

Linux Utilisation

Module 04 – Fichiers et dossiers sous Linux



1

Fichiers et dossiers sous Linux



Objectifs

- Découvrir le standard des dossiers Unix (FHS)
- Savoir créer, copier, déplacer et supprimer des fichiers et dossiers
- Savoir se situer et se déplacer dans l'arborescence de fichiers
- Utiliser les métacaractères de bash

Savoir manipuler les dossiers et fichiers



2

Arborescence de fichiers : les chemins



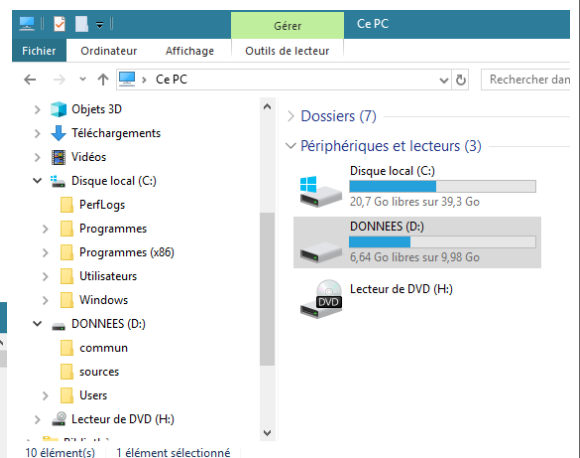
3

Arborescence de fichiers Arborescence Windows

- Plusieurs lecteurs
- Plusieurs racines séparées
- En terminal, besoin de passer d'une racine de lecteur à une autre

```

C:\>cd d:\sources
C:\>d:
d:\sources>
  
```



4

- Une racine unique /
- Possibilité de se déplacer partout en une seule commande
- Une arborescence standard
- Connexion des disques & lecteurs inclus



- Les dossiers sont séparés par le caractère /

```
user30@deb:~$ cd /home/user/rep01
user30@deb:~/rep01$
```

- Si un fichier doit être précisé, il est ajouté après le dernier / **sans espace**

```
user30@deb:~$ cat /home/user/rep01/bonjour.txt
Bonjour le monde !
```



- Contient les répertoires personnels des utilisateurs.
- Le chemin du répertoire personnel d'un utilisateur est contenu dans la variable d'environnement **\$HOME**

```
user30@deb:~$ echo $HOME  
/home/user30
```

```
user29@deb:~$ echo $HOME  
/home/user29
```



- Chemin relatif : écriture depuis l'emplacement actuel

```
user30@deb:~$ cd dossier01  
user30@deb:~/dossier01$
```

- Chemin absolu : écrire le chemin complet depuis la racine du système de fichier

```
user30@deb:~$ cd /home/user30/dossier01  
user30@deb:~/dossier01$
```



- Chaque répertoire, en dehors de la racine, contient au minimum deux sous-répertoires particuliers :
 - .. Le répertoire parent
 - . Le répertoire courant

```
user30@deb:~/dossier01$ cd ..  
user30@deb:~$
```

```
user30@deb:~/dir02/dir03$ cd ../../dossier01  
user30@deb:~/dossier01$
```



- Le caractère . est utile pour lancer des scripts ou préciser le répertoire actuel.

```
user30@deb:~/scripts$ ./monscript.sh
```

```
user30@deb:~$ find . -name "*.txt"  
./aFaire.txt  
./fic01.txt  
./contacts.txt
```



Arborescence de fichiers : le FHS



11

- FHS : Standardisation de l'arborescence Unix Linux
- Actuellement en version 3.0



12

Répertoire	Description
/home	Répertoire contenant les dossiers de tous les utilisateurs hormis root
/home/dom	Répertoire de l'utilisateur dont le login est dom
/root	Répertoire personnel de l'utilisateur root
/media	Points de montages des médias amovibles (clé USB, CD-rom...)
/mnt	Points de montage temporaires de volumes disques



Répertoire	Description
/bin	Commandes systèmes disponibles aux utilisateurs standards. Peut être un lien de /usr/bin
/usr/bin	Commandes systèmes disponibles aux utilisateurs standards
/usr/local/bin	Commandes systèmes disponibles aux utilisateurs standards
/sbin	Commandes d'administration du super-utilisateur root Peut être un lien de /usr/sbin
/usr/sbin	Commandes d'administration du super-utilisateur root
/usr/local/sbin	Commandes d'administration du super-utilisateur root
/opt	Répertoires pour les applications tierces (compilation manuelle de logiciels)



