suivant le deuxième champ, on utilise la commande :

```
% tail -n +3 resultat.txt | tr "$\t" " " | tr -s " " ":" | sort -n -k2,2 -t :
```

Remarque : Dans ce cas Mohammed est affiché avant Omar

g. Trier le fichier par ordre croissants suivant le premier résultat, puis suivant le troisième. `% tail -n +3 resultat.txt | tr "$\t" " " | tr -s " " ":" | sort -n k2,2 -k4,4 -t :`

Dans ce cas « Omar » sera classé avant « Mohammed » f. Afficher avec classement

```
% tail -n +3 resultat.txt | tr "$\t" " " | tr -s " " ":" | sort -n k2,2 -k4,4 -t : | cat -n
```

```
% tail -n +3 resultat.txt | tr "$\t" " " | tr -s " " ":" | sort -n k2,2 -k4,4 -t : | cat -b
```

L'option « -b » identique à « -n » en ignorant les lignes vides.

24). Donner la commande qui permet d'afficher la liste de tous les utilisateurs.

Rep 1: `cut -d: -f1 /etc/passwd` ou `cut -d":" -f1 /etc/passwd`

25) on a 4 fichiers st1.txt st2.txt st3.txt st4.txt

a)afficher successivement le contenu des 4 fichiers

rep: `cat st*.txt`

b)la forme générale de ces fichiers est : mot étiquette lemma (le séparateur est une tabulation \t). extraire la colonne correspondant aux mots, puis en utilisant les commandes `sort(1)` (alphabétique et numérique) et `uniq(1)` (avec l'option -c) calculer la liste de fréquence des mots de ce fichier

```
rep: $ cat st*.txt | cut -f1 | sort | uniq -c
```

c) Modifier la commande précédente de manière à n'afficher que les 20 mots les plus fréquents.

```
rep: $ cat st*.txt | cut -f1 | sort -f | uniq -c | sort -gr | head -20
```

d) Modifier la commande précédente de manière à afficher les 30 lemmes les plus fréquentes.

```
$ cat st*.txt | cut -f3 | sort -f | ? uniq -c | sort -gr -k 1 | head -30
```

e) Modifier la même commande de manière à afficher les 5 étiquettes les plus fréquentes.

```
$ cat st*.txt | cut -f2 | sort -f | ? uniq -c | sort -gr -k 1 | head -5
```

26) soit le fichier public\_doc.html.

a) Y a-t-il des caractères en gras dans ce document ?

```
rep: $ egrep -o "<b>.+</b>" public_doc.html
```

b)Obtenez la liste des balises HTML de ce document.

```
$ egrep -o "<[:alnum:]]+>" public_doc.html | sort | uniq
```

c)Quelle commande taper pour obtenir la liste des URLs se trouvant dans cette page ?

```
$ grep "<a href=" public_doc.html
```

## Web

Un virtual host est un fichier de configuration d'apache 2 qui lui dira que s'il reçoit une requête d'un domaine donné (par exemple : `www.monsite.ma`) il devra utiliser un dossier précis, avec une configuration particulière.

En plus d'avoir une url plus propre que `http://localhost/monsite.ma`, un vhost vous permettra d'activer des configurations avancées, souvent nécessaire lors de l'installation d'un symfony 2 par exemple

La première étape est de dire à notre machine que le domaine `monsite.ma` doit pointer sur notre propre machine (c'est à dire `127.0.0.1`) pour cela, deux possibilités, soit vous modifier votre fichier `hosts`, soit vous redirigez automatiquement tous les domaines en `.dev` vers votre machine.

Modification du fichier `hosts`

```
sudo gedit /etc/hosts
```

et ajoutez à la fin du fichier :

127.0.0.1 monsite.ma www.monsite.ma

Nous allons commencer par créer le dossier qui contiendra tous nos sources de sites. On peut créer des sous-dossiers dans /var/www/html, mais cela pourrait porter à confusion. Mais on peut utiliser le dossier de base /var/vhosts

Créons ce dossier à l'aide de la commande mkdir

sudo mkdir /var/vhosts

#### **146)a2ensite**

active un vhost Apache : a2ensite monsite.ma.

#### **147)a2dissite**

désactive un vhost Apache : a2dissite monsite.ma.

#### **148)a2enmod**

active un module Apache : a2enmod rewrite.

#### **149)a2dismod**

désactive un module Apache : a2dismod rewrite.

