

Commandes Bash

Bash (Bourne-Again SHell) est un shell Linux.

Raccourcis clavier

Tab	Auto-complétion.
Ctrl-c	Envoie le signal SIGINT au processus courant (kill).
Ctrl-d	Envoie EOF.

Caractères spéciaux

?	Remplace un seul caractère
*	Remplace 0 ou plusieurs caractères
[]	Remplace par n'importe quel caractère situé à l'intérieur des crochets
[-]	Remplace par un caractère qui se situe dans la suite (0-9 de 0 à 9, a-z de a à z)
[!]	Le contraire de []
' " "	Chaine dont le contenu sera interprété
' '	Chaine dont le contenu sera interprété
#	Commentaire
\$a	Contenu de la variable a

Groupement de commandes

commande1 ; commande2	Lance commande2 après commande 1
commande1 commande2	Renvoie le résultat de commande1 à commande2
commande &	Lance commande en arrière-plan
commande1 commande2	SINON : ne lance commande2 que si commande1 échoue
commande1 && commande2	ET : Lance commande2 si commande1 réussi
(liste)	Groupe de commandes
{ liste; }	Groupe de commandes
\$(commande)	Substitution de commandes

Les substitutions de variables

\$variable ou \${variable}	valeur de \$variable
\${#variable}	Nombre de caractères de \$variable
\${variable:-mot}	Renvoie sa valeur si variable existe et est non nulle, sinon renvoie mot
\${variable:+mot}	Renvoie mot si la variable existe et est non nulle, sinon retourne 0
\${variable:=mot}	Affecte mot à variable si elle est nulle
\${variable:?mot}	Affiche mot si variable n'existe pas, et termine la commande ou le script
\${variable#expression}	Si expression = debut de \$variable alors supprime la partie reconnue et retourne le reste
\${variable##expression}	Si expression = debut de \$variable alors supprime la plus grande partie reconnue et retourne le reste
\${variable%expression}	Si expression = fin de \$variable alors supprime la partie reconnue et retourne le reste
\${variable%%expression}	Si expression = fin de \$variable alors supprime la plus grande partie reconnue et retourne le reste

Conditions

[test]	Une condition
-d fichier	Vrai si le fichier existe et est un répertoire
-e fichier	Vrai si le fichier existe
-f fichier	Vrai si le fichier est un fichier ordinaire
-h fichier	Vrai si le fichier existe et est un lien symbolique
-r fichier	Vrai si le fichier existe et si l'utilisateur a le droit de le lire
-s fichier	Vrai si le fichier existe et à une taille supérieure à 0 (non-vidé)
-w fichier	Vrai le fichier existe et si l'utilisateur a le droit d'y écrire
-x fichier	Vrai si le fichier existe et peut être exécuté par l'utilisateur
fichier1 -nt fichier2	Vrai si fichier1 est plus récent (par rapport à la date de modification) que fichier2
fichier1 -ol fichier2	Vrai si fichier1 est plus vieux (par rapport à la date de modification) que fichier2
[-n] chaine	Vrai si la chaine n'est pas vide
-z chaine	Vrai si le chaine est vide
chaine1 = chaine2	Vrai si les deux chaines sont identiques
chaine1 != chaine2	Vrai si les deux chaines sont différentes
entier1 -eq entier2	Vrai si entier1 est égal à entier2
entier1 -ge entier2	Vrai si entier1 est plus grand ou égal à entier2
entier1 -gt entier2	Vrai si entier1 est supérieur à entier2
entier1 -le entier2	Vrai si entier1 est plus petit ou égal à entier2
entier1 -lt entier2	Vrai si entier1 est plus petit que entier2
entier1 -ne entier2	Vrai si entier1 est différent de entier2

Structures de controles

if condition; then command1; commande2; ... else autrecommande1; autrecommande2; ... fi	SI condition ALORS ... SINON ... FIN_SI
for i in liste; do command1; commande2; ... done	POUR i DANS liste FAIT ... FIN_FAIT
until condition; do command1; commande2; ... done	JUSQUACEQUE condition FAIT ... FIN_FAIT
while condition; do command1; commande2; ... done	TANTQUE condition FAIT ... FIN_FAIT

Scripts

Pour écrire un script, il suffit d'éditer un fichier, d'entrer des commandes, puis de le rendre exécutable (chmod u+x fichier).	
#!/bin/bash echo "Hello World"	Le script classique, le shebang (sharp bang) indiquant l'interpréter à utiliser

Les variables de paramètres

\$0	Nom du script en cours
\$1, \$2, \$3 ...	Argument du script en cours
\$#	Nombre d'arguments
\$	Tous les paramètres (sous la forme d'une unique chaine)
\$@	Tous les paramètres (sous la forme d'une liste itérable)
\$\$	PID du script
\$!	PID du dernier processus shell lancé en arrière-plan
\$?	Code retour de la dernière commande