Répertoire	Description
/boot	Fichiers nécessaires au démarrage du système
/dev	Liaisons des périphériques (disques, claviers, écrans ...)
/etc	Contient les fichiers de paramètres du système
/var	Données de tailles variables (sites web, bases de données, journaux...)
/var/log	Dossiers contenant les fichiers de journalisation du système
/lib	Contient les bibliothèques principales utiles aux exécutables dans /bin et /sbin
/tmp	Fichiers temporaires
/proc	Répertoire virtuel contenant les fichiers de paramètres du noyau
/sys	Répertoire virtuel contenant les informations des périphériques, pilotes...



15



Arborescence Linux



16

Manipuler des dossiers



17

cd <dossier cible>



```
user30@deb:~$ cd dossier01
user30@deb:~/dossier01$ cd $HOME
user30@deb:~$ cd dir02/dir03
user30@deb:~/dir02/dir03$ cd
user30@deb:~/ cd -
user30@deb:~/dir02/dir03$ cd ~
user30@deb:~$
```



18

`mkdir [-pv] <nouveau dossier>`



```
user30@deb:~$ mkdir dossier01
user30@deb:~$ mkdir -v dossier02
mkdir: création du répertoire 'dossier02'
user30@deb:~$ mkdir -p -v dir02/dir03
mkdir: création du répertoire 'dir01'
mkdir: création du répertoire 'dir01/dir02'
```

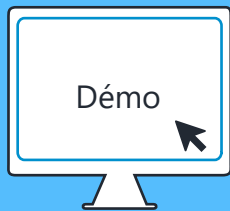


`rmdir [-pv] <nouveau dossier>`



```
user30@deb:~$ rmdir -v dossier01
rmdir: suppression du répertoire, 'dossier01'
user30@deb:~$ rmdir -pv dir02/dir03
rmdir: suppression du répertoire, 'dir02/dir03/'
rmdir: suppression du répertoire, 'dir02'
user30@deb:~$
```






- Manipuler des dossiers
-



Lister des fichiers et dossiers



ls [-aAlrRt] <dossier ou fichier>



```
user30@deb:~$ ls
aFaire.txt      dossier01  fic01.txt
contacts.txt    Edition
user30@deb:~$ ls -l
total 12
-rw-r--r-- 1 user30 user30   36 juin  26 12:08 aFaire.txt
-rw-r--r-- 1 user30 user30    0 juil.  2 10:47 contacts.txt
drwxr-xr-x 2 user30 user30 4096 juil.  1 11:04 dossier01
-rw----- 1 user30 user30  586 avril 16 17:26 Edition
-rw-r--r-- 1 user30 user30    0 juil.  2 10:47 fic01.txt
```



Détail des colonnes du format long

1	2	3	4	5	6	7
-rw-r--r--	1	user30	user30	1568	juil. 2 10:47	fic01.txt

- 1 : type de fichier et droits
- 2 : nombre de liens physiques
- 3 : utilisateur propriétaire
- 4 : groupe propriétaire
- 5 : Taille du fichier
- 6 : date / heure de modification
- 7 : nom du fichier ou dossier

- Les différents types de fichiers :
- fichier standard
 - d répertoire
 - l lien symbolique
 - b périphérique de type bloc
 - c périphérique de type caractère
 - p pipe (tube) nommé
 - s socket unix





```
user30@deb:~$ ls -a
.          .bash_history  .bashrc      fic01.txt
..         .bash_logout  contacts.txt  Edition
user30@deb:~$ ls -A
aFaire.txt .bash_logout  .bashrc      Edition
.bash_history contacts.txt  fic01.txt    .profile
user30@deb:~$ ls -lA
total 40
-rw-r--r-- 1 user30 user30  36 juin  26 12:08 aFaire.txt
-rw----- 1 user30 user30 757 juil.  2 02:21 .bash_history
-rw-r--r-- 1 user30 user30 220 avril 18  2019 .bash_logout
```



```
user30@LinuxUtil:~$ ls -l dossier01
total 113720
-rw-r--r-- 1 user30 user30 104857600 juil.  3 10:53 fic100M
-rw-r--r-- 1 user30 user30  10485760 juil.  3 10:53 fic10M
-rw-r--r-- 1 user30 user30   1048576 juil.  3 10:53 fic1M
-rw-r--r-- 1 user30 user30    51200 juil.  3 10:54 fic50k
user30@LinuxUtil:~$ ls -lh dossier01
total 112M
-rw-r--r-- 1 user30 user30 100M juil.  3 10:53 fic100M
-rw-r--r-- 1 user30 user30  10M juil.  3 10:53 fic10M
-rw-r--r-- 1 user30 user30 1,0M juil.  3 10:53 fic1M
-rw-r--r-- 1 user30 user30  50K juil.  3 10:54 fic50k
```