#### **150)a2enconf**

sur le même modèle que pour les vhost et les modules, a2enconf et a2disconf permettent d'activer ou de désactiver des configurations. Par exemple, lorsqu'on ne renseigne pas explicitement le fichier d'accès.log d'un vhost, Apache logue le tout dans un fichier dédié : other\_vhosts\_access.log. Dans le cas où l'on ne voudrait pas de logs d'accès par exemple, on peut désactiver cette configuration : a2disconf other-vhostsaccess-log.

#### **151)apache2ctl**

apache2ctl permet d'une part d'agir en tant que script init (ce qui n'a pas grand intérêt puisqu'on utilise en général la commande service). Et d'autre part de contrôler le processus Apache et de récupérer des infos sur ce dernier. apache2ctl -S est assez utile puisqu'elle permet de voir comment Apache interprète nos Vhost, l'option -M liste les modules Apache activés. Enfin, l'option -help vous en dira plus sur les autres commandes disponibles !

#### **152)curl**

curl permet, comme wget, de récupérer un fichier depuis une URL, mais s'il mérite sa place ici, c'est qu'il permet bien plus. C'est en effet le couteau suisse du HTTP. On l'utilisera en effet pour tester des requêtes dans différents formats, analyser les HEADERS, etc.

ex:

# requête get classique

curl https://monsite.ma

# afficher les headers

curl https://buzut.fr -D -

# faire une requête de type X (HEAD, POST, PUT, PATCH, DELETE...)

curl -X HEAD https://monsite.ma

# passer des paramètres au format form data

curl -X POST --data "email=moi@mail.com&passwd=azerty" https://monsite.com/login

# même requête avec les paramètres en request payload, JSON. (on ajoute ici un header)

curl -H "Content-Type: application/json" -X POST --data '{"email":"moi@mail.com","passwd":"azerty"}' https://monsite.ma/login

### **Gestion réseau**

#### **153)iftop**

comme top, iftop sert à surveiller toutes les connexions réseau. Attention, iftop nécessite les privilèges root pour être lancé. Si vous n'êtes pas root, le faire précéder de sudo.

#### **154)speedometer**

un peu plus graphique que iftop, speedometer surveille le trafic de vos entrées/sorties, permet de surveiller la progression d'un téléchargement, de savoir combien de temps il faudra pour transférer tel fichier ou encore de connaître la vitesse d'écriture de votre système.

#### **155)exim4**

Exim est un MTA (Mail Transfer Agent) qui permet d'envoyer des e-mails depuis le serveur. Sans lui (ou un autre MTA) la fonction mail() de PHP ou d'autres langages ne sera pas effective. Il en existe d'autres, mais celui-ci est robuste, sécurisé, modulable et demande peu de ressources.



### 156)ifconfig

cette commande des plus indispensables permet d'obtenir des infos et de configurer les interfaces réseau. Employée sans argument, elle fournit des infos sur les interfaces réseau. Mais elle permet aussi de modifier la configuration. Par exemple pour changer une adresse mac, on utilisera la commande  
ifconfig \$INTERFACE ether \$MAC. Donc pour changer l'adresse mac de la carte Ethernet, ce sera en général ifconfig eth0 ether5E:FF:56:A2:AF:15

Notez cependant que cette commande, bien que toujours fonctionnelle, est dépréciée en faveur de la commande ip ci-dessous.

### 157)ip

cette commande permet d'afficher et de manipuler le routage et les interfaces. On s'en sert souvent pour lier ou supprimer une ip à une interface :

# lister toutes les adresses

ip addr

# ne lister que les informations ipv6

ip -6 addr

# lister une interface en particulier

ip addr show dev em2

# ajouter une adresse ipv4

ip addr add 192.168.0.7 dev eth0

# supprimer une ipv4

ip addr del 192.168.0.7 dev eth0

# pour la v6, il faut utiliser l'argument -6

ip -6 addr del 2574:104::b08b:c107 dev eth0

### 158)ping

permet de pinguer un client pour voir s'il est en ligne ou s'il répond au ping, ex. ping google.com. L'option -c permet de préciser le nombre de ping à envoyer avant que la commande ne s'arrête (elle prend donc en argument un nombre entier exemple : ping -c 8 google.com), l'option -f permet de flood, c'est-à-dire que la carte réseau enverra autant de ping qu'elle est capable d'envoyer par seconde.

### 159)traceroute

trace la route d'un paquet, routeur par routeur, jusqu'à sa destination. On peut utiliser indifféremment une ip ou un nom de domaine.

