

## GESTION DES FICHIERS

<b>pwd</b>	chemin du répertoire courant (print working directory)
<b>cd</b>	change directory
.	répertoire courant
..	répertoire parent
/	root
~	répertoire personnel (home)
~user	répertoire personnel d'un utilisateur
-	répertoire précédent
<b>ls</b>	contenu d'un répertoire (list)
-a	fichiers cachés (all)
-l	liste détaillées (long)
-i	i-nœud
-R	récurivement
-F	format des fichiers : @ lien; * exe;   pipe;
<b>file fic</b>	détermine le type du fichier
<b>touch fic</b>	création d'un fichier
<b>mkdir rep</b>	création d'un répertoire
<b>rmdir rep</b>	suppression d'un répertoire vide
<b>rm fic</b>	suppression d'un fichier
-r	supprimer un répertoire non vide
-i	confirmation (interactif)
-f	force les permissions (default)
<b>mv src dest</b>	déplace ou renomme un fichier
<b>cp src dest</b>	copie physique d'un fichier
<b>ln src dest</b>	lien physique
-s	lien symbolique
<b>whereis fic</b>	recherche un fichier
<b>find rep</b>	recherche de fichiers selon
-name fic	nom
-inum n	i-nœud
-perm n	droits d'accès
-type [fdbclps]	type
-size ±n	taille ± n
-user uid	utilisateur
-group gid	group (gid)
-print	affiche les résultats (défaut)
-exec cmd {} \;	exécute la commande
-ok cmd {} \;	exécute la cmd avec confirmation
<b>cat fic</b>	affiche le contenu d'un fichier
<b>more fic</b>	affiche sur plusieurs pages
return	ligne suivante
espace ou f	page suivante (forward)
b	début du fichier (bottom)
q	quitter (quit)
/texte	cherche le texte
n	occurrence suivante (next)
=	numéro de la ligne courante
<b>cmp fic1 fic2</b>	compare deux fichiers (sur 2 colonnes)
-s	statut : 0 égal; 1 diff; 2 erreur;
<b>diff fic1 fic2</b>	affiche les différences (lignes à lignes)
<b>tar fic</b>	archive d'un fichier
cvf	créer une archive
rvf	rajoute des fichiers
tvf	table des entrées de l'archive
xvf	extrait le contenu
<b>compress fic</b>	compression d'un fichier (extension .z)
<b>uncompress fic.z</b>	décompression d'un fichier

## DROITS D'ACCES

<b>chmod n fic</b>	change les droits d'accès
r	read (4)
w	write (2)
x	execution (1)
u±=rwx	propriétaire (user)
g±=rwx	groupe
o±=rwx	autres (others)
a±=rwx	tous (all)
ex : <b>chmod 751</b>	u=7=rwx, g=5=r-x, o=1=-x
<b>umask n</b>	modifie les droits d'accès par défaut
<b>chown u fic</b>	change le propriétaire
<b>chgrp g fic</b>	change le groupe

## REDIRECTIONS DE FLUX

< fic	stdin (0) à partir d'un fichier
> fic	stdout (1) vers un fichier (RAZ)
>> fic	stdout (1) vers un fichier (cat)
2> ou 2>>	stderr (2) vers un fichier
cmd	tube (pipe) vers une commande

## EXECUTION D'UN PROCESSUS

<b>cmd &amp;</b>	exécution arrière-plan
<b>nohup cmd &amp;</b>	no hang up
<b>nice -n cmd</b>	baisse priorité processus
<b>ps</b>	liste des processus (process)
-e	tous les utilisateurs
-l	plus de détails (long)
-f	ordre spécifique
-u user	processus d'un utilisateur
<b>kill n pid</b>	envoie un signal au processus
-3	quitte (ctrl+f)
-9	tue radicalement (ctrl+c)
-15	arrêt normal (défaut)
-l	liste des signaux disponibles
<b>which cmd</b>	affiche le chemin de la commande

## FILTRES

<b>sort fic</b>	trie les lignes d'un fichier
-f	pas de casse
-d	ordre du dico
-n	ordre numérique
-r	décroissant
-u	unique (suppr occurrence double)
-tcar +n	trie champ n délimité par car
<b>grep txt fic</b>	recherche ligne dans un texte
-n	numéros des lignes
-i	pas de casse
-v	ne contenant pas
<b>wc fic</b>	compte le nombre de
-l	lignes (lines)
-w	mots (words)
-c	caractères (char)
<b>head -n fic</b>	affiche n 1 <sup>ère</sup> lignes d'un fichier
<b>tail ±n</b>	affiche de la ligne n à la fin/début
<b>paste fic1 fic2</b>	affichage multi-colonnes
<b>cut fic</b>	extraction de colonnes
-dcar -fn	extrait champ n délimité par car
-cn	extrait colonne n
A-B	de A à B
A-	de A jusqu'à la fin
B+	du début jusqu'à B
A,B	A et B
<b>tr fic txt1 txt2</b>	remplace txt1 par txt2

## SYSTEME

<b>news</b>	nouvelles du système (/usr/news)
<b>id</b>	affiche le uid et gid
<b>who</b>	liste des utilisateurs connectés
-T	autorisation des messages
-R	nom de la machine
<b>su user</b>	change d'identité
exit	revenir identité principale
<b>passwd</b>	change le mot de passe
<b>mount dev rep</b>	montage d'une unité logique
<b>umount rep</b>	libère la liaison avec l'unité logique
<b>chsh user path</b>	change le shell de l'utilisateur

