

Christophe Blaess

Commandes du shell

Ecriture de scripts shell

Ce document a été réalisé sur système Linux avec les logiciels libres :

- **Formation+** pour préparer le cours complet,
- **L^AT_EX** pour produire les transparents et le support de cours imprimé,
- **Xfig** pour les graphiques vectoriels,
- **The Gimp** pour les images et les photos.

Support de cours : version 15

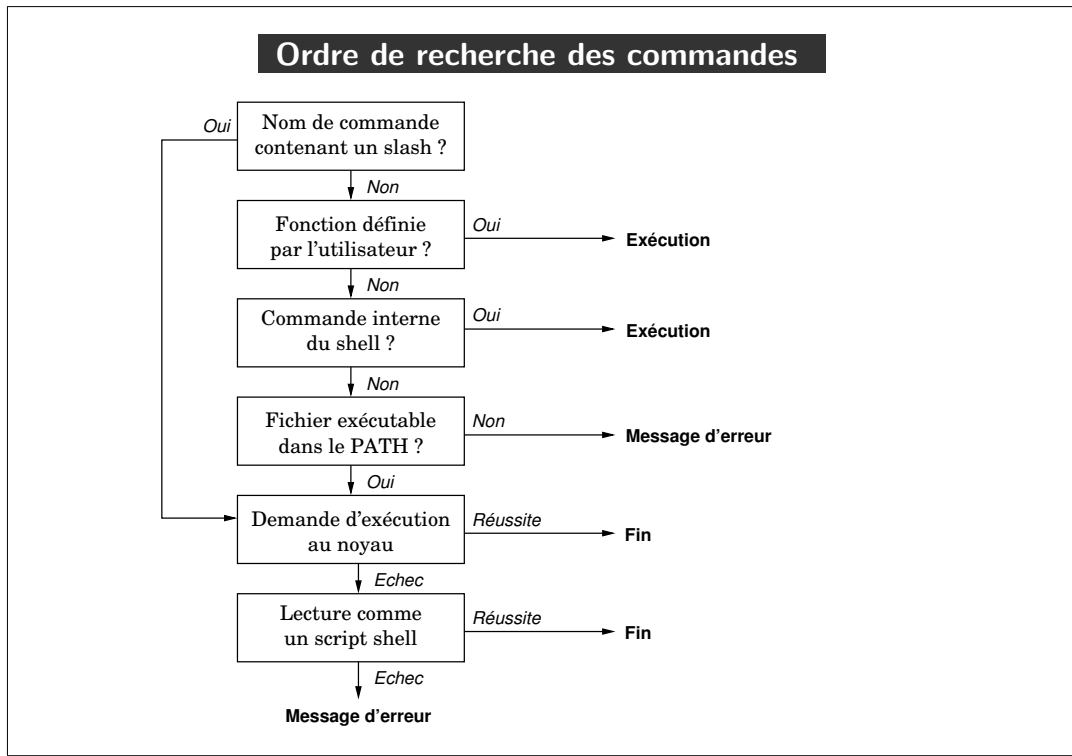
© CHRISTOPHE BLAESS 2001-2006 – Tous droits réservés

Aucune partie de ce cours ne peut être reproduite ou transmise à quelque fin ou par quelque moyen que ce soit, électronique ou mécanique, sans l'autorisation expresse et écrite de l'auteur.

*Ecriture de scripts shell***Commandes du
shell**

Christophe Blaess

– Exécution des commandes	3
– Ordre de recherche des commandes	3
– Exécution de commande binaire dans un processus fils	6
– Exécution de commande binaire dans le même processus	7
– Exécution de script dans un processus fils	8
– Exécution de script dans le même processus	9
– Commandes essentielles pour les scripts	10
– Entrées-sorties	10
– Interactions avec le système	14
– Configuration du shell	16
– Arguments en ligne de commande	17
– Commandes Unix standards	18
– Manipulation des fichiers	18
– Contenu des fichiers	20
– Interactions avec le système	22
– Travaux pratiques	23
– Exercices	23



	sh	bash	ksh	SUSv3
.	*	*	*	*
:	*	*	*	*
[*	*	*	*
alias		*	*	*
bg		*	*	*
bind		*		
break	*	*	*	*
builtin		*	*	
cd	*	*	*	*
command		*	*	*
continue	*	*	*	*
declare		*		
echo	X	*	*	*
enable		*	X	
eval	*	*	*	*
exec	*	*	*	*
exit	*	*	*	*
export	*	*	*	*
false	X	X	*	*
fc		*	*	*
fg		*	*	*
getopts	*	*	*	*
hash	*	*	*	*
jobs		*	*	*
kill	X	*	*	*
let		*	*	
local		*	*	
print			*	
printf		*	X	*
pwd	*	*	*	*
read		*	*	*
readonly	*	*	*	*
return	*	*	*	*
set		*	*	*
shift	*	*	*	*
source		*		
test	*	*	*	*
times	*	*	*	*
trap	*	*	*	*
true	X	X	*	*
type		*	*	*
typeset		*	*	
ulimit		*	*	*
umask	*	*	*	*
unalias		*	*	*
unset	*	*	*	*
wait		*	*	*
whence			*	

* = commande interne, X = commande externe

La commande `type` indique si son argument est une fonction, un alias, une commande interne (*builtin*) ou une commande externe.