### 160)host

permet de vérifier le reverse. Vous entrez donc une adresse ip et la commande vous retourne le nom de domaine associé.

### 161)nslookup

un peu dans le même genre que dig, nslookup fournit des infos sur le nom de domaine passé en argument, adresse ip, type de réponse DSN... Exemple : nslookup monsite.ma.

### 162) route

cette commande d'une praticité sans pareil vous permet de visionner les routes, mais aussi, en lui spécifiant quelques arguments, de modifier les routes. Pour ajouter une route par défaut : route add default addr ip et pour supprimer une route par défaut : route delete default.

### 163)arp -an

cette commande permet de voir la table arp actuelle. C'est-à-dire la correspondance entre les adresses ip et mac sur votre réseau.

### 164)nmap

nmap est un outil qui scanne le réseau et les ports réseau d'une machine afin de voir lesquels sont ouverts et de détecter d'éventuelles failles sur les machines. Il permet aussi de détecter des machines connectées au réseau et bien plus encore

### 165) netstat

cette commande est très pratique pour avoir un aperçu de ce qui se passe sur le réseau. On va donc voir directement quelques combinaisons de paramètres qui reviennent souvent:

-nr

pour la table de routage (revient au même que route).

-i

donne des statistiques sur les différentes interfaces réseau.

- s  
c'est un résumé de toutes les stats réseau, alors ça peut être utile de temps à autre.
  - uta  
liste toutes les connexions ouvertes (u pour udp t pour tcp et a pour afficher toutes les connexions (all)),
  - nap  
alternative à -uta, cette commande donne un résultat plus lisible. Elle affiche toutes les connexions (a) avec les adresses en format numérique (n) et donne le PID correspondant (p).
  - peanuts  
Elle donnera un résultat un peu plus détaillé que -nap puisqu'elle ajoute UDP et TCP avec du détail (-e pour extended) et des statistiques (-s) pour toutes les connexions.
  - ltupn  
encore une variante qui nous permet de visualiser tous les services qui sont à l'écoute sur des ports (donc toutes les connexions ouvertes). Toutes les connexions tcp -t et udp -u, à l'écoute -l avec les programmes correspondants -p et accessoirement,
  - l  
permet d'afficher toutes les connexions en écoute (listen), on aura donc netstat -lt pour toutes les connexions tcp en écoute.
  - n  
peut être pratique aussi, car elle permet d'afficher les adresses en format numérique au lieu de tenter de déterminer le nom symbolique d'hôte, de port ou d'utilisateur.
  - p  
permet d'afficher le nom et le PID des processus propriétaires des connexions.
- netstat est la commande historique pour lister les connexions. Cependant, elle a été dépréciée en faveur de ss qui est plus moderne et plus performante, surtout en présence de nombreuses connexions.

## Manipulation

### 1) Interface réseau

- a) Se connecter en tant que root sur une console texte.
- b) Afficher la liste des interfaces réseau.
- c) Noter l'adresse IP, le masque réseau et la passerelle de votre système.
- d) Noter le ou les DNS configurés sur votre système.
- e) Après l'avoir sauvegardé, éditer le fichier de configuration de votre réseau pour configurer l'adresse trouvée ci-dessus en adresse IP fixe et redémarrer la machine.
- f) Pouvez-vous toujours « pinger » la passerelle ? L'extérieur ?
- g) Pouvez-vous résoudre les noms ?

Indication La commande **ifconfig -a** permet de lister l'ensemble des interfaces réseau présentes sur le système.

### 2) DNS

- a) Trouver l'adresse du serveur www.google.com d'au moins deux manières différentes.
- b) Quel est le serveur DNS qui répond ?
- c) Quel est le nom DNS du serveur ayant l'adresse 212.27.48.10 ?
- d) Comment obtenir cette même réponse depuis le serveur DNS public de Google ?
- e) Quels sont les serveurs DNS du domaine yahoo.fr ?
- f) Quelles sont leurs adresses IP ?
- g) Quels sont les noms DNS des serveurs de messagerie du domaine free.fr ?
- h) Quelles sont leurs adresses IP ?
- i) Modifier votre serveur DNS pour utiliser le serveur openDNS.
- j) Vérifier les informations précédentes.

Indication: Pour obtenir l'adresse IP de www.google .com, on peut utiliser les commandes : **ping**, **nslookup**, **host** ou **dig**.

