

La ligne de commande bash

Navigation : Une seule arborescence

pwd : indique le répertoire courant.

ls /un/répertoire : liste le contenu du répertoire /un/répertoire

(note : l'option -al fourni plus d'info)

cd : Change Directory, permet de se déplacer dans son répertoire principal

cd /un/répertoire : permet de se déplacer dans /un/répertoire

↑ : permet de rappeler une commande déjà passée

[ctrl]r : permet de rechercher dans l'historique

Dénomination : Tout est fichier

un/chemin/relatif : on nomme le fichier «relatif» en fonction du répertoire courant : «.» (équivalent à ./un/chemin/relatif)

/un/chemin/absolu : on nomme le fichier «absolu» en fonction de la racine de l'arborescence «/»

(note : l'utilisation des chemin absolu est préférable)

. : le répertoire courant

.. : le répertoire parent

~ : votre répertoire principale (home)

* : une chaîne de caractères quelconque

? : un caractère quelconque

Gestion des variables :

MA_VARIABLE=ma-valeur : définit la variable MA_VARIABLE et lui affecter ma-valeur

export MA_VARIABLE : met cette variable à disposition de tous mes futurs sous-processus

unset MA_VARIABLE : supprime la variable

\$MA_VARIABLE : sera systématiquement remplacé par ma-valeur au moment de l'exécution de la commande

echo \$MA_VARIABLE : affiche ma-valeur

echo '\$MA_VARIABLE' : affiche \$MA_VARIABLE

Commandes de consultation :

cat fic1 fic2 : renvoie sur la sortie la concaténation des fichiers fic1 et fic2

(note : bzcata et gzcata font de même mais décompresse le fichier à la volée)

more fic1 : affiche sur la sortie le contenu du fichier fic1 page par page

less fic1 : affiche sur la sortie le contenu du fichier fic1 page par page

(note : /chaîne permet de rechercher la chaîne de caractères chaîne)

grep une-chaîne /un/fichier : retourne sur la sortie toutes les lignes d' /un/fichier qui contiennent une-chaîne

file /un/fichier : retourne le format de fichier d'/un/fichier

type une-commande : retourne comment le shell va

interpréter une-commande

Gestion des fichiers :

mkdir ~/un/répertoire : crée ~/un/répertoire

rmdir ~/un/répertoire : supprime ~/un/répertoire (s'il est vide)

touch ./un/fichier : crée un fichier ou change sa date de modification

mv ~/un/fichier un-autre : déplace/renomme le fichier ./un/fichier vers ./un-autre

cp ~/un/fichier ~/un/autre : copie le fichier un/fichier vers un/autre

cp ~/un/fichier un/autre un/répertoire : copie les fichiers un/fichier et un/autre vers un/répertoire

(note : l'option -p préserve les droits sur le fichier **cp -p**)

rm ~/un/fichier : supprime le fichier ~/un/fichier (note : l'option -rf permet de forcer la suppression récursive d'une sous-arborescence)

Archive/compression :

tar cvf une/archive.tar des fichiers à mettre dedans : crée une/archive.tar contenant les fichiers et sous-arborescences : ./des, ./fichiers, ./a, ./mettre et ./dedans

tar xvf une/archive.tar : extrait une archive.tar dans le répertoire courant

tar xvf une/archive.tar un-fichier : extrait un-fichier d'une/archive.tar

(note : les options c, x, v et f signifient Create eXtract Verbose File)

tar xvf - : extrait une archive lue sur entrée standard.

Exemple : bzcata une-archive.tar.bz2 | tar xvf -

gzip un/fichier : compresse le fichier un/fichier et le remplace par le fichier compressé un/fichier.gz

bzip2 un/fichier : idem au format bz2 avec le fichier un/fichier.bz2

gunzip un/fichier.gz : décompresse le fichier

bunzip2 un fichier.bz2 : avec le format bz2

tar zxvf une/archive.tgz : décompresse au format gzip puis extrait l'archive

tar jcvf une/archive.tar.bz2 liste fichiers : crée l'archive compressée au format bz2

Gestion des droits :

-rwxr-x--- : lecture, écriture et exécution pour le propriétaire, lecture et exécution pour les membres du groupe, aucun droits pour les autres

drwxr-x--- : Lister les fichiers, créer un fichier et passer dans ce répertoire pour le propriétaire, lister les fichiers et passer dans ce répertoire pour les membres du groupe, aucun droits pour les autres

chmod ugo+rwx : donne les droits rwx aux User, Group, Other

(note : commande à ajuster suivants les droits à positionner avec + pour ajouter et - pour supprimer puis omettre les caractères non souhaité parmi rwxugo)

\$?: le code retour de la dernière commande lancée en avant plan

Gestion des processus :

Présentation :

lancé depuis un autre processus (en général le shell), par défaut il hérite :

des arguments passés sur la ligne de commande

de la sortie standard (flux 1 : l'écran terminal)

de la sortie d'erreur (flux 2 : l'écran terminal)

de l'environnement (les variables exportées par le shell)

de l'entrée standard (flux 0 : le clavier terminal)

en fin de traitement, il retourne un code retour au processus parent : 0 pour exécuté avec succès, toute autre valeur pour signaler une erreur.