## DIVERS

<b>man cmd</b>	aide sur la commande
n	section : cmd (1), fct sys (2), fct lib C (3)
<b>echo "txt"</b>	affiche un texte à l'écran
\$VAR	affiche la valeur d'une variable
-n	empêche le retour chariot
<b>clear</b>	efface les lignes du terminal
<b>banner "txt"</b>	texte en grosses lettres
<b>date</b>	affiche la date
+"%d/%m/%y %H:%M:%S"	
<b>cal</b>	affiche un calendrier
<b>write uid</b>	envoie un message
ctrl+d	fin du message
! cmd	ligne interprétée par le shell
<b>mesg</b>	réception des messages
y/n	autorise ou interdit
<b>history</b>	liste des commandes utilisées
<b>r</b>	rappel la dernière commande
-n	numéro n de l'historique
text	commençant par texte
txt=txt2	en remplaçant txt par txt2
<b>alias nom=cmd</b>	définie nouvelle commande abrégée
<b>unalias nom</b>	détruit l'alias

**METACARACTERES**

*	chaîne quelconque
?	caractère quelconque
[abc]	caractère égal à a, b ou c
[A-B]	caractère compris entre A et B
[!0-9]	caractère différent d'un chiffre

**VARIABLES****affectation**

**VAR=txt** affectation (sans espace autour =)  
**\$VAR=\$(PWD)** copie de variable

**évaluation**

**\$((expr))** évalue la chaîne (avec : +-\*/\*)  
**\${#VAR}** longueur de la chaîne VAR

**affichage**

**echo \$VAR** affiche le contenu de VAR  
**echo "\$VAR"** idem  
**echo \${VAR}** idem

**variables script**

**\$0** nom du script  
**\$n** valeur de l'argument *n* (*n* < 10)  
**\${n}** valeur de l'argument *n* (si *n* > 10)  
**\$#** nombre d'arguments (sans compter \$0)  
**\$\*** globalité des arguments (sauf \$0)  
**\$\$** PID : identificateur du processus (tq le shell)  
**\$?** résultat de la dernière cmd synchrone  
**!** résultat de la dernière cmd asynchrone &  
**set** liste de toutes les variables  
**env** liste des variables de l'environnement  
**setenv var val** affecte une valeur à une variable environnement  
**unset var** détruit une variable  
**export var** sauvegarde la variable

**SCRIPTS BASH**

**#!/bin/bash** 1<sup>ère</sup> ligne : précise l'interpréteur utilisé tq *bash*  
**read var** stocke l'entrée clavier dans *var*

**condition si** **condition selon**  
 if [ *condition* ] ; then *instructions* case *var* in  
 elif [ *condition* ] ; then *instructions* val) *instructions* ;;  
 else *instructions* \*) *instructions* ;;  
 fi esac

**boucle for** **tant que/jusqu'à**  
 for *var* in *liste de valeurs* while/until [ *condition* ]  
 do *instructions* do *instructions*  
 done done

**logique** -o (or) ; -a (and)

**test de nombres**  
 -eq equal (=)  
 -ne not equal (!=)  
 -gt great than (>)  
 -lt less than (<)  
 -ge great equal (>=)  
 -le less than (<=)

**test de chaînes**  
 -z *\$var* si *var* = chaîne vide (zero)  
 -n *\$var* si *var* = chaîne non vide (not zero)  
 txt1=txt2 égalité

**test de fichiers**  
 -f *fic* si fichier ordinaire  
 -d si répertoire (directory)  
 -r -w -x si droit en lecture/écriture/exécution  
 -s si existant

**opérateurs**  
*cmd1* || *cmd2* si *cmd1* renvoie vrai, alors exécute *cmd2*  
*cmd1* && *cmd2* si *cmd1* renvoie faux, alors exécute *cmd2*

**COMPILATIONS C**

**gcc fic.c** compilation de *fic.c* en tant qu'exécutable *a.out*  
 -o *nom* spécifie le nom de l'exécutable  
 -c simple compilation (*fic.o*) sans *link*  
 -I*rep* spécifie le répertoire contenant les includes  
 -lm indique l'utilisation de la librairie <math.h>

**EDITEUR DE TEXTE VI**

**vi fic** édite le fichier texte  
 +*n* commence à la ligne *n*  
 -*r* récupération du fichier après un crash

**commandes** mode de contrôle du texte  
 [ESC] pour accéder au mode commande  
 flèches déplace le curseur (pas tjrs bien configuré)  
 [espace] → avance à droite  
 h ← recule à gauche  
 [ctrl+n] ↑ monte d'une ligne  
 [ctrl+p] ↓ descend d'une ligne  
 [ctrl+b] # monte d'une page  
 [ctrl+f] # descend d'une page  
 nG va à la ligne *n*  
 r remplace le caractère sous le curseur  
 x supprime un caractère  
 d\$ efface jusqu'à la dernière ligne  
 dd efface la ligne courante  
 /*txt* cherche la prochaine occurrence de *txt*  
 ?*txt* cherche la précédente occurrence de *txt*  
 :w [enter] écrit le fichier  
 :x [enter] écrit le fichier puis quitte vi  
 :q! [enter] quitte vi sans sauvegarder  
 !!*cmd* [enter] exécute *cmd* (shell) sans quitter vi

**insertion** mode d'édition du texte  
 [i] édite avant le curseur  
 [I] édite en début de ligne  
 [A] édite en fin de ligne