



Système d'exploitation Linux (Debian)



Pré requis

Matériel nécessaire :

- Laboratoire DIDAVDI+
- Station étudiante complète

Durée : 8 heures

Sommaire

1	Présentation Debian.....	5
2	Debian en mode graphique.....	7
2.1.1	Introduction	7
2.1.2	Gestionnaire de Fichiers	8
2.1.3	Configuration réseau.....	10
2.1.4	Terminaux ligne de commande	10
3	Debian en mode texte (shell).....	11
3.1	Le mode texte.....	12
3.1.1	Qu'est ce qu'un shell ?.....	12
3.1.2	La console.....	12
3.1.3	Application (TD)	14
3.2	L'aide sous Linux (nam).....	15
3.2.1	Introduction	15
3.2.2	Ouvrir le man.....	15
3.2.3	Application (TD)	15
3.3	Droits et Propriétaires	16
3.3.1	Introduction	16
3.3.2	Exemple.....	17
3.3.3	Modifications des droits	17
3.3.4	Modification de l'utilisateur et/ou du groupe	19
3.3.5	Application (TP).....	20
3.4	Commandes utiles.....	21
3.4.1	Les répertoires	21
3.4.2	Les principales commandes système	22
3.4.3	Autres commandes système.....	24
3.4.4	Commandes pratiques.....	25
3.4.5	Manipulation de fichiers.....	26
3.4.6	Edition de Fichiers.....	28
3.4.7	Application (TD)	30

4	Configuration du réseau	31
4.1	Configuration carte réseau	31
4.1.1	Mode graphique	31
4.1.2	Ligne de commande.....	32
4.2	Commandes de bases	34
4.2.1	Ping	34
4.2.2	Nmap	34
4.2.3	Traceroute	34
5	Installation de paquets	35
5.1	Les paquets.....	35
5.2	Les dépendances	35
5.3	Les dépôts	36
5.3.1	La notion des dépôts	36
5.3.2	Gérer ses dépôts.....	37
5.4	Télécharger des paquets.....	37
5.4.1	Mise à jour des paquets	38
5.4.2	Recherche du paquet.....	39
5.4.3	Installation	40
5.4.4	Suppression de paquet	41
5.4.5	Alternative à apt-get	42

1 Présentation Debian

Debian est une distribution Linux lancée en 1993 par Ian Murdock avec le soutien de la FSF (Free Software Foundation). Ce projet a pour principal objectif de fournir un système d'exploitation entièrement composé de logiciels libres.

- Un système d'exploitation est la couche logicielle qui sert d'intermédiaire entre le matériel et les applications. Exemples : GNU/Linux, Microsoft Windows, MacOS...
- Un logiciel libre est un logiciel dont l'utilisation, l'étude, la modification, la duplication et la diffusion sont universellement autorisées sans contrepartie (y compris financière).

Aujourd'hui il existe plusieurs dizaines de distribution Linux.

Au sein du laboratoire Did@VDI, nous avons choisi la distribution Debian car elle fait partie des distributions les plus populaires.

Rappelons que d'autres distributions très populaires comme Ubuntu dérivent de Debian et sont donc par essence très proches.

Un autre gros avantage de Debian est le gestionnaire de paquets apt-get. C'est un programme qui gère tous les logiciels installés et qui vous permet de les désinstaller en un rien de temps. D'autre part, tous les logiciels sont centralisés en un même endroit, ce qui fait que vous n'avez pas à parcourir tout le Web pour retrouver un programme.

En fait, vous avez juste à indiquer le nom du logiciel que vous désirez : Debian ira le télécharger et l'installer pour vous. C'est extrêmement simple d'utilisation.

3.4 Commandes utiles

Après avoir fait quelques manipulations sur les droits et accès, nous allons voir comment manipuler les fichiers, et par conséquent les commandes nécessaires.