### 3) openSSH

- a) S'il n'est pas déjà présent, installer le package openssh-server.
- b) Changer le port d'écoute du serveur SSH en 2222 et interdire les connexions en tant que root.
- c) Démarrer ou redémarrer le service SSH pour que les modifications soient prises en compte.
- d) Se connecter en SSH sur localhost en tant que root, puis en tant que neo. Que se passe-t-il ?
- e) Générer une clé d'authentification au format RSA pour l'utilisateur root.

f) Copier la clé générée ci-dessus dans `/home/neo/.ssh/authorized_keys`.  
g) Se connecter en SSH en tant que neo sur localhost depuis une session root, puis depuis une session neo. Que se passe-t-il ? Pourquoi ?

Indication: Pour Debian :

```
[root]# aptitude search openssh-server
```

i A openssh-server - serveur shell sécurisé (SSH), pour accéder à des machines à distance

Si nécessaire :

```
[root]# aptitude install openssh-server
```

...

2.

Modifier le fichier `sshd-config` comme suit :

```
[root]# vi /etc/ssh/sshd_config
```

...

Port 2222

PermitRootLogin no

4) Apache (httpd)

a) S'il n'est pas déjà présent, installer le package httpd (Fedora) ou apache2 (Debian).

b) Démarrer le serveur si nécessaire et s'y connecter à l'aide d'un navigateur.

c) En s'inspirant des fichiers présents dans le répertoire `/etc/httpd/conf.d/` (Fedora) ou `/etc/apache2/conf-available` (Debian), créer le fichier `docpam.conf` pour ajouter un alias /PAM pointant vers la documentation de PAM.

d) Redémarrer le serveur Apache et se connecter à l'URL `http://votreip/PAM`, puis à l'URL `http://votre-IP/PAM/Linux-PAM_SAG.html`. Que se passe-t-il ?

e) Ajouter la directive `DirectoryIndex Linux-PAM_SAG.html` portant sur le chemin `/usr/share/doc/pam/html` et redémarrer Apache. Que se passe-t-il ?

Pour Debian :

```
[root]# aptitude search apache2
```

p apache2 - Serveur HTTP Apache

Installer au moins le paquet `libpam-doc` pour la suite.

```
[root]# aptitude install apache2 libpam-doc
```

...

5) Serveur NFS

a) Créer le répertoire `/export/distrib` pour accueillir une copie du DVD-ROM de la distribution Fedora ou Debian.

b) Copier l'intégralité du DVD-ROM de la distribution dans le répertoire `/export/distrib`.

c) Rendre le répertoire accessible via NFS à toutes les machines du réseau.

d) Relancer le service NFS pour prendre en compte les modifications du fichier de configuration et vérifier la présence de ce nouveau répertoire dans les partages réseau NFS actifs.

f) Monter le partage NFS sur `/mnt/distrib` et vérifier que l'accès est en lecture seule.

Indication

```
[root]# mkdir /export
```

```
[root]# mkdir /export/distrib
```

ou :

```
[root]# mkdir -p /export/distrib
```

2.

Pour copier le DVD-ROM de la distribution, effectuer les opérations suivantes :

```
[root]# mount /dev/cdrom /mnt
```

```
[root]# cp -dpr /mnt/* /export/distrib
```

```
[root]# umount /mnt
```

3.

```
[root]# vi /etc/exports
```

et ajouter la ligne suivante :

```
/export/distrib *(ro)
```

4.

```
[root]# systemctl restart nfs-server.service
```

```
[root]# systemctl status nfs-server.service
```

- nfs-server.service - NFS server and services  
 Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; disabled; vendor  
 Active: active (exited) since mar. 2016-12-27 17:08:46 CET; 33s ago  
 Process: 4549 ExecStart=/usr/sbin/rpc.nfsd \$RPCNFSARGS (code=exited, status=0  
 Process: 4545 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS  
 Main PID: 4549 (code=exited, status=0/SUCCESS)  
 Tasks: 0 (limit: 4915)  
 CGroup: /system.slice/nfs-server.service

déc. 27 17:08:46 localhost.localdomain systemd[1]: Starting NFS server and servi

déc. 27 17:08:46 localhost.localdomain systemd[1]: Started NFS server and servic

Ou :

[root]# service nfs restart

puis, pour vérifier que le répertoire est bien exporté via NFS :

[root]# exportfs

/export/distrib

<world>

5.

On utilise localhost comme serveur.

[root]# mkdir -p /mnt/distrib

[root]# mount ...

### Travail complémentaire

#### CONFIGURATION RESEAU STATIQUE

Objectif : acquérir l'adresse nécessaire à la configuration manuelle d'un réseau.

