

Command Line Interface

le système GNU/Linux Debian possède toutes les applications graphiques nécessaires pour accomplir vos tâches quotidiennes, alors pourquoi utiliser la ligne de commande ?

- c'est plus rapide
- les options ne sont pas toutes présentes dans les interfaces graphiques
- utiliser la commandes directement sans surcouche graphique économise les ressources
- ça facilite l'apprentissage du système GNU/Linux Debian

la commande ultime: **man**

Sommaire

Command Line Interface.....	1
naviguer dans les répertoires.....	1
actions sur les fichiers/dossiers.....	2
afficher/comparer les fichiers.....	2
utilisateurs.....	2
processus.....	3
tâches périodiques (cron).....	3
matériel.....	4
réseau.....	4
permissions.....	5
format des droits et permissions.....	5
changer les droits et permissions.....	6
recherche.....	6
archives.....	7
kernel.....	7

naviguer dans les répertoires

commande	action
<code>pwd</code>	affiche le répertoire courant
<code>cd rep</code>	se place dans le répertoire <i>rep</i>
<code>cd</code>	se place dans le répertoire de l'utilisateur <code>~/</code>
<code>cd ..</code>	se place dans le répertoire parent
<code>ls rep</code>	liste les fichiers du répertoire <i>rep</i>
<code>ls -a</code>	ls avec les fichiers cachés
<code>ls -l</code>	ls avec les droits d'accès

actions sur les fichiers/dossiers

commande	action
<code>mv source cible</code>	deplace le fichier <i>source</i> vers <i>cible</i>
<code>cp source cible</code>	copie le fichier <i>source</i> vers <i>cible</i>
<code>cp -R source cible</code>	copie le répertoire <i>source</i> vers <i>cible</i>
<code>ln source lien</code>	créer un lien fort de <i>source</i> vers <i>lien</i>
<code>ln -s source lien</code>	créer un lien symbolique de <i>source</i> vers <i>lien</i>
<code>touch file</code>	créer le fichier <i>file</i> ou met à jour sa date de modification
<code>mkdir rep</code>	créer un repertoire <i>rep</i>
<code>mkdir -p rep/rep2</code>	mkdir avec création des dossiers parents si nécessaire
<code>rm file</code>	supprime le fichier <i>file</i>
<code>rm -f file</code>	supprime le fichier <i>file</i> protégé en écriture
<code>rmdir rep</code>	supprimer un répertoire vide
<code>rm -R rep</code>	supprime un répertoire
<code>du -h file ou rep</code>	affiche la taille du fichier <i>file</i> ou du répertoire <i>rep</i>

afficher/comparer les fichiers

commande	action
<code>wc fichier</code>	compte le nombre de lignes, de mots, d'octets de <i>fichier</i>
<code>cat fichiers</code>	concatène les <i>fichiers</i>
<code>more fichier</code>	affiche <i>fichier</i> page après page. 'Espace'=page suivante, 'Entrée'=ligne suivante, 'u'=remonter
<code>less fichier</code>	affiche <i>fichier</i> avec une navigation Left/Right/Up/Down/PageUp/PageDown
<code>head -n x fichier</code>	affiche les x premières lignes de <i>fichier</i>
<code>tail -n x fichier</code>	affiche les x dernières lignes de <i>fichier</i>
<code>tail -f fichier</code>	affiche la dernière ligne de <i>fichier</i> au fur et à mesure qu'il se remplit
<code>diff file1 file2</code>	affiche les différences entre deux fichiers texte
<code>diff -u file1 file2</code>	affiche les différences au format utilisé par patch
<code>comp file1 file2</code>	compare deux fichiers binaires
<code>comp file1 file2 n N</code>	compare deux fichiers, <i>file1</i> à partir du nième octet, et <i>file2</i> à partir du Nième

utilisateurs

commande	action
<code>whoami</code>	affiche le login de l'utilisateur
<code>who</code>	affiche les utilisateurs connectés
<code>id</code>	afficher les <i>uid</i> , <i>gid</i> et <i>groupes</i> de l'utilisateur
<code>id user</code>	afficher les <i>uid</i> , <i>gid</i> et <i>groupes</i> de <i>user</i> (root only)
<code>finger user</code>	affiche les informations de <i>user</i>
<code>write user</code>	afficher un message sur le terminal de <i>user</i>
<code>tty</code>	afficher le nom de son terminal
<code>su - sudo</code>	passer en mode administrateur

passwd	changer le mot de passe de l'utilisateur courant
adduser	ajouter un utilisateur
deluser	supprime un utilisateur
addgroup	ajoute un groupe
delgroup	supprime un groupe