Lancement :

cmd arg : exécute la commande cmd avec l'argument arg en avant-plan

cmd arg1 arg2 & : exécute la commande en arrière plan

nohup cmd arg1 arg2 & : lance la commande en la détachant du terminal courant

Pendant l'exécution d'une commande :

[Ctrl]c : Arrête le processus en avant plan

[Ctrl]z : interrompt le processus

bg : le fait repartir en arrière-plan

jobs : liste les traitements en arrière-plan

fg 4 : remet le processus 4 en avant plan

wait : attend la fin de tous les traitements lancés en arrière-plan

kill -X 1234 : transmet le signal X au processus dont le pid est 1234.

Gestion des entrées sorties :

une-commande >> un/fichier : redirige la sortie standard d'une-commande vers un/fichier

une-commande > un/fichier : redirige la sortie standard d'une-commande vers un/fichier en l'écrasant

une-commande | une-autre-commande : redirige la sortie standard d'une-commande vers l'entrée d'une-autre-commande

une-commande 2> un/fichier : écrase ou crée un/fichier et y écrit la sortie d'erreur d'une-commande

une-commande >> un/fichier 2>&1 : ajoute à la fin d'un/fichier la sortie standard (>>) et redirige la sortie d'erreur vers la sortie standard (2>&1)

Variable internes du shell :

\$* : la ligne de commande

\$0 : l'appel de l'exécutable

\$1 .. \$9 .. \$n : les arguments

\$# : le nombre d'arguments

\$\$: le pid courant

\$! : le pid de la dernière commande lancée en arrière-plan

Editeur vi :

vi un/fichier : édite avec vi un/fichier

Présentation :

l'éditeur vi permet d'éditer du texte depuis un terminal.

Il est très puissant mais nécessite un apprentissage. Une fois l'éditeur lancé, plusieurs modes d'utilisation sont possibles. Vim est une version plus souple : elle contient notamment un tutoriel accessible *via* la commande vimtutor.

Mode navigation : mode permettant de naviguer dans le texte et de passer des commande de gestion de texte.

(note : c'est le mode par défaut, la touche [esc] permet de retourner dans ce mode)

h j k l : gauche, bas, haut et droite

x : supprime le caractère courant

dd : supprime la ligne courante

d\$ ou **D** : supprime tout jusque la fin de ligne

yy : copie la ligne courante

yw : copie le mot courant

cw : remplace le mot courant

p : paste, colle à partir du curseur

(note : ces commandes peuvent être modifié par un quantificateur

y4y : copie 4 ligne, **c3w** change 3 mots ...)

ma : créé le marqueur **a** sur le caractère courant

Mode insertion : mode d'ajout de texte, on entre dans ce mode avec les commandes de gestion de texte :

i : à la place du curseur

a : après le curseur

A : en fin de ligne

o : au début de la ligne suivante alors créée

Mode commande : mode permettant de passer des commandes à l'éditeur de texte. On entre dans ce mode depuis le mode navigation avec la touche :

:w : write, sauvegarde le fichier

:q! : quitte (! permet de forcer l'action)

:wq! : sauvegarde et quitte en forçant

:x : sauvegarde et quitte

:set nu : set active l'option de numérotation des lignes

:set nonu : désactive la numérotation des lignes

(note : un grand nombre d'options sont disponibles)

:a,bs/toto/tata/gc : du marqueur a jusqu'au marqueur b

(a,b), substitue la chaîne «toto» par la chaîne «tata»

(/toto/tata/) pour toutes les occurrences de chaque lignes (g) en demandant une confirmation (c)

:a,bmc : déplace (m) le texte compris entre les marqueurs a et b au niveau du marqueur c

:a,bd : supprime (d) le texte compris entre les marqueurs a et b

La ligne de commande bash

(note : a, b et c peuvent être remplacés par :
rien : pour la ligne courante
42 : un numéro de ligne

+5 : une référence de numéro de ligne par rapport à la ligne courante)