Configuration de départ : un système Linux qui utilise une configuration réseau dhcp.

Mise en situation : le serveur dhcp est en panne ! Vous devez mettre votre station de travail sur le réseau afin de modifier manuellement les fichiers de configuration appropriés et définir une configuration réseau statique.

Sequence 1 : définition de l'adresse IP

Mise en situation :

Le serveur dhcp sur votre réseau est en panne. Vous devez alors définir une adresse IP statique afin de pouvoir remettre votre station de travail sur le réseau.

Tâches :

1. Premièrement, désactivez votre interface Ethernet à l'aide de la commande ifdown :

ifdown eth0

2. Ouvrez ensuite le fichier /etc/sysconfig/network-scripts/ifcfg-eth0 dans un éditeur de texte et modifiez son contenu de la façon suivante (où X doit être remplacé par un numéro de votre réseau) :

```
DEVICE=eth0 BOOTPROTO=none ONBOOT=yes IPADDR=192.168.0.X NETMASK=255.255.255.0
GETEWAY=192.168.0.254
```

3. Visualisez le contenu du fichier /etc/resolv.conf. Vous devriez y retrouver les réglages valides obtenus du serveur dhcp. Si ce n'est pas le cas, faites les modifications nécessaires afin que le tout corresponde à ce qui suit :

```
search exemple.com nameserver 192.168.0.254
```

4. Activez l'interface que vous venez de configurer à l'aide de la commande ifup :

ifup eth0

5. Vérifiez vos réglages réseau en appliquant la commande ping à server1

6. Redémarrez votre ordinateur et vérifiez de nouveau vos réglages réseau en appliquant la commande ping à server1.

Résultat :

Un système configuré de façon à pouvoir fonctionner avec des réglages de réseau statique.

Remise à l'état initial :

Une fois que votre instructeur aura réactivé le serveur dhcp, remettez vos fichiers de configuration à leur état initial, désactivez eth0, puis réactivez-le. Le fichier ifcfg-eth0 devrait alors ressembler à ceci :

```
DEVICE=eth0
BOOTPROTO=dhcp
ONBOOT=yes
```

Travaux pratiques : installation d'un serveur DNS

## PARTIE 1 : INSTALLATION DU SERVEUR

Pour récupérer les adresses IP, on lance un terminal sur chaque une des deux machines et on tape la commande ifconfig, on regarde la ligne "inet adr". On trouve par exemple pour le client : 192.168.108.2 et pour le serveur DNS 192.168.108.1.

Sur la machine serveur on lance le serveur dns en utilisant la commande suivante :  
/etc/init.d/named start

Sur le serveur DNS :

Dans le fichier /etc/named.conf, on ajoute les lignes suivantes :

```
Domaine.com
zone "domaine.com" { type master;
file domaine.com.zone";
};
```

domaine.com est le nom de domaine pour lequel le serveur sera primaire (c'est à dire où c'est ce serveur DNS qui sera utilisé), domaine.com.zone désigne le fichier où seront stockés les enregistrements de la zone.

On crée le fichier domaine.com.zone et on ajoute les lignes suivantes :

```
$TTL 604800
domaine.com. IN SOA machine1 root.machine1.domaine.com. (
1 ; Serial 604800 ; Refresh 86400 ; Retry 2419200 ; Expire
604800 ) ; Negative Cache TTL
;
domaine.com. IN NS machine1 machine1 IN A 192.168.108.1. machine2 IN A 192.168.108.2.
```

Pour la résolution inverse, dans le fichier named.conf, on rajoute :

```
zone "108.168.192.in-addr.arpa" { type master;
file "Domaine.com.rev";
};
```

Dans le fichier domaine.com.rev, on tape :

```
$TTL 604800
;$ORIGIN 108.168.192.in-addr.arpa.
@ IN SOA machine1 root.machine1.domaine.com (
1 ; Serial 604800 ; Refresh 86400 ; Retry 2419200 ; Expire
604800 ) ; Negative Cache TTL
;
IN NS machine1.domaine.com.
;
1      IN A machine1
2      IN A machine2
```

Une fois les modifications faites, on tape /etc/init.d/named restart

Le serveur se relance donc et enregistre les modifications.

