



La manipulation des répertoires

Principales commandes pour gérer répertoires

cd (Change Directory)

Changement de répertoire de travail

- Syntaxe :


```
cd [nom_répertoire]
```
- Description :
 - nom_répertoire en relatif ou absolu, si nom_répertoire n'est pas indiqué alors retourne au répertoire d'accueil
- Exemple :


```
hans> cd
hans> cd repl
hans> cd ..
```

http://ltdm.fr/~jean-paul.ARDRECHY/28_Telcom/PhdDoc



La manipulation des répertoires

Principales commandes pour gérer répertoires

pwd (Print Working Directory)

Affiche le chemin d'accès complet du répertoire courant

- Syntaxe :


```
pwd
```
- Description :
 - Affiche le chemin d'accès complet du répertoire courant
- Exemple :


```
hans> pwd
/home/hans/cours
hans> cd
hans> pwd
?????/?
```

http://ltdm.fr/~jean-paul.ARDRECHY/28_Telcom/PhdDoc



La manipulation des répertoires

Principales commandes pour gérer répertoires

mkdir (MaKe DIRectory)

Création d'un ou plusieurs répertoires

- Syntaxe :


```
mkdir [-p] nom_répertoire[...]
```
- Options :
 - p : crée tous les répertoires intermédiaires, si nécessaire (--parent)
- Description :
 - création du répertoire pour chaque nom passé en argument
- Exemple :


```
hans> mkdir repl
hans> mkdir repl/srepl
hans> mkdir -p a/b/c (équivalent mkdir a a/b a/b/c)
```

http://ltdm.fr/~jean-paul.ARDRECHY/28_Telcom/PhdDoc



La manipulation des répertoires

Principales commandes pour gérer répertoires

rmdir (ReMove DIRectory)

Suppression d'un ou plusieurs répertoires

- Syntaxe :


```
rmdir [-p] nom_répertoire[...]
```
- Options :
 - p : supprime tous les répertoires intermédiaires, si vide (--parent)
- Description :
 - suppression du répertoire vide pour chaque nom passé en argument
- Exemple :


```
hans> rmdir repl/srepl
hans> rmdir repl
hans> rmdir -p a/b/c (équivalent rmdir a/b/c a/b a)
```

http://ltdm.fr/~jean-paul.ARDRECHY/28_Telcom/PhdDoc



La manipulation des fichiers

Principales commandes de travail sur les fichiers

ls (liSt files)

Affiche le contenu d'un répertoire

- Syntaxe :


```
ls [-alF...] [nom_répertoire...]
```
- Description :
 - sans argument, affiche la liste du répertoire courant
- Options :
 - l : liste détaillée des caractéristiques de chaque fichier
 - a : affiche également les fichiers commençant par le caractère . (fichiers cachés)
 - F : liste fichiers avec indication du type (/ : répertoire * : exécutable,)
- Exemple :


```
hans> ls -F
a/ toto.txt prog*
hans>
```

http://ltdm.fr/~jean-paul.ARDRECHY/28_Telcom/PhdDoc



La manipulation des fichiers

Principales commandes de travail sur les fichiers

touch (TOUCH)

Modification date d'accès au fichier

- Syntaxe :


```
touch fichier
```
- Description :
 - création d'un fichier vide
- Exemple :


```
hans> touch f1
hans> ls -F
a/ toto.txt prog* f1
hans>
```

http://ltdm.fr/~jean-paul.ARDRECHY/28_Telcom/PhdDoc

 **La manipulation des fichiers**

Principales commandes de travail sur les fichiers

cp (CoPy)

Copie fichiers et répertoires

- Syntaxe :


```
cp [-ir] source destination
cp [-ir] source[...] repertoire_destination
```
- Description : Copie source vers destination ou multiple sources vers repertoire_destination
- Options :
 - i : demande de confirmation en cas de copie sur fichier existant
 - r : copie récursive
- Exemple :


```
hans> cp f1 f2
hans> cp f1 repl
hans> cp -r repl save
```

hans@hans:~\$ cat /etc/passwd | grep hans

 **La manipulation des fichiers**

Principales commandes de travail sur les fichiers

mv (MoVe list)