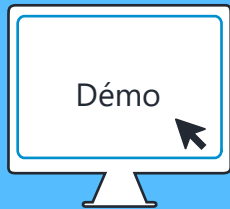
```
user30@LinuxUtil:~$ ls -l dossier01
total 113720
-rw-r--r-- 1 user30 user30 104857600 juil.  3 10:53 fic100M
-rw-r--r-- 1 user30 user30  10485760 juil.  3 10:53 fic10M
-rw-r--r-- 1 user30 user30   1048576 juil.  3 10:53 fic1M
-rw-r--r-- 1 user30 user30    51200 juil.  3 10:54 fic50k
user30@LinuxUtil:~$ ls -ld dossier01
drwxr-xr-x 2 user30 user30 4096 juil.  3 11:12 dossier01/
```



```
user30@deb:~$ ls -lt
total 12
-rw-r--r-- 1 user30 user30    0 juil.  2 10:47 contacts.txt
drwxr-xr-x 2 user30 user30 4096 juil.  1 11:04 dossier01
-rw-r--r-- 1 user30 user30   36 juin  26 12:08 aFaire.txt
-rw----- 1 user30 user30  586 avril  16 17:26 Edition
user30@deb:~$ ls -lr
total 12
-rw----- 1 user30 user30  586 avril  16 17:26 Edition
drwxr-xr-x 2 user30 user30 4096 juil.  1 11:04 dossier01
-rw-r--r-- 1 user30 user30    0 juil.  2 10:47 contacts.txt
-rw-r--r-- 1 user30 user30   36 juin  26 12:08 aFaire.txt
user30@deb:~$ ls -lrt /var/log
[...]
```

```
-rw-rw-r-- 1 root utmp 310396 juil.  3 09:52 lastlog
-rw-r----- 1 root adm 318286 juil.  3 10:07 syslog
-rw-r----- 1 root adm  70290 juil.  3 10:07 daemon.log
```





-
- Utilisation de la commande ls
-



Manipuler des fichiers



`touch <nouveau fichier>`



```
user30@deb:~$ ls fichier01.txt
ls: impossible d'accéder à 'fichier01.txt': Aucun fichier
ou dossier de ce type
user30@deb:~$ touch fichier01.txt
user30@deb:~$ ls -l fichier01.txt
-rw-r--r-- 1 user30 user30 0 juil.  3 11:12 fichier01.txt
```




`touch <fichier existant>`



```
user30@deb:~$ ls -l aFaire.txt
-rw-r--r-- 1 user30 user30 36 juin  26 12:08 aFaire.txt
user30@deb:~$ touch aFaire.txt
user30@deb:~$ ls -l aFaire.txt
-rw-r--r-- 1 user30 user30 36 juil.  3 11:16 aFaire.txt
```



`mv [-fiv] <source> <cible>`



```
user30@deb:~$ mv fichier01.txt /tmp/  
user30@deb:~$ ls /tmp/fichier01.txt  
/tmp/fichier01.txt  
user30@deb:~$ mv -iv fic01.txt /tmp  
mv : voulez-vous écraser '/tmp/fic01.txt' ? o  
renommé 'fic01.txt' -> '/tmp/fic01.txt'
```



`mv <anciennom> <nouveaunom>`




```
user30@deb:~$ mv -v aFaire.txt Fait.txt  
renommé 'aFaire.txt' -> 'Fait.txt'
```

Il est bien sûr possible de déplacer un dossier.



```
user30@deb:~$ mv -v dossier01/ dirtest01  
renommé 'dossier01/' -> 'dirtest01'
```


Enfin on peut tout à fait déplacer et renommer en une seule commande



```
user30@deb:~$ mv -v dossier01/ dirtest01  
renommé 'dossier01/' -> 'dirtest01'
```



`cp [-fiv] <source> <cible>`

 user30@deb:~\$ `cp -v contacts.txt contacts2.txt`
'contacts.txt' -> 'contacts2.txt'