Pour voir si le fichier named.conf ne comprend pas d'erreur, on tape la commande :

tail /var/log/messages et on repère les lignes des fichiers dans lesquelles il y a des erreurs. Pour faire l'essai de la résolution directe, on tape sur le serveur la commande :

```
dig @localhost machine1.Domaine.com A
```

Pour tester la résolution inverse on tape :

```
dig @localhost -x 192.168.108.2
```

En résultat nous obtenons :

Avec la première commande nous avons la correspondance IP du nom alors qu'avec la deuxième commande nous avons la correspondance du nom par rapport à l'IP.

Cela atteste que la résolution directe et la résolution inverse fonctionnent.

Pour rajouter des ressources de types CNAME, on rajoute dans le fichier domaine.com.zone la ligne :

```
www 86400 IN CNAME machine1.Domaine.com.
```

Pour rajouter des ressources de types MX, on rajoute dans le fichier domaine.com.rev la ligne:

```
@ IN MX 5 machine1.domaine.com.
```

## CONFIGURATION D'UN CLIENT SOUS LINUX

Par défaut le fichier /etc/resolv.conf doit contenir l'adresse du serveur de noms que l'on possède et le nom de notre domaine. On peut spécifier plusieurs serveurs de noms grâce à la commande nameserver, on peut également positionner notre nom de machine à l'aide la commande hostname.

Pour que la commande ping réussisse il faut indiquer le serveur DNS que l'on veut utiliser, ici on met la ligne :  
nameserver 192.168.108.1

On teste le ping :

ping machine1.domaine.com, la commande marche le fichier /etc/resolv.conf est donc bien configuré.

En rajoutant une ressource de type CNAME comme nous l'avons vu au dessus, lorsque l'on tape la commande ping www.domaine.com, il y a une réponse de la machine, l'alias fonctionne.

Lorsque l'on lance une capture avec ethereal pendant que les pings sont effectués, il y a au niveau des trames : une demande dns du client vis à vis du serveur pour lui demander l'adresse de www.domaine.com, le DNS répond que c'est un alias de machine1 et il donne l'adresse IP de machine1 ensuite le client contacte par un ping l'IP de machine1.

## PARTIE 2 MISE EN PLACE D'UN SERVEUR SECONDAIRE

Sur le serveur secondaire dans le fichier named.conf, on écrit :

```
zone "domaine.com" {
```

```
type slave; //on indique que c'est un serveur secondaire
```

```
masters {192.168.108.1}; //on indique que le serveur primaire est 192.168.108.1
```

```
};
```

Si on modifie les fichiers de configurations sur le DNS principal, il faut incrémenter le numéro de série, ainsi lors d'une requête DNS en utilisant le serveur secondaire celui ci vérifie si le numéro de série est le même que celui présent dans les fichiers du DNS principal, si ce n'est pas le cas, il met ces fichiers à jours.

Travaux pratiques : installation d'un serveur DHCP

Indications pour la réalisation du TP

L'atelier propose

- d'installer un serveur DHCP sous Linux,
- d'installer un client DHCP sous Linux
- d'installer un client DHCP sous Windows
- de réaliser une phase de test avec les commandes winipcfg et ipconfig de Windows

Matériel nécessaire :

Deux machines en dual boot Linux / Windows en réseau.

Les éléments sur l'analyse de trame, notamment les trames bootp, seront retraités lors des TP sur la métrologie.

Installation du serveur

Les paquets sont déjà installés.

Attention : vous pouvez avoir sur votre distribution, plusieurs serveurs DHCP.

dhcpcd est conforme à la RFC 2131. Il fournit un exemple de configuration assez détaillé. dhcp3, intègre l'inscription auprès d'un DNS Dynamique. C'est ce package que nous allons utiliser dans le TP. Par contre si vous n'avez pas de DNS dynamique sur le réseau, vous devrez mettre en entête du fichier dhcpd.conf, la ligne :  
ddns-update-style none;

#### Configuration du serveur

La configuration consiste à créer 2 fichiers :

- /etc/dhcp/dhcpd.conf, ce fichier sert à la configuration même du serveur (plage d'adresses, paramètres distribués),
- /var/lib/dhcpd/dhcpd.leases, ce fichier va servir à l'inscription des clients. Chaque client DHCP, génère l'écriture d'un enregistrement dans ce fichier. Cela permet le suivi, les statistiques de l'activité du serveur.