3.4.1 Les répertoires

3.4.1.1 Le répertoire courant

Toutes vos commandes sont exécutées dans le répertoire courant. Ce répertoire est indiqué dans le prompt : dans l'exemple suivant, la commande `ls` va travailler sur le répertoire `/usr/share`.

```
root@PCClientDidaVDI:~# ls
```

Lorsque vous connectez-vous sur un serveur, vous êtes connecté, le répertoire courant est fixé à votre répertoire personnel, c'est-à-dire le répertoire de l'utilisateur qui est abrégé par le symbole `~`. Vous pouvez changer de répertoire avec la commande `change directory` ou `cd`. Avec un argument, elle vous envoie dans le répertoire donné ; sans argument, elle vous ramène au répertoire personnel :

```
root@PCClientDidaVDI:~# cd /usr/share
root@PCClientDidaVDI:/usr/share# cd
root@PCClientDidaVDI:~#
```

3.4.1.2 Utiliser les répertoires

Lorsque vous souhaitez agir sur un fichier, vous devez indiquer l'endroit où il se trouve, son *chemin* — à la commande que vous utilisez pour cela. Il y a trois façons de désigner un fichier :

de façon absolue

en indiquant son chemin complet, en commençant par la racine `/`, par exemple :

```
/usr/share/doc/bash/copyright ;
```

de façon relative

en indiquant son chemin par rapport au répertoire courant, sans commencer par une `/`, par exemple `doc/bash/copyright`, si je me situe dans le répertoire `/usr/share`,

par rapport à votre répertoire personnel

en indiquant son chemin à partir de votre répertoire personnel, abrégé par le caractère `~`, par exemple :

```
~/bash_history
```

Par ailleurs, vous pouvez vous déplacer dans un répertoire en utilisant le répertoire spécial `..`, que l'on appelle *répertoire parent*. Ainsi, quand vous êtes dans votre répertoire personnel, vous pouvez passer dans le répertoire `/home/`, puis dans votre répertoire personnel ainsi :

```
~$ cd ..
/home$ cd ..
/$ cd
~$
```

3.4.2 Les principales commandes système

mount

Elle sert à intégrer un système de fichiers dans un autre. Elle doit être exécutée en tant que root. Sa syntaxe habituelle est :

```
# mount -o options /dev/périphérique /mnt
```

Il faut au préalable, vérifier le système de montage avec la commande « fdisk -l ».

```
On exécute la commande sans périphérique afin de repérer les systèmes existants.
root@PCClientDidaVDI:~# fdisk -l

Disque /dev/sda : 320.1 Go, 320072933376 octets
[...]

Périphérique Amorce Début Fin Blocs Id Système
/dev/sda1 * 2048 31252047 15625000 83 Linux
/dev/sda2 617134078 625141759 4003841 5 Étendue
/dev/sda5 617134080 625141759 4003840 82 partition d'échange Linux / Solaris

root@PCClientDidaVDI:~#

On connecte le périphérique, puis on ré-exécute la commande
root@PCClientDidaVDI:~# fdisk -l

Disque /dev/sda : 320.1 Go, 320072933376 octets
[...]

Périphérique Amorce Début Fin Blocs Id Système
/dev/sda1 * 2048 31252047 15625000 83 Linux
/dev/sda2 617134078 625141759 4003841 5 Étendue
/dev/sda5 617134080 625141759 4003840 82 partition d'échange Linux / Solaris

Disque /dev/sdb : 4009 Mo, 4009754624 octets
[...]

Périphérique Amorce Début Fin Blocs Id Système
/dev/sdb1 * 128 7831551 3125000 1 FAT32

root@PCClientDidaVDI:~#
```

On aperçoit alors un périphérique qui est non monté, la commande est alors :

```
# mount /dev/sdb1 /mnt
```

Pour démonter ce système de fichiers, il suffit de taper en root :

```
# umount /mnt
```

Nota : si vous avez plusieurs périphériques à brancher, il est conseillé de faire des sous répertoires par périphérique dans /mnt

Par contre, n'importe quel utilisateur peut taper la commande **mount** tout court pour savoir quels sont les systèmes de fichiers montés à l'instant d'exécution de la commande.

su

Cette commande sert à changer d'utilisateur, après avoir rentré le bon mot de passe, bien sûr !

su -

permet de devenir root.

su - toto

permet de devenir l'utilisateur *toto*.