L'option `-a` permet de connaître toutes les réponses possibles pour une commande.

À vous...

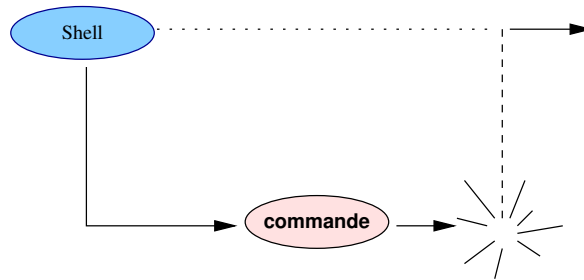
```
$ type ls
$ type -a ls
```

```
$ type echo
$ type -a echo
```

```
$ type mkdir
$ mkdir azerty
$ type mkdir
```

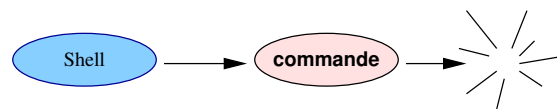
Exécution de commande binaire dans un processus fils

Appel classique, recherche dans le PATH, exécution par le noyau
\$ commande



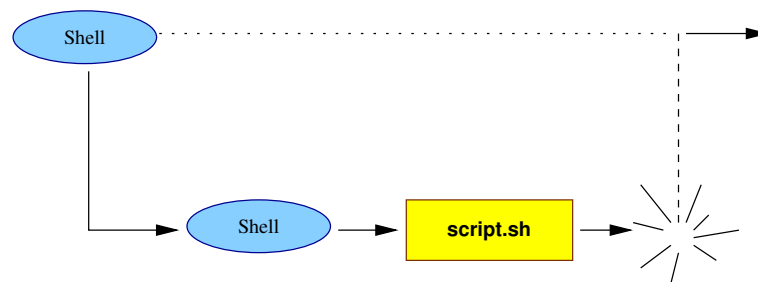
Exécution de commande binaire dans le même processus

Remplacement du processus shell par la commande invoquée
`$ exec commande`



Exécution de script dans un processus fils

Invocation explicite de l'interpréteur

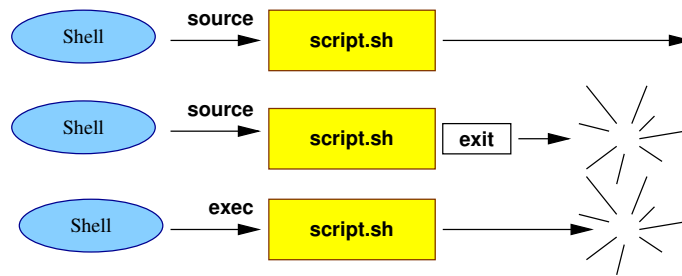
`$ sh script.sh`Invocation implicite (par la ligne *shebang*)`$./script.sh`

Exécution de script dans le même processus

Interprétation dans le même processus

`$. script.sh``$ source script.sh`

Remplacement du shell courant

`$ exec ./script.sh`

Entrées-sorties

Les entrées-sorties pour un script shell peuvent se faire avec diverses commandes.

Affichage de données :

- **echo** : affichage classique de message,
- **cat** : affichage du contenu d'un fichier ou document en-ligne.
- **printf** : mise en forme d'arguments numériques, chaînes, etc.

Saisie de données :

- **read** : lecture d'une ligne de texte,
- **cat** : lecture d'un bloc de texte.

echo`echo [options] arguments...`

affiche les arguments séparés par un espace, avec un saut-de-ligne final.

Options		Séquences spéciales	
<code>-n</code>	pas de saut-de-ligne final	<code>\a</code>	sonnerie (<i>alarm</i>)
<code>-E</code>	ignorer les séquences spéciales	<code>\b</code>	retour en arrière (<i>backspace</i>)
<code>-e</code>	traiter les séquences spéciales	<code>\c</code>	pas de saut-de-ligne final
		<code>\f</code>	saut-de-page (<i>form feed</i>)
		<code>\n</code>	saut-de-ligne (<i>new line</i>)
		<code>\r</code>	début de ligne (<i>carriage return</i>)
		<code>\t</code>	tabulation horizontale
		<code>\v</code>	tabulation verticale
		<code>\\</code>	caractère <i>backslash</i>
		<code>\nnn</code>	caractère Ascii <i>nnn</i> (octal)

Les options de `echo` n'étant pas disponibles dans toutes les implémentations, il est parfois nécessaire de faire appel à `printf(1)`.

```
$ echo abc      def
$ echo "abc    def"
```

```
$ echo -n "message"
```

```
$ echo "\007"
$ echo -E "\007"
$ echo -e "\007"
```

```
$ while true; do d=$(date +%X); echo -en "\r$d"; done
```

```
$ echo "erreur" >&2
$ (echo "erreur" >&2) 2>/dev/null
```

cat

```
cat [options] fichiers. . .
```

copie le contenu des fichiers indiqués sur sa sortie standard.

