

<p>Notations =====</p> <p>Les commandes et leurs options sont données en minuscules. Les arguments variables (noms de fichiers, etc.) sont donnés en MAJUSCULES.</p> <p>Le prompt du shell est représenté par le symbole "\$".</p> <p>Par exemple, \$ cd RÉPERTOIRE indique que la commande "cd" prend un argument variable "RÉPERTOIRE" qu'il faut remplacer par un répertoire existant...</p> <p>Autre exemple : \$ CMD > FICHER donne une manière de rediriger le résultat d'une commande variable "CMD" (qu'il faut remplacer par une commande existante) dans un fichier dont le nom doit être donné (variable "FICHER").</p> <p>Chemins et fichiers =====</p> <p>fichier caché un fichier dont le premier caractère du nom est le point "."</p> <p>chemin absolu liste des répertoire, en commençant à la racine "/", pour accéder à un fichier ou répertoire Dans le cas d'un chemin de fichier, le nom de fichier se trouve à la fin du chemin.</p> <p>exemples : /home/hyvernats/info202/tp1.html /home/hyvernats/info202/</p> <p>répertoire de travail Tous les processus ont un répertoire de travail ("working directory"). Les chemins relatifs sont relatifs à ce répertoire de travail</p> <p>voir aussi cd (shell), pwd (shell)</p> <p>chemin relatif liste des répertoire, à partir du répertoire de travail, pour accéder à un fichier ou répertoire Dans le cas d'un chemin de fichier, le nom de fichier se trouve à la fin du chemin. Note : un chemin relatif ne commence pas par "/".</p> <p>exemples : info202/tp1.html info202/</p> <p>.. abréviation pour le répertoire parent (en général, du répertoire de travail) Cette abréviation marche dans tous les chemins (absolu ou relatifs).</p> <p>exemple : \$ cd .. permet de changer le répertoire de travail au répertoire parent du répertoire de travail actuel</p> <p>.</p> <p>abréviation pour le répertoire lui même (en général, le répertoire de travail)</p>	<p>Cette abréviation marche dans tous les chemins (absolu ou relatifs).</p> <p>exemple : \$ mv Téléchargement/*.py . permet de déplacer tous les fichiers dont l'extension est ".py" du dossier Téléchargement dans le répertoire de travail</p> <p>~ abréviation pour le dossier personnel de l'utilisateur Cette abréviation ne marche que dans le shell.</p> <p>Motifs shell =====</p> <p>Les motifs shells sont des manières concises de spécifier des listes de fichiers. Ils sont basés sur une notion de "joker" qui permet d'omettre certaines parties des noms de fichiers.</p> <p>* joker représentant une chaîne de caractères arbitraire (potentiellement vide)</p> <p>? joker représentant exactement un caractère arbitraire</p> <p>[...] joker représentant exactement un caractère parmi ceux donnés</p> <p>[!...] joker représentant exactement un caractère qui n'est pas parmi ceux donnés</p> <p>Exemples:</p> <p>*.py liste des fichiers du répertoire de travail dont l'extension est ".py"</p> <p>*.* liste des fichiers du répertoire de travail dont l'extension comporte exactement un caractère</p> <p>*-*.txt liste des fichiers dont l'extension est ".txt" et dont le nom comporte un tiret "-"</p> <p>*[!gv] liste des fichiers dont le dernier caractère du nom n'est ni un "g" ni un "v"</p> <p>* tous les fichiers du répertoire de travail</p> <p>Note : ni le "*" ni le "?" ne peuvent contenir le point initial "." d'un fichier caché.</p> <p>Entrée et sortie standard =====</p> <p>sortie standard L'affichage normal d'un processus se fait sur la sortie standard. Sans aucune redirection, la sortie standard d'une commande shell est le terminal dans lequel elle a été lancée.</p> <p>entrée standard Une commande interactive peut lire des chaînes sur l'entrée standard. Sans</p>
--	--

aucune redirection, l'entrée standard d'une commande est le clavier.

sortie d'erreur

Les messages d'erreurs d'un processus sont normalement affichées sur la sortie d'erreur. Sans aucune redirection, la sortie d'erreur d'une commande shell est le terminal dans lequel elle a été lancée.

Note : sans aucune redirection, on ne peut pas savoir si un affichage est fait sur la sortie standard ou la sortie d'erreur.