ps

Cette commande sert à lister les processus et leurs propriétés. Sous Unix, chaque tâche est exécutée en un ou plusieurs processus. Chaque processus a un PID (Processus ID) qui lui est propre. Un processus peut avoir d'autres processus ne sont pas affectés. On peut tuer un processus avec la commande **kill**.

ps

liste les processus de l'utilisateur qui exécute la commande. Les processus sont rattachés au terminal depuis lequel la commande est exécutée.

ps -u

liste les processus de l'utilisateur qui exécute la commande, quel que soit le terminal de rattachement.

ps -au

liste les processus de tous les utilisateurs, quel que soit le terminal de rattachement.

ps -aux

liste les processus de tous les utilisateurs, même ceux qui ne sont rattachés à aucun terminal. Cette commande liste également les processus du système. Elle est équivalente à la commande **ps -A**.

ps -faux

liste tous les processus du système en les regroupant par enchaînement d'exécution.

kill et pkill

Les commandes **kill** et **killall** servent à envoyer des signaux à des processus.

kill 42

envoie le signal *TERM* au processus dont le PID est 42. En gros, on demande au processus 42 de se terminer tout seul. Bien sûr, on ne peut terminer que les processus que l'on a soi-même lancé, sauf le root qui peut faire ce qu'il veut avec tous les processus.

kill -9 42

envoie le signal *KILL* au processus dont le PID est 42. Quand un processus est bloqué, le seul moyen de l'arrêter, car la commande précédente n'aura pas d'effet.

pkill vlc

envoie le signal *TERM* au processus dont le nom est *vlc*. Cette commande sert à tuer plusieurs fois s'il y a plusieurs processus qui portent le nom *vlc*.

pkill -9 vlc

envoie le signal *KILL* au processus dont le nom est *vlc*.

3.4.3 Autres commandes système

passwd

change le mot de passe (il commence par demander l'ancien mot de passe, il s'agit d'un simple utilisateur).

groups

pour savoir à quels groupes appartient un utilisateur.

adduser +

ajoute l'utilisateur au système.

deluser toto

enlève l'utilisateur toto du système.

adduser toto disk

ajoute l'utilisateur *toto* au groupe *disk* (modification effective après que l'utilisateur *toto* se soit déconnecté puis reconnecté).

deluser toto audio

Enlève l'utilisateur *toto* du groupe *audio*.

df -h

fait le point sur l'espace libre de chaque système de fichiers monté.

du -sh

mesure la taille du répertoire depuis lequel il est exécuté.

uptime

Indique depuis combien de temps le système n'a pas redémarré. Certains s'amusent ainsi à faire des *concours d'uptime* pour prouver la stabilité de leur machine GNU/Linux !

w et who

Permet de savoir quels utilisateurs sont connectés sur le système et ce qu'ils font.

lspci, lsusb et lshw

donnent des informations sur les périphériques connectés à votre système (PCI, AGP, USB ou autres) : très pratique pour avoir des renseignements sur son matériel ! Attention, quand la commande affiche *hwid: unknown device*, c'est juste que l'identifiant du périphérique n'a pas de nom correspondant dans la base de données des commandes, mais cela ne veut pas dire que le périphérique marche, ne marche pas ou ne marche pas sur Linux !

cat /proc/cpuinfo

affiche des informations sur le processeur.

uname -a

donne des informations sur le système, notamment la version du noyau.

3.4.4 Commandes pratiques

date

donne l'heure, selon l'horloge de votre ordinateur ;

cal

affiche un calendrier du mois ; **cal** affiche le calendrier de l'année 2005 ;

bc

affiche une matrice en binaire ;

Ctrl+l

permet de rafraîchir l'affichage d'une application en console quand l'affichage est perturbé (par un message d'erreur par exemple) ;

Ctrl+s et Ctrl+q

permettent respectivement de bloquer et de débloquer l'affichage d'un terminal.

Ctrl+c

arrête un programme, ou annule une commande en cours de saisie

3.4.5 Manipulation de fichiers

3.4.5.1 Afficher un fichier

Pour afficher le contenu d'un fichier, il faut utiliser la commande `cat`. Cette commande permet juste de visualiser un fichier, mais ne permet pas la manipulation.

```
cat fichier_x.htm
```

Il est possible d'afficher des fichiers plus importants que `/var/log/syslog`. Cependant « `cat` » affiche tout le contenu d'un fichier en une seule fois.