Renomme le nom d'un fichier

- Syntaxe :


```
mv [-i] source destination
mv [-i] source[...] repertoire_destination
```
- Description : renomme source par destination ou déplace source vers repertoire_destination
- Options :
 - i : demande de confirmation en cas de copie sur fichier existant
- Exemple :


```
hans> mv f1 fichier1
hans> mv save sauvegarde
hans> mv fichier1 fichier2 repl
```

hans@hans:~\$ cat /etc/passwd | grep hans

 **La manipulation des fichiers**

Principales commandes de travail sur les fichiers

rm (ReMove files)

Supprime un ou plusieurs fichiers

- Syntaxe :


```
rm [-ir] fichier[...]
```
- Description : supprime un ou plusieurs fichiers
- Options :
 - i : demande de confirmation
 - r : suppression récursive
- Exemple :


```
hans> rm -i fichier1
hans> rm -r repl
```

hans@hans:~\$ cat /etc/passwd | grep hans

 **La manipulation des fichiers**

Principales commandes de travail sur les fichiers

less

Affiche le contenu d'un fichier

- Syntaxe :


```
less fichier
```
- Description : more ou less affiche le contenu d'un fichier page par page.
- Exemple :


```
hans> less /etc/passwd
affichage page par page du fichier
/etc/passwd
hans>
```

hans@hans:~\$ cat /etc/passwd | grep hans

 **La manipulation des fichiers**

Principales commandes de travail sur les fichiers

ln

Permet d'établir un lien entre deux (plusieurs) noms et un fichier

- Syntaxe :


```
ln fichier_source fichier_cible
```
- Description : Permet de désigner un fichier par plusieurs nom, un seul espace disque
- Exemple :


```
hans> ln f2 f2_new
Le fichier porte deux noms mais n'occupe qu'un seul espace
disque.
hans> ls
f2 f2_new
```

hans@hans:~\$ cat /etc/passwd | grep hans

 **Modification des protections**

Notation relative (aux droits existants)

`chmod [-R] [who]op[permission] fichiers`

- op prend les valeurs + ou - et permet d'ajouter ou supprimer un droit
- Exemple :


```
hans> chmod -R o-rwx /home/hans
enlève toutes les permissions d'accès des fichiers du rép.
personnel de hans (et des sous-rép.) aux autres (others).
hans> chmod -R g-rwx,o-rwx /home/hans
enlève toutes les permissions d'accès des fichiers du rép.
personnel de hans (et des sous-rép.), à tous sauf au
propriétaire, c'est-à-dire hans.
```

hans@hans:~\$ cat /etc/passwd | grep hans



Modification des protections

- **Notation absolue**

```
chmod [-R] [who]op[permission] fichiers
```

- Pour chaque type d'utilisateurs, elle permet de fixer les nouveaux droits qui remplacent les anciens. Si un type n'est pas présent, ses anciens droits s'appliquent.
- Exemple :


```
hans> chmod u=rwx,g=rw,o=r fichier
```

 remplace les permissions précédentes de fichier, en les fixant à `rw-rw-r--`.


```
hans> chmod u=rwx,g=r fichier
```

 fixe pour fichier les permissions à `rw-r--???` en ne changeant pas les permissions précédentes du groupe



Protection par défaut

umask

Permet d'établir la protection par défaut.

- Syntaxe :


```
umask [-S] [who]op[permission]
```
- Description :

Permet d'établir la protection par défaut. L'option `-S` permet d'afficher les protections par défaut de l'utilisateur en mode symbolique.
- Exemple :


```
hans> umask u=rwx,g=rx,o=r
hans> umask -S
u=rwx,g=rx,o=r
hans> touch f1
hans> ls -l f1
-rw-r--r-- 1 hans hans 0 Sep 21 12:55 f1
```



Protection par défaut

chgrp

Changement de groupe.