processus

commande	action
ps	afficher les processus de l'utilisateur rattachés à un terminal
ps ax	afficher tous les processus
ps aux	afficher tous les processus en indiquant les utilisateurs
pstree	afficher les processus dans une arborescence
top	afficher un tableau de bord des processus les plus gourmands
kill <i>signal</i> pid	tuer un processus en utilisant son pid
pkill <i>signal</i> nom	tuer un processus en utilisant le nom du programme

signaux utilisés par *kill/pkill*

signal	action
-1 (HUP)	recharger le fichier de configuration du processus
-2 (INT)	interrompre le processus
-3 (QUIT)	quitter le processus
-9 (KILL)	tuer le processus (à éviter, tenter -15 avant)
-15 (TERM)	terminer le processus proprement
-18 (STOP)	geler le processus
-20 (CONT)	reprandre l'exécution d'un processus gelé

tâches périodiques (cron)

commande	action
crontab -l	liste les tâches de l'utilisateur
crontab -e	editer les tâches de l'utilisateur
crontab -e -u <i>user</i>	editer les tâches de <i>user</i> (root only)

syntaxe de crontab

le fichier crontab est composé de commandes comprenant 5 champs temporels, un nom d'utilisateur si il s'agit du fichier de configuration principal et d'une commande à exécuter.

les 5 champs temporels:

- **m** minutes (0-59)
- **h** heures (0-23)
- **dom** numéro du jour du mois (1-31)
- **mon** numéro du mois (1-12)
- **dow** numéro du jour de la semaine (0-7 avec dimanche = 0 & 7)

les types de variables:

- **x,y** liste de valeurs
- **x-y** intervalle de valeurs
- ***** toute la valeur
- **/n** période

exemples:

crontab	action
0 5 * * * <i>commande</i>	exécute <i>commande</i> tous les jours à 5h00
* /5 * * * * <i>commande</i>	exécute <i>commande</i> toutes les 5 minutes
10 22 * * 2/2 <i>commande</i>	exécute <i>commande</i> un mardi sur deux à 22h10
53 21 * * 1-5 <i>commande</i>	exécute <i>commande</i> du lundi au vendredi à 21h53
36 18 10,20 * * <i>commande</i>	exécute <i>commande</i> les 10 et 20 du mois à 18h36
0 12 20 7 * <i>commande</i>	exécute <i>commande</i> tous les 20 juillet à 12h00 (anniversaire)

matériel

commande	action
lsusb	liste les périphériques de type <i>usb</i> connectés
lspci	liste les périphériques de type <i>pci</i> connectés
cat /proc/cpuinfo	affiche les informations processeur
cat /proc/partitions	affiche les partitions montées

exemples: (sources: [linuxtrack](#))

commande	action
lspci grep -i vga cut -d: -f3	afficher le modèle de sa carte graphique
lspci grep -i network cut -d: -f3	afficher le modèle de sa carte wifi
lspci grep -i audio cut -d: -f3	afficher le modèle de sa carte son

réseau

commande	action
hostname	affiche le nom d'hôte de la machine
ping <i>machine</i>	envoie un ping à une <i>machine</i>
traceroute <i>machine</i>	fait un traceroute vers <i>machine</i>
netstat	affiche l'usage du réseau pas les processus
netstat -a	netstat avec l'affichage des processus serveurs
lsdf	liste détaillée de l'usage des fichiers et du réseau
ifconfig	affiche la configuration des interfaces réseaux
ifconfig <i>interface IP masque</i>	configure une <i>interface</i> réseau
route	affiche la table de routage

<code>curl ifconfig.me</code>	vérifier son IP publique
<code>ip address show eth0 grep "inet " tr -s " " ":" cut -d: -f3</code>	vérifier son IP locale pour eth0
<code>/sbin/ifconfig eth0 grep "inet " tr -s " " ":" cut -d: -f4</code>	
<code>ip address show eth0 grep "inet " tr -s " " ":" tr -s "/" ":" cut -d: -f3</code>	

permissions

format des droits et permissions

format classique des droits et permissions d'un dossier/fichier: prenons l'exemple du dossier source de [wmfs](#)

```
[arp@thinkdeb]--[~]
└─┬─$ ls -l /home/arp/wmfs/wmfs
total 64
-rwxr-xr-x 1 arp arp 2449 Apr 1 18:05 configure
-rw-r--r-- 1 arp arp 1528 Apr 1 18:05 COPYING
drwxr-xr-x 3 arp arp 4096 Apr 1 18:05 debian
-rw-r--r-- 1 arp arp 2128 Apr 1 18:05 Makefile.in
-rw-r--r-- 1 arp arp 22 Apr 1 18:05 README
drwxr-xr-x 2 arp arp 4096 Apr 1 18:05 scripts
drwxr-xr-x 2 arp arp 4096 Apr 1 18:05 src
-rw-r--r-- 1 arp arp 12741 Apr 1 18:05 wmfs.1
-rw-r--r-- 1 arp arp 117 Apr 1 18:05 wmfs.desktop
-rw-r--r-- 1 arp arp 1320 Mar 26 16:26 wmfs-gdb.log
-rw-r--r-- 1 arp arp 8629 Apr 1 18:05 wmfsrc
```

les permissions: -rwxr-xr-x ou drwxr-xr-x

le premier caractère correspond au **type d'objet**. les suivants se lisent par groupe de trois et définissent respectivement les **droits accordés**:

- au propriétaire du fichier/dossier
- au groupe propriétaire du fichier/dossier
- aux autres utilisateurs

label	type d'objet
-	fichier
d	répertoire
l	lien
s	socket
p	fifo
c	device (mode caractère)
b	device (mode block)

label	type de droits
-	pas d'accès
r	accès en lecture
w	accès en écriture
x	accès en exécution pour les fichiers, traversée pour les dossiers
s	bit set-UID (user), bit set-GID (groupe)
t	sticky bit: impossible d'effacer les fichiers des autres
S	bit set-UID sans les droits d'exécution
T	sticky bit sans droits de travesée

changer les droits et permissions

commande	action
chown <i>user fichier</i>	change le propriétaire de <i>fichier</i> à <i>user</i>
chgrp <i>groupe fichier</i>	change le groupe de <i>fichier</i> à <i>groupe</i>
chmod <i>mode fichier</i>	change les droits d'accès au <i>fichier</i>

le **mode** de *chmod* peut être sous la forme groupe(s)+droit(s) ou sous la forme octale **SUGO/UGO**

- la forme groupe(s)+droit(s): ou *groupe*= **u** pour user, **g** pour groupe et **o** pour others.

```
$ chmod ugo+rx $HOME/repertoire      ## permet à tous de lister (lire) et
traverser le répertoire
$ chmod go-rwx $HOME/repertoire/fichier  ## retire tous les droits à tous sauf
au propriétaire du fichier
```

- la forme octale **SUGO** où:
 - S**: 1 (sticky-bit), 2 (set-GID) ou 4 (set-UID)
 - U,G et O**: 1 (exécutable), 2 (écriture), 4(lecture). on additionne les formes par chiffre pour obtenir un mode octal à 4 chiffres.

- \$ chmod 1750 \$HOME/repertoire ## le propriétaire et le groupe peuvent lire et traverser le répertoire, seul le propriétaire peut y écrire ou l'effacer.
- \$ chmod 660 \$HOME/fichier ## le fichier est accessible au propriétaire et au groupe en lecture et écriture

recherche

commande/option	action
locate <i>motif</i>	recherche des fichiers dont le nom correspond au <i>motif</i>
updatedb	mettre à jour la base de données de locate
find <i>chemin options</i>	recherche les fichiers correspondant aux <i>options</i> dans <i>chemin</i>
find -name <i>motif</i>	recherche sur le nom du <i>fichier</i> . ex: find -name '*.html'
find -type <i>f/d/l</i>	recherche par type où f=fichier, d=répertoire et l=lien
find -exec <i>cmd</i>	exécute la commande <i>cmd</i> à tous les fichiers trouvés

exemple: trouver toutes les images avec l'extension *png* dans le dossier images de l'utilisateur et les copier dans le dossier *tmp* ('{}' représente les fichiers trouvés).

```
find $HOME/pics -name '*.png' -exec cp {} $HOME/tmp/ \;
```

archives

les commandes utilisées possèdent bien plus d'options que celles présentées ici, je vous invite à consulter les manuels concernés.

format	compression	extraction
.tar.bz2, .tbz2	<code>tar -cvjf archive.tar.bz2 repertoire</code>	<code>tar xvjf</code>
.tar.gz, .tgz	<code>tar -cvzf archive.tar.gz repertoire</code>	<code>tar xvzf</code>
.bz2	<code>bzip2 fichiers</code>	<code>bunzip2</code>
.rar	-	<code>unrar x</code>
.gz	<code>gzip fichiers</code>	<code>gunzip</code>
.tar	<code>tar -cvf archive.tar fichiers</code>	<code>tar xvf</code>
.zip	<code>zip -r archive.zip fichiers</code>	<code>unzip</code>
.Z	<code>compress fichiers</code>	<code>uncompress</code>
.7z	<code>7z a fichiers</code>	<code>7z x</code>
.xz	<code>xz -z repertoire</code>	<code>unxz</code>

kernel

diverses infos sur le noyau linux utilisé

commande	description
<code>cat /proc/version</code>	version du noyau Linux utilisé, son nom, la version du compilateur utilisé.
<code>uname -r</code>	version du noyau utilisé.
<code>dpkg -l grep -Ei "linux-headers linux-image"</code>	liste les noyaux installés sur votre machine.