Il peut être utile dans certains cas de lire les pages successivement (par exemple pour un manuel). Nous préférons utiliser la commande « `less` ».

Exemple :

```
less /var/log/syslog
```

Dans ce cas, il faut appuyer sur la touche « Espace » pour afficher la ligne suivante, « Enter » pour passer à la page suivante, ou sur « q » pour arrêter la lecture.

3.4.5.2 Créer / Supprimer un fichier ou un dossier

Pour créer :

- un fichier, il faut utiliser la commande « `touch` » ou « `nano` »

```
touch mon_fichier.txt
```

- un dossier, il faut utiliser la commande « `mkdir` »

```
mkdir mon_dossier
```

Afin de vérifier sa présence, nous pouvons exécuter la commande « `ls` » qui liste le contenu d'un répertoire.

Pour supprimer :

- un fichier, il faut utiliser la commande « `rm` »

```
rm mon_fichier.txt
```

- un dossier s'il est vide, la commande « `rm` » s'applique aussi

```
rm mon_dossier
```

Dans le cas contraire, si le dossier comprend des sous-dossiers ou fichiers, il sera nécessaire d'ajouter l'option « `-R` », soit « `rm -R mon_dossier` ».

3.4.5.3 Copier / déplacer un fichier

cp : copier un fichier ; cp source destination

La commande cp (abréviation de *CoPY*, « copier » en anglais, voir [l'article](#)) et dont le nom l'indique de copier un fichier... mais aussi de copier plusieurs fichiers, et même de copier des dossiers. Le nom du fichier original peut être changé lors de la copie.

Copie d'un fichier

```
cp /x.html mon_fichier
```

Copie d'un dossier et de son contenu (il faut l'option -R)

```
cp -R Images/ MesImages/
```

Il est bien sûr possible de copier des fichiers/dossiers depuis/vers un endroit différent.

Dans ce cas, il faut spécifier les chemins complets :

```
cp /var/log/syslog /home/utilisateur/www/MaCopieDeSyslog
```

mv : déplacer un fichier ; mv source destination

La commande mv (abréviation de *MoVe*, « déplacer » en anglais) vous permet de faire un couper / coller d'un fichier. Par conséquent, le fichier original n'existe plus.

```
mv MaCopieDeSyslog AncienSyslog
```

Exécutez la commande « ls -l » pour vous assurez des copies effectuées.

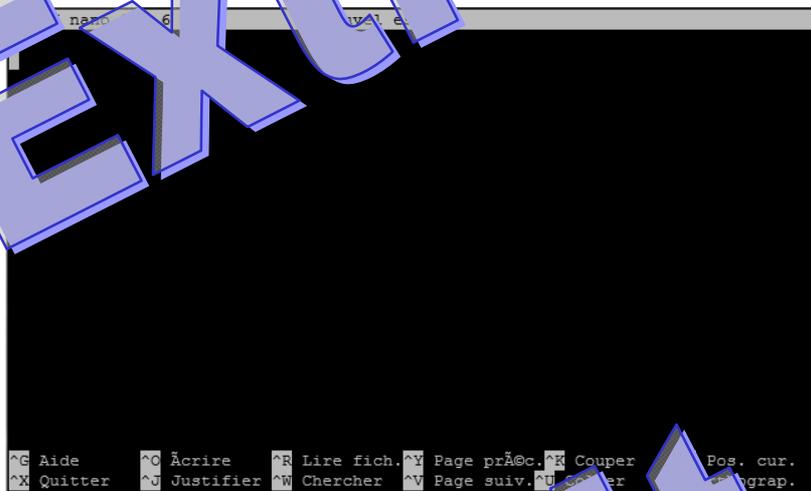
3.4.6 Edition de Fichiers

Lors de la création d'un fichier, il est indispensable de pouvoir l'éditer puis de la suite le modifier. Il existe 2 principaux éditeur ; nano très simple et pratique d'utilisation, Vim bien plus complexe.

Par soucis de simplicité, nous ne présenterons que nano, libre à vous d'étudier Vim.