- Syntaxe :


```
chgrp [-R] group fichiers[...]
```
- Description :

Permet le changement de groupe pour les fichiers et répertoires. L'option `-R` permet la récursivité de la commande.
- Exemple :


```
/home/hans/> ls -l f1
-rw-rw-r-- 1 hans hans 0 Sep 21 12:55 f1
/home/hans> chgrp etudiant f1
/home/hans> ls -l f1
-rw-rw-r-- 1 hans etudiant 0 Sep 21 12:55 f1
```



Les filtres

less (more)

Affiche le contenu d'un fichier à l'écran

- Syntaxe :


```
less [-option] [fichier...]
```
- Options :
- Description :

filtre qui affiche page par page le contenu d'un fichier ou l'entrée standard
- Exemple :


```
hans> less /etc/passwd
```

 Affiche le contenu de passwd page par page


```
hans>
```



Les filtres

tail

liste le contenu d'un fichier à partir de la ligne n

Exemple :

```
hans> tail +5 /etc/passwd
```

Affiche le contenu de passwd partir de la ligne 5

```
hans>
```

head

liste les n premières lignes d'un fichier

Exemple :

```
hans> head -5 /etc/passwd
```

Affiche le contenu de passwd de la ligne 1 à la ligne 5

```
hans>
```



Les filtres

tail

liste le contenu d'un fichier à partir de la ligne n

- Syntaxe :


```
tail [+ -n] [fichier...]
```
- Options :
 - `+n` affiche à partir de la n^{ème} ligne à partir du début
 - `-n` affiche à partir la n^{ème} ligne à partir de la fin
- Description :

filtre qui affiche à partir de la n ligne du début ou de la fin de l'entrée standard
- Exemple :


```
hans> tail +5 /etc/passwd
```

 Affiche le contenu de passwd partir de la ligne 5


```
hans>
```

 **Les filtres**

head

liste les n premières lignes d'un fichier

- Syntaxe :
`head [-n] [fichier...]`
- Options :
 - -n affiche les n premières lignes
- Description :
filtre qui affiche les n premières lignes d'un fichier ou de l'entrée standard
- Exemple :
hans> `head -5 /etc/passwd`
Affiche le contenu de passwd de la ligne 1 à la ligne 5
hans>

http://ltdlmur - 300-Pdf-APPRECHU-28-ENQP-2011-2012

 **Les filtres**

sort

Lie, trie le contenu d'un fichier et affiche le résultat à l'écran

- Syntaxe :
`sort [-options] [+pos1] [-pos2][fichier...]`
- Options :
 - -o file résultat du tri dans le fichier file
 - -tx choix du séparateur de champs (x) (espace ou tab par défaut)
 - +pos1 exclure les champs de 1 à c1
 - -pos2 exclure les champs à partir de c2
- Description :
filtre qui lie et trie le contenu d'un fichier ou l'entrée standard en s'appuyant sur la notion de champs par ligne.
- Exemple :
hans> `sort -t: +2, -3 -o passwd.tri /etc/passwd`
Tri le contenu de passwd sur le numéro UID (3ième champs) et met le résultat dans passwd.tri du répertoire courant.
hans>

http://ltdlmur - 300-Pdf-APPRECHU-28-ENQP-2011-2012

 **La gestion des processus : la commande ps**

ps

donne un ensemble de renseignements sur les processus en court

- Syntaxe :
`ps [options]`
- Options :
 - -e : affiche des renseignements sur tous les processus en cours
 - -f : (full) information complète
 - --forest : affiche les commandes en hiérarchie
 - -u user : sélectionne les processus de user

http://ltdlmur - 300-Pdf-APPRECHU-28-ENQP-2011-2012

 **La gestion des processus : kill**

kill

Envoie un signal à un processus

- Syntaxe :
`kill -signal pid`
- Options :
 - -signal : correspond au signal envoyé au processus
 - SIGKILL SIGnal KILL : tuer le processus (-9)
 - SIGSTOP SIGnal STOP : stopper le processus
- Description :
La commande kill envoie un signal à un processus. Essentiellement utilisé pour tuer (-9) un processus. `kill -l` liste des signaux.