Redirections

=====

CMD1 | CMD2

fait agir la commande CMD2 sur le résultat de la commande CMD1

notes :

- si CMD1 n'affiche rien à l'écran, CMD2 n'agit sur rien du tout
- cette redirection ne sert à rien si CMD2 ne lit aucune information sur l'entrée standard
- on peut enchaîner cette redirection
\$ CMD1 | CMD2 | CMD3

exemple :

\$ ls *.py | wc -l
permet de compter le nombre de fichiers du répertoire de travail dont l'extension est ".py"

CMD > FICHIER

redirige le résultat de la commande CMD dans le fichier FICHIER
Attention, si le fichier existait, son contenu est écrasé.

Notes :

- redirige uniquement l'affichage de la sortie standard
- la sortie standard n'est plus affichée sur l'écran

voir aussi : tee (shell)

exemple :

\$ xxd test.pgm > test.hexa
permet de sauvegarder la représentation hexadécimale du fichier "test.pgm" dans le fichier "test.hexa". (Sans redirection, la représentation hexadécimale est simplement affichée à l'écran.)

CMD >> FICHIER

sauvegarde le résultat de la commande CMD à la fin du fichier FICHIER

Notes :

- redirige uniquement l'affichage de la sortie standard
- la sortie standard n'est plus affichée sur l'écran

CMD 2> FICHIER

CMD 2>> FICHIER

redirige l'affichage sur la sortie d'erreur dans le fichier FICHIER

CMD < FICHIER

les lectures sur l'entrée standard de CMD sont effectuées dans le fichier FICHIER.

Attention : si le fichier n'existe pas, cela provoque une erreur et la commande ne s'effectue pas.

Lancer des processus

=====

CMD

lance la commande CMD
Le shell attend que CMD soit terminée pour reprendre la main.

CMD &

lance la commande CMD en tâche de fond
Le shell reprend la main sans attendre que la commande soit terminée.

CMD1 ; CMD2 ; ... ; CMDn

lance successivement les commandes données. Le shell attend qu'une commande soit terminée pour passer à la suivante.

Commandes très importantes

=====

ls

affiche la liste des fichiers et répertoires du répertoire de travail courant

ls FICHIER1 ... FICHIERn

affiche la liste des fichiers donnés en argument
Ceci est surtout utilisé avec les motifs shell :
\$ ls *.py
affiche la liste des fichiers dont l'extension est ".py"

ls RÉPERTOIRE1 ... RÉPERTOIREn

affiche le contenu des répertoires donnés en arguments

La commande "ls" a de nombreuses options :

"-a" ou "-A" affiche aussi les fichiers cachés
"-sh" affiche la taille physique des fichiers
"-S" affiche les gros fichiers en premier
"-Sr" affiche les gros fichiers en dernier
"-l" affiche les méta-données des fichiers

exemple :

\$ ls -sSr
affiche le contenu du répertoire courant dans l'ordre croissant des tailles, avec indication de la taille physique de chaque fichier

cd RÉPERTOIRE

change le répertoire de travail au répertoire donné. (RÉPERTOIRE peut être un chemin absolu ou relatif)

cd

change le répertoire de travail au dossier personnel de l'utilisateur

cd -

change le répertoire de travail au répertoire de travail précédent.
Attention, il n'y a pas d'historique des répertoires de travail :

\$ cd -
\$ cd -
ne change pas le répertoire de travail.

voir aussi : pushd et popd (shell)

pwd

affiche le répertoire de travail
Note : le répertoire de travail est souvent affiché dans le prompt. Cette commande n'est dans ce cas pas très utile.

```

mv FICHER1 ... FICHERn RÉPERTOIRE
déplace les fichiers donnés dans un autre répertoire, sans changer leur nom

mv FICHER CHEMIN
déplace un fichier pour qu'il se retrouve au chemin donné. Si le chemin est un
chemin de répertoire, le fichier ne change pas de nom. Sinon, le nom du fichier
est modifié.

exemple :
$ mv tp3.py OK/
déplace le fichier "tp3.py" dans le répertoire "OK" sans changer son nom
$ mv tp3.py OK/tp3-rendu.py
déplace le fichier "tp3.py" dans le répertoire "OK", en changeant son nom

cp FICHER1 ... FICHERn RÉPERTOIRE
duplique les fichiers donnés dans un autre répertoire, sans changer leur nom

cp FICHER CHEMIN
duplique un fichier pour qu'il se retrouve également au chemin donné. Si le
chemin est un chemin de répertoire, la copie a le même nom que l'original.
Sinon, la copie a le nouveau nom.

exemple :
$ cp tp3.py tp3.py.BACKUP
fait une copie du fichier "tp3.py" dans le répertoire courant en ajoutant
".BACKUP" à la fin du nom

rm FICHER1 ... FICHERn
supprime définitivement les fichiers donnés.
Attention : le shell n'a pas de mécanisme de corbeille !

rm -r FICHER1 ... FICHERn
Supprimer les fichiers et répertoires donnés.

mkdir RÉPERTOIRE
crée un nouveau répertoire
L'option "-p" permet de créer les répertoires intermédiaires.

exemples :
$ mkdir tmp/INFO202
crée le répertoire "INFO202" dans le répertoire existant "tmp". (Si "tmp"
n'existe pas, la commande échoue.)
$ mkdir -p tmp/INFO202/TP3
crée le répertoire TP3 dans "tmp/INFO202". Si certains de ces répertoires
n'existent pas, il sont créés également.

man CMD
affiche la page de manuel de la commande CMD

raccourcis clavier :
q        quitte la page de manuel
Espace   fait défiler une page du manuel
/        fait une recherche dans la page de manuel
n        passe à l'occurrence suivante de la chaîne cherchée

Autres commandes
=====

head FICHER1 ... FICHERn
affiche sur la sortie standard les 10 premières lignes des fichiers
Si aucun fichier n'est donné, agit sur l'entrée standard