3.4.6.1 L'éditeur Nano

Pour exécuter l'application, il suffit de taper à l'invite de commande nano



Dès lors vous êtes libre de taper le texte souhaité.

3.4.6.1.1 Les raccourcis de nano :

En bas de votre écran, vous pouvez voir l'aide de Nano. Que signifie-t-elle exactement ? Il s'agit d'un aide-mémoire pour vous rappeler tout simplement les raccourcis principales que vous pouvez lancer sous Nano.

Le symbole ^ signifie Ctrl (la touche contrôle, entre clavier et souris).

Voici les raccourcis les plus importants :

- Ctrl + G : l'aide
- Ctrl + O : couper la ligne de texte (et la mettre dans le presse-papier) ;
- Ctrl + U : coller le texte que vous venez de couper ;
- Ctrl + C : quel endroit du fichier votre curseur est positionné (numéro de ligne...) ;
- Ctrl + W : chercher dans le fichier ;
- Ctrl + O : enregistrer le fichier (écrire) ;
- Ctrl + X : quitter Nano.

Vous pouvez vous déplacer dans le fichier avec les flèches du clavier ainsi qu'avec les touches Page Up et Page Down pour avancer de page en page (les raccourcis Ctrl + Y et Ctrl + V fonctionnent aussi).

3.4.6.1.2 La recherche

La combinaison de touches Ctrl + W lance une recherche dans le fichier.

Il vous suffit d'écrire le mot que vous voulez rechercher...

... puis de taper Entrée.

Recherche dans Nano

Le curseur est automatiquement positionné à la première occurrence. Si le curseur est à la fin, la recherche recommence du début.

Si vous voulez sortir du mode recherche, tapez Ctrl + C (Abandonner).

Si vous souhaitez aller au résultat suivant (au cas où il y en a plusieurs), faites Ctrl + W pour lancer une recherche. La recherche précédente est sauvegardée et apparaît en arrière-plan. Si vous voulez rechercher le même mot (et donc aller au résultat suivant), appuyez juste sur la touche de mot à rechercher (figure suivante).

3.4.6.1.3 Enregistrer et quitter

Pour enregistrer le fichier, faites Ctrl + O.

Si vous essayez de quitter sans enregistrer auparavant, un message vous demandera si vous voulez sauvegarder.

Si vous appuyez sur la touche O, vous passerez en mode enregistrement.

Si vous appuyez sur la touche n, Nano quittera sans enregistrer.

Si vous utilisez la combinaison Ctrl + C, vous annulerez votre demande de sortie de Nano et ne quittera donc pas le logiciel.

En appuyant sur o, vous vous retrouvez en mode enregistrement. Tapez juste le nom du fichier que vous voulez créer puis pressez Entrée.

Après ça, Nano sera fermé et vous revenez en ligne de commandes.

3.4.6.1.4 Les paramètres de la commande Nano

Lorsque vous appelez Nano dans la ligne de commandes, vous pouvez spécifier plusieurs paramètres. Le plus courant est d'indiquer en paramètre le nom du fichier qu'on veut ouvrir:

```
nano mon_fichier
```

... ouvrira le fichier mon_fichier qui est en réalité une copie du fichier original au cas où le précédent.

Si le fichier n'existe pas, il sera automatiquement créé lors du premier enregistrement.

La commande nano accepte de nombreux paramètres. Voici les plus intéressants qui font partie des plus utiles.

- **-m** : autorise l'utilisation de la souris avec nano. En console, oui, oui. Vous pouvez vous en servir pour cliquer avec la souris sur la ligne de texte où vous voulez placer votre curseur.
- **-i** : indique en automatique que le tab (tabulation) de la ligne précédente sera respecté lorsque vous irez à la ligne suivante, même si vous êtes dans un fichier de code source.
- **-A** : active l'option intelligente de fin de ligne. Normalement, lorsque vous appuyez sur la touche Origine (elle est connue sous le nom de Home) située à côté de la touche Fin, le curseur se repositionne au tout début de la ligne. Avec cette commande, il se positionnera après les alinéas. Comme -i, il s'agit d'une option utile pour les programmeurs.

Nous pouvons donc lancer nano avec toutes ces options à la fois:

```
nano -miA mon_fichier
```