Il est possible de copier un fichier à un autre emplacement

 user30@deb:~\$ `cp -v contacts.txt ./dossier2/`
'contacts.txt' -> './dossier2/contacts.txt'


On peut copier un fichier dans un autre dossier avec un nouveau nom

 user30@deb:~\$ `cp -pv contacts.txt /tmp/contacts3`
'contacts.txt' -> '/tmp/contacts3'



On peut copier plusieurs fichiers ou dossiers vers un dossier cible existant

`cp [-prv] <ficsource> <ficsource> <dirsource> <dossiercible>`

 user30@deb:~\$ `mkdir perso`
user30@deb:~\$ `cp -rpv contacts.txt contacts-2.txt dirtest01/ perso/`
dirtest01/ perso/
'contacts-2020.txt' -> 'perso/contacts-2020.txt'
'contacts.txt' -> 'perso/contacts.txt'
'dirtest01/' -> 'perso/dirtest01'
'dirtest01/fic1M' -> 'perso/dirtest01/fic1M'
'dirtest01/fic10M' -> 'perso/dirtest01/fic10M'
'dirtest01/fic01.txt' -> 'perso/dirtest01/fic01.txt'



`rm [-firv] <fic1> <fic1>`



```
user30@deb:~$ rm -v /tmp/fichier01.txt contacts-2.txt  
'/tmp/fichier01.txt' supprimé  
'contacts-2.txt' supprimé
```



`rm [-rvi] <dossier>`



```
user30@deb:~$ rm -rv dirtest01/  
'dirtest01/fic1M' supprimé  
'dirtest01/fic50k' supprimé  
'dirtest01/fichier01.txt' supprimé  
'dirtest01/fic100M' supprimé  
'dirtest01/fic10M' supprimé  
répertoire 'dirtest01/' supprimé
```



`rm -rf <dossier>` <-- Attention



```
user30@deb:~$ rm -rf $HOME
```

EFFACE TOUT VOS DOSSIERS ET FICHIERS PERSONNELS !



`rmdir -v <dossier>`



```
user30@deb:~$ touch dirtest02/fictest02
```

```
user30@deb:~$ rmdir -v dirtest02/
```

rmdir: impossible de supprimer 'dirtest02/': Le dossier n'est pas vide

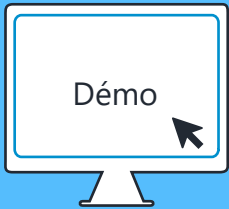
```
user30@deb:~$ mkdir dirtest03
```

```
user30@deb:~$ rmdir -v dirtest03/
```

rmdir: suppression du répertoire, 'dirtest03'




Manipuler des fichiers




Démonstration

-
- Manipulation de fichiers
-




41

Manipuler des fichiers



TP

-
- Manipulation de fichiers
-



42

Métacaractères and co



43

- Les métacaractères sont des caractères spéciaux, de substitution, qui servent à créer des recherches ou sélections génériques.
- Il s'agit d'un mécanisme très important du bash. Les utilisateurs d'Unix et Linux les utilisent au quotidien.
- S'ils empruntent à certains standards, notamment les expressions régulières qui seront abordées plus loin, la plupart de ces métacaractères et autres méthodes vues dans cette partie sont spécifiques aux shells Unix/Linux.



44

- Certains métacaractères vous sont déjà connus, principalement l'étoile *



```
user30@deb:~$ ls *.txt
contacts.txt  Fait.txt  liste.txt  Tel-2020.txt
edition1.txt  l2.txt    Tel2018.txt  tel2018.txt
```

- Dans ce premier exemple, on liste tous les fichiers qui se terminent par **.txt**



```
user30@deb:~$ ls tel*
tel2018  tel2018.txt  tel2020
```

- Ce second exemple liste tous les fichiers qui commencent par **"tel"**



- Assez proche dans son fonctionnement on trouvera le **?**. Celui-ci indique un caractère quelconque et un seul.



```
user30@deb:~$ ls tel20??
tel2018  tel2020  tel20xx
```

- Dans cet exemple, on liste tous les fichiers qui commencent par tel20 et possèdent ensuite deux caractères quelconque.