Exemples :
hans> `kill -9 1635 (kill -SIGKILL 1635)`
Cette commande tue le processus 1635. Le numéro PID est obtenu avec la commande ps

http://ltdlmur - 300-Pdf-APPRECHU-28-ENQP-2011-2012

 **Gestion de l'occupation disque : du**

du [-s][-a][-H] noms

- Affiche le nombre de blocs occupés par les fichiers et répertoires noms .
 - -s : résultats par noms
 - -a : détails de noms et du contenu de noms (récursif)
 - -H : exprimé plus clairement (K et G)

Exemples:
hans> `cd ; du -sH *`
Indique l'occupation disque de chacun de vos répertoires contenus dans votre répertoire d'accueil.
hans>

http://ltdlmur - 300-Pdf-APPRECHU-28-ENQP-2011-2012

 **La sauvegarde : la commande tar**

- Cette commande permet d'archiver des fichiers ou une arborescence de fichiers, c-à-d de les regrouper dans un seul fichier
- Pour créer une nouvelle archive :
`tar -cvf nom_archive.tar repertoire`
- Pour afficher le contenu d'une archive :
`tar -tvf nom_archive.tar`
- Pour extraire les fichiers archivés :
`tar -xvf nom_archive.tar`

Les fichiers sont créés à partir du répertoire courant s'ils sont stockés sur l'archive en chemin relatif.

http://ltdlmur - 300-Pdf-APPRECHU-28-ENQP-2011-2012

La compression

- **compress** : extension .Z
- **gzip** : extension .gz
- **bzip2** : extension bz2

Exemples:

```
hans> compress fichier?
hans> gzip file?
hans > ls
fichier1.Z fichier2.Z file1.gz file2.gz
hans> gunzip file1.gz file2.gz
```

La synchronisation : rsync

rsync permet de synchroniser le contenu de deux fichiers ou répertoires. **rsync** est similaire à la commande **cp** mais ne transfère que les différences entre fichiers. **rsync** est possible entre deux ordinateurs. **rsync** peut être assimilée à une commande permettant de réaliser un miroir entre deux disques.

rsync [options] source destination

- Options :
 - -a : --archive option complète (-rptgoD)
 - --delete : supprime les fichiers destination n'existant plus dans la source
 - -v : affiche le nom et répertoire traités

Exemples:

```
hans> rsync -av cours/ /data1/cours.sav
hans>
```

Arborescence d'UNIX

```

graph TD
    root[" / "] --> bin
    root --> etc
    root --> dev
    root --> home[" /home "]
    root --> lib
    root --> var
    root --> tmp
    root --> usr
    home --> hans
    home --> gretc
    tmp --> st2
    tmp --> st3
    
```

/ : racine (unique)
/bin : commandes et utilitaires (binaire) essentiels au SE
/etc : commandes de gestion et d'administration du système
/lib : bibliothèques de programmes (notamment langage C);
/usr : répertoire pour les commandes plus générales et locales.
/dev : fichiers associés aux périphériques du système
/var : contient des fichiers dont la taille change rapidement (spool files, log files, ...)
/tmp : répertoire des fichiers temporaires;
/home : répertoire de login des utilisateurs du système unix

L'état d'un job : Synthèse

Un job peut se trouver dans 3 états

```

graph TD
    Commande1[commande] --> FG["Foreground  
Avant plan"]
    FG -- fg --> BG["Background  
Arrière plan"]
    BG -- bg --> Suspended["Suspended  
Suspendu"]
    Suspended -- ^Z --> FG
    Suspended -- stop --> BG
    Commande2[commande &] --> BG
    
```

stop = kill -SIGSTOP

Le job control : synthèse

Action	Etat initial du processus	Etat final du processus
hans> commande		processus en avant plan
hans> commande &		processus en arrière plan
hans> fg %numéro_job	processus en arrière plan	processus en avant plan
<ctrl-z>	processus en avant plan	processus stoppé (suspendu)
hans> fg %numéro_job	processus stoppé	processus en avant plan
hans> bg %numéro_job	processus stoppé	processus en arrière plan
• xstra> KILL -STOP %numéro_job • tentative lecture sur terminal • tentative d'écriture sur le terminal (si sity tostep)	processus en arrière plan	processus stoppé