Calculs

let b=a+1	On peut faire des calculs en utilisant l'instruction let (attention aux espaces)
-----------	--

Redirections

0	Entree Standart (clavier)
1	Sortie Standard (écran)
2	Sortie Erreur Standard (écran)
> fichier	Redirection en sortie vers un fichier, écrase le fichier si le fichier existe
< fichier	L'entrée standard est un fichier
>> fichier	Redirection en sortie vers un fichier, sans écrasement de fichiers
<< mot	Lecture au clavier jusqu'à mot
<> fichier	L'entrée et la sortie standard sont un fichier

Commandes usuelles d'Unix

man commande	Obtenir de l'aide sur la commande commande
man -k mot_cle	Chercher de l'aide sur le mot clé mot_cle
ls -la	Liste détaillée (-l) de tous les fichiers (-a) du répertoire courant
cd repertoire	Se déplacer dans le repertoire
pwd	Obtenir le chemin absolu courant
mkdir	Créer un repertoire
rm fichier et rm -r repertoire	Supprime
cp source destination	Copier
mv source destination	Déplace
cat fichier1 fichier2	Concatène les contenus de fichier1 et fichier2 (s'il n'y a qu'un seul fichier, cela revient à l'afficher)
more fichier	Afficher le contenu de fichier sous forme paginée (navigation)
commande more	Paginer le résultat de commande (navigation)
commande > fichier	Rediriger la sortie de commande dans le fichier
seq	Affiche des suites de nombres
grep chaine	Recherche chaine sur l'entrée standard
sed -e 's/avant/après/g'	Remplacer avant par après sur l'entrée standard
clear	Efface l'écran
echo	Visualise une chaine ou variable
ps -A	Obtenir la liste des processus en cours
kill -9 pid	Arrêter le processus d'identifiant pid (attention : sans sécurité)
who -s	Utilisateurs connectés et leur station
su utilisateur	Changer d'utilisateur
exit	Se délogger
shutdown	Arrêt de la machine
at	Planifier l'exécution d'une commande
crontab	Planifier la répétition de l'exécution d'une commande

Autres commandes utiles

read variable	Attend une saisie au clavier et l'affecte à variable
date	Renvoie la date
cal	Calendrier
whoami	donne le nom de login de l'utilisateur courant
hostname	Nom de la station sur laquelle on est connecté
uname	Nom de l'utilisateur courant
logname	Nom de login courant
id	renvoie les UID et GID de l'utilisateur courant
env	Visualise les variables d'environnement
whereis	Localise une commande sur les disques
apropos	Recherche des rubriques connexes de l'aide
touch	Crée un fichier vide ou met à jour les flags d'un fichier
df	Donne l'utilisation des disques
find	Trouver un fichier

Tris et actions d'extraction

sort	Tri lexicographique
uniq	Élimine les lignes identiques consécutives
cut	Extraction d'une partie d'un fichier, colonnes, caractères, ...
tr -s 'a'	Remplace plusieurs occurrences consécutives de a par une seule.
paste	Colle ligne à ligne deux fichiers
wc	Compte le nombre de lignes, mots, caractères

Principes de gestion des fichiers et répertoires

Sous UNIX, tout est fichier. Un périphérique, un écran, une imprimante, un répertoire, tout est fichier. Un fichier peut être exécutable si son flag d'exécution x est positionné. Un répertoire est un fichier qui contient la liste des fichiers sensés être stockés dans ce répertoire.	
Un fichier est constitué de deux parties : l'inode qui est le descripteur du fichier (secteurs, numéro d'inode, ...) et des données. Lorsqu'un fichier est copié, il y a copie de l'inode et des données. Lorsqu'un lien physique est créé, seule l'inode du fichier est dupliqué car les données elles-même sont partagées.	
ln source destination	Créer un lien physique (logique -s)

Droits courants des fichiers et répertoires

Tout fichier/répertoire a une série de bits définissant les droits d'accès (r lire, w écrire, x exécuter/parcourir) :				
	type	user	group	others
répertoires	d	rwX	rwX	rwX
fichiers	-	rwX	rwX	rwX
chmod u+x fichier	Ajoute les droits d'exécution à l'utilisateur sur fichier			
chmod 777 fichier	Place tous les droits à tout le monde sur fichier			

Fichiers divers liés au terminal (les noms changent suivant la version)

.bashrc	Exécuté au démarrage d'un terminal
.profile ou .bash_profile ou .login	Exécuté au démarrage d'une session sur un terminal
.logout	Exécuté à la fermeture d'une session sur un terminal