-n	numéroter les lignes affichées
-b	numéroter les lignes non-blanches
-v	afficher les caractères spéciaux

Exemples. . .

```
cat <<- FIN_MESSAGE
Ceci sera affiché sur
la sortie standard du processus
en respectant les sauts-de-ligne
FIN_MESSAGE
```

```
saisie=$(cat)
```

read`read variables...`

`read` lit une ligne depuis l'entrée standard, et remplit les *variables* avec les mots successifs de la ligne.

La dernière *variable* accueille tout le reste de la ligne.

Si aucune variable n'est indiquée, `read` place le contenu de la ligne lue dans la variable `REPLY`.

`read` utilise les caractères contenus dans la variable `IFS` comme séparateurs pour l'extraction des mots.

Le code de retour de `read` est nul (vrai) tant que la lecture réussit, et non-nul (faux) en cas d'erreur ou de fin de fichier.

```
$ read mot1; echo "mot1=$mot1"  
(entrez une phrase)  
$ read m1 m2 s; echo "m1=$m1 m2=$m2 suite=$s"  
(entrez une phrase)  
$ read; echo "REPLY=$REPLY"  
(entrez une phrase)
```

```
$ cat /etc/passwd  
(Observez la construction du fichier)  
$ IFS=': '  
$ while read nom mdp uid reste  
> do echo "$nom = $uid"  
> done < /etc/passwd
```

Interactions avec le système

`exec redirections...`

La commande `exec` sans argument applique les redirections indiquées au shell en cours.

La notation `n>&-` ferme le descripteur n . Il ne sera plus accessible.

A vous...

```
$ exec >&-  
$ ls  
$ exec 2>&-  
ls  
exec 2> /dev/tty  
exec > /dev/null  
ls  
exec > /dev/tty  
ls
```

cd

```
cd répertoire
```

`cd` modifie le répertoire de travail du processus pour celui indiqué en argument.

pwd

La commande `pwd` affiche le répertoire de travail du processus sur la sortie standard.

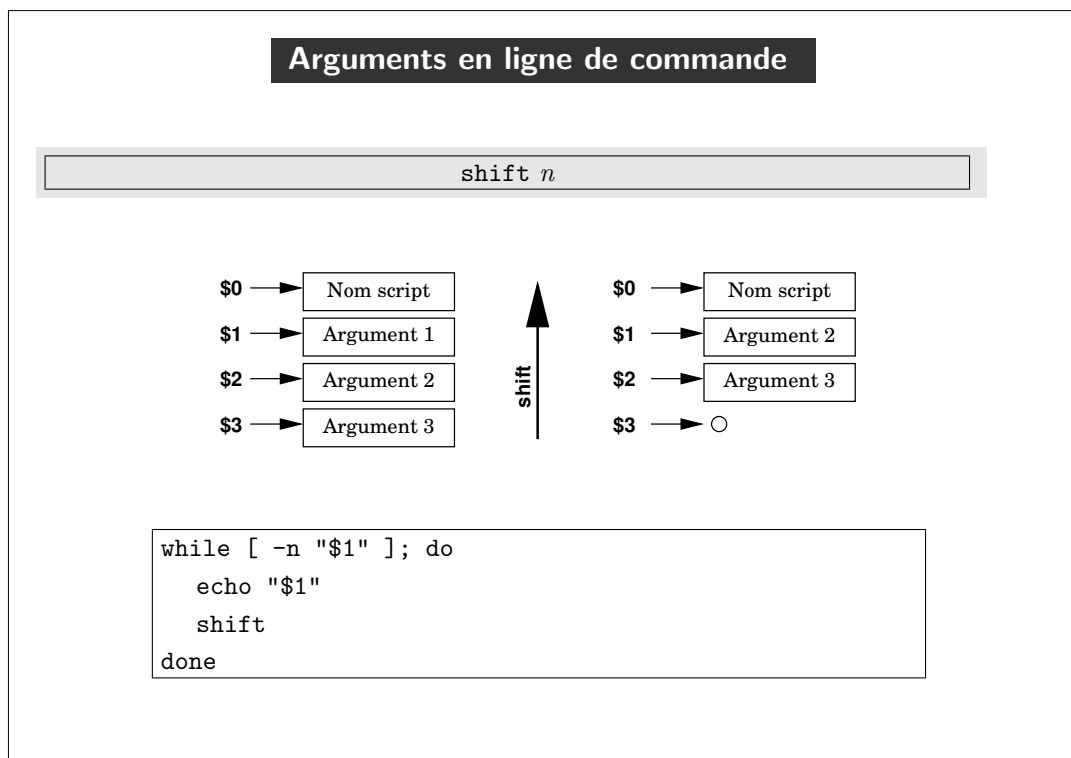
PWD

La variable `PWD` est mise à jour à chaque déplacement