#### Le fichier de configuration dhcpd.conf

Vous trouverez un exemple de fichier commenté qui permet de réaliser cet atelier dans la documentation de dhcpd sous /usr/share/doc/dhcp\*. Vous pouvez créer ce fichier avec un éditeur.

```
$>cat dhcpd.conf

# ici il s'agit du réseau 192.168.0.0
subnet 192.168.0.0 netmask 255.255.255.0 {

#La plage d'adresse disponible pour les clients
range 192.168.0.10 192.168.0.20;

# Les clients auront cette adresse comme passerelle par défaut
option routers      192.168.0.254;

# Ici c'est le serveur de noms, on peut en mettre plusieurs
option domain-name-servers  192.168.0.1;

# Enfin on leur donne le nom du domaine
option domain-name      "freeduc-sup.org";

# Et l'adresse utilisée pour la diffusion
option broadcast-address 192.168.0.255;

    default-lease-time  86400;

    # On le laisse avec un maximum de 7 jours
    max-lease-time 604800;

    #Ici on désire réserver des adresses à des machines
    group {

#use-host-decl-names indique que toutes les machines dans l'instruction « group »
# auront comme nom, celui déclaré dans l'instruction host.

use-host-decl-names true ;

    # ici définir les machines
    host m1 {

hardware ethernet 00:80:23:a8:a7:24;
fixed-address 192.168.0.125;

    } # End m1
```

Le fichier `/var/lib/dhcp3/dhcpd.leases` peut contenir après l'inscription du premier client :

```
[root@master /etc]# more /var/lib/dhcpd/dhcpd.leases
lease 192.168.0.10 {

    starts 1 2002/12/14 18:33:45;

    ends 1 2002/12/14 18:34:22;

    hardware ethernet 00:40:33:2d:b5:dd;

    uid 01:00:40:33:2d:b5:dd;
```

On distingue les informations suivantes : Début du bail, Fin du bail, adresse MAC du client, le nom d'hôte du client. Attention ce nom est différent du nom Netbios utilisé sur les réseaux Microsoft.

Activation du serveur

Le serveur est configuré, il n'y a plus qu'à le mettre en route. Utilisez la commande suivante pour arrêter ou activer le service : `/etc/init.d/dhcpd start | stop`.

Le script lance le serveur en mode daemon.

### Réalisation du TP

1. Installez un serveur DHCP minimal sous Linux et vérifiez le bon démarrage du service
2. Installez un client DHCP sous Linux, vérifiez le bon démarrage du service réseau et l'inscription dans le fichier *dhcpd.leases* du serveur. Testez le renouvellement du bail. Il suffit de relancer le service réseau.
3. Installez un client DHCP sous Windows, vérifiez le bon démarrage du service réseau et l'inscription dans le fichier *dhcpd.leases* du serveur. Testez le renouvellement du bail.
4. Modifiez l'étendue du serveur. Vérifiez le bon fonctionnement de la distribution d'adresses aux clients.
5. Modifiez la configuration du serveur afin qu'il distribue également l'adresse de la passerelle par défaut, l'adresse du serveur de nom. Testez la configuration.
6. Modifiez la configuration du serveur DHCP afin de réserver une adresse au client, vérifiez que le processus a bien fonctionné.



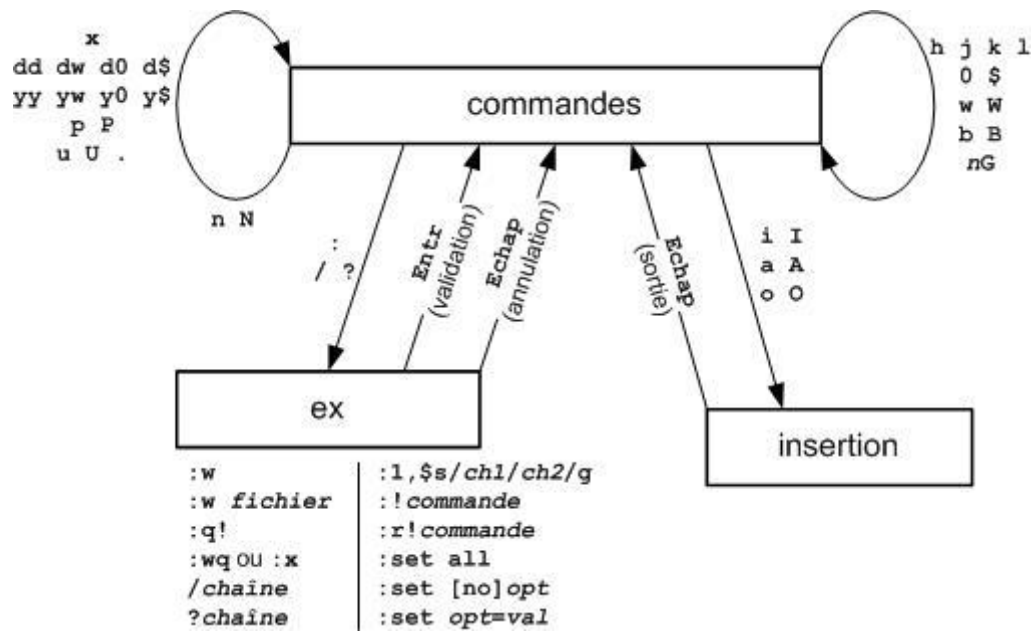
Références :