- De manière plus précise, on peut définir des listes de caractères pouvant valider notre sélection. On utilise les syntaxes [...] et [...-...] .
- [...] indique la possibilité d'utiliser un caractère parmi ceux indiqués.
- [...-...] indique la possibilité d'utiliser un caractère présent dans un intervalle.

```
user30@deb:~$ ls tel20[12][0-9]
tel2018  tel2020
```

Le premier choix **[12]** indique le choix entre le chiffre 1 et le chiffre 2.
La deuxième partie **[0-9]** indique un caractère entre 0 et 9, autrement dit : un chiffre.



- On peut lister tout type de caractères, à commencer par des lettres.

```
user30@deb:~$ ls [tT][a-zA-Z]*
tel2018  Tel2018  tel2020  Tel-2020.txt  toto  tel20xx
```

- Ici on donne le choix entre t minuscule ou T majuscule **[tT]** . Ensuite **[a-zA-Z]** précise que l'on veut une lettre en minuscule ou majuscule. Enfin ***** permet d'indiquer n'importe quel caractère à suivre, ou aucun.

- On peut mélanger les critères à l'envie.


```
user30@deb:~$ ls [tT]*[0-9][0-9]
tel2018  tel2020  Tel2018
```



- Il est possible de reprendre les listes précédentes en inversant la recherche avec ces deux syntaxes.

- `[^...]` / `[^...-...]`

- `[!...]` / `[!...-...]`



```
user30@deb:~$ ls -d [!tT]*
contacts2020  dirttest01  edition1.txt  l2.txt      newcontacts
contacts.txt  Edition     Fait.txt     liste.txt   perso
```

- Ici on affiche tous les fichiers et dossiers ne commençant pas par la lettre **t**.



Pour compléter l'utilisation de métacaractères et de classes de caractères, les *facteurs d'occurrence* permettent de définir le nombre de fois où une recherche est effectuée.

- `?(...)` 0 à 1 fois : optionnel une seule fois
- `* (...)` 0 à n fois : optionnel illimité en nombre
- `+ (...)` 1 à n fois : au moins une fois
- `@ (...)` 1 fois : exactement une fois
- `!(...)` l'expression entre parenthèses ne sera pas présente



```
user30@deb:~$ ls -d +([eE])*  
Edition  edition1.txt  eedition  Eedition  eeeeexemple
```

- La recherche +([eE]) indique que l'on veut au moins une fois le caractère e en majuscule ou en minuscule.

```
user30@deb:~$ ls -d !(*.txt)  
contacts2020  dirtest01  Edition  eedition  Eedition  
eeeeexemple  newcontacts  perso  tel2018  tel2020  toto
```

- La recherche !(*.txt) indique tous les fichiers sauf ceux se terminant par l'extension txt.



NB : L'option du shell extglob doit être activée. Cela se fait et vérifie avec la commande shopt


```
user30@deb:~$ shopt -s extglob  
user30@deb:~$ shopt  
[...]  
extglob    on  
[...]
```



- On peut utiliser les accolades pour offrir un choix parmi plusieurs possibilités.
- Par exemple pour rechercher plusieurs extensions de fichiers à la fois.

 user30@deb:~\$ `ls images/*.{jpeg,gif,png}`
images/icon1.gif images/icon2.gif images/logo.png
images/wallp.jpeg

- Cette syntaxe ne se limite pas à la recherche

 user30@deb:~\$ `mkdir -v {image,video}`
mkdir: création du répertoire 'image'
mkdir: création du répertoire 'video'



Le bash nous permet également l'utilisation d'autres caractères spéciaux avec les utilisations suivantes :

- | | |
|------------------------------|--|
| • ' ' Simples quotes | Banalise tous les caractères situés entre les quotes |
| • " " Doubles quotes | Banalise tous *SAUF \$, \ et ` (backquote) |
| • \$ Dollar | Il permet d'afficher le contenu d'une variable |
| • \ Antislash | Il banalise le caractère qui le suit (\$, *, ? ...) et permet de l'afficher sans l'interpréter |
| • \$(...) Dollar parenthèses | Récupérer le résultat de la commande entre parenthèses, par exemple dans une variable |
| • `...` backquotes | Même résultat que ci-dessus, mais syntaxe dépréciée |





Utilisation de métacaractères



TP

Utilisation de métacaractères



- Vous connaissez l'arborescence des dossiers d'un système Unix/Linux
- Vous savez vous déplacer dans les dossiers et afficher leur contenu
- Vous savez créer, déplacer, copier et supprimer des fichiers et des dossiers
- Vous savez effectuer des actions complexes grâce aux métacaractères