```

```

options :
-n N      affiche les N premières lignes (au lieu de 10)
-c N      affiche les N premiers octets

exemple :
$ ls -S | head
affiche la liste des 10 fichiers les plus gros du répertoire de travail

tail FICHERS1 ... FICHERn
affiche sur la sortie standard les 10 dernières lignes des fichiers
Si aucun fichier n'est donné, agit sur l'entrée standard

options :
-n N      affiche les N dernières lignes (au lieu de 10)
-c N      affiche les N derniers octets

wc FICHER1 ... FICHERn
compte les octets, mots et lignes des fichiers donnés
Si aucun fichier n'est donné, agit sur l'entrée standard

options :
-l        compte seulement les lignes

find RÉPERTOIRE CONDITIONS
recherche des fichiers à partir du répertoire donné, en n'affichant que les
fichiers qui vérifient les contraintes.
Note : RÉPERTOIRE est facultatif. S'il n'est pas donné, la recherche se fait à
partir du répertoire de travail.

Les conditions principales sont données par des options supplémentaires :
-name MOTIF      le nom du fichier correspond au motif donné
-iname MOTIF     le nom du fichier correspond au motif donné, sans faire de
différence entre minuscules et majuscules
-type TYPE       le type du fichier correspond à TYPE
                  Les deux types principaux sont
                  f      fichier standard
                  d      répertoire
-size +S         la taille du fichier est supérieure à S (la taille peut
utiliser les unités k, M ou G )
-size -S         la taille du fichier est inférieure à S (la taille peut
utiliser les unités k, M ou G )

D'autres conditions permettent de spécifier la date de modification, le
propriétaire, etc. Voir
$ man find
pour toutes les possibilités.

tree
affiche l'arborescence des fichiers
L'option "-d" permet de n'afficher que les répertoires

ps
affiche les processus en cours d'exécution lancés par le shell
options :
--forest         affiche l'arborescence des processus
-e              affiche tous les processus (pas seulement ceux lancés par
le shell)
-u NOM          affiche les processus lancé par l'utilisateur NOM

pstree
alternative à ps (moins standard, et avec moins d'options): affiche
l'arborescence des processus
options :

```

-p affiche les numéros de processus
-p \$\$ affiche seulement les processus lancés par le shell courant

kill PID1 ... PIDn

envoie le signal d'arrêt aux processus dont les numéros sont donnés

Note : on peut obtenir les numéros de processus grâce à la commande "ps" ou "pstree".

Si le signal d'arrêt n'est pas suffisant, on peut envoyer le signal d'arrêt brutal avec l'option -9

xargs CMD

exécute la commande CMD avec les arguments lus sur l'entrée standard

S'utilise essentiellement avec la commande find.

exemple

\$ find -size 0 -type f | xargs rm

supprime tous les fichiers dont la taille est 0.

grep "CHAINE" FICHIER1 ... FICHIERn

recherche la chaîne donnée dans les fichiers donnés

Chaque ligne contenant la chaîne est affichée sur la sortie standard.

Si aucun fichier n'est donné, agit sur l'entrée standard.

options :

-v affiche seulement les lignes qui ne contiennent pas la chaîne

-i ne différencie pas les minuscules et majuscules

note : "CHAINE" peut être une "expression régulière" pour faire des recherches plus sophistiquées