- 1) <https://buzut.developpez.com/tutoriels/101-commandes-indispensables-sous-linux/> (2019)
- 2) <https://www.scribd.com/document/379922300/TPs-Admin-Serv-Res-Linux> (2019)
- 3) ABRAHAM SILBERSCHATZ et.al "OPERATING SYSTEM CONCEPTS" JOHN WILEY & SONS. INC (2016)
- 4) Wang, Paul S., " Mastering modern Linux ", Taylor & Francis, CRC Press (2018)
- 5) [https://doc.lagout.org/LPI/Sybex\\_lpic1\\_3e-2013.pdf](https://doc.lagout.org/LPI/Sybex_lpic1_3e-2013.pdf) (2019)
- 6) [https://www.academia.edu/33309551/CompTIA\\_Linux\\_LPIC\\_1\\_Cert\\_Guide\\_Exams\\_LX0\\_103\\_and\\_LX0\\_104101\\_400\\_and\\_102\\_400\\_](https://www.academia.edu/33309551/CompTIA_Linux_LPIC_1_Cert_Guide_Exams_LX0_103_and_LX0_104101_400_and_102_400_)

30)vi

éditeur de texte en mode plein écran

### Aide-mémoire Vi



source: <http://www.epons.org/vi.php>

## DÉPLACEMENT DU CURSEUR

<b>h</b> vers la gauche	<b>0</b> début de ligne	<b>\$</b> fin de ligne
<b>j</b> vers le bas	<b>b</b> début de mot précédent	<b>w</b> début de mot suivant
<b>k</b> vers le haut	<b>&lt;Ctrl&gt;b</b> page précédente	<b>&lt;Ctrl&gt;f</b> page suivante
<b>l</b> vers la droite	<b>G</b> dernière ligne	<b>nG</b> ligne <i>n</i>

## INSERTION

<b>i</b> insère avant le curseur	<b>a</b> insère après le curseur	<b>o</b> insère une ligne après le curseur
<b>I</b> insère en début de ligne	<b>A</b> insère en fin de ligne	<b>O</b> insère une ligne avant le curseur

## ÉDITION

<b>dd</b> coupe la ligne	<b>yy</b> copie la ligne	<b>x</b> supprime 1 caractère
<b>dw</b> coupe jusqu'au mot suivant	<b>yw</b> copie jusqu'au mot suivant	<b>p</b> colle après le curseur
<b>d0</b> coupe jusqu'au début de ligne	<b>y0</b> copie jusqu'au début de ligne	<b>P</b> colle avant le curseur
<b>d\$</b> coupe jusqu'à la fin de ligne	<b>y\$</b> copie jusqu'à la fin de ligne	<b>.</b> répète la dernière commande
		<b>u</b> annule la dernière commande
		<b>U</b> restaure la ligne modifiée

## ENREGISTREMENT/SORTIE

<b>:w</b> enregistre	<b>:q</b> sort	<b>:wq</b> enregistre et sort
<b>:w fichier</b> enregistre sous <i>fichier</i>	<b>:q!</b> sort sans enregistrer	

## RECHERCHE

<b>/chaîne</b> recherche <i>chaîne</i> en descendant	<b>n</b> occurrence suivante
<b>?chaîne</b> recherche <i>chaîne</i> en montant	<b>N</b> occurrence précédente

## OPTIONS

<b>:set all</b> affiche toutes les options	<b>:set number</b> numérote les lignes	<b>:set tabstop=n</b> une tabulation = <i>n</i> espaces
---	---	--

## DIVERS

<b>:1,\$s/chaîne/CHAINE/g</b> remplace <i>chaîne</i> par <i>CHAINE</i> partout	<b>:! cmd</b> exécute la commande <i>cmd</i>	<b>:r! cmd</b> insère le résultat de la commande <i>cmd</i>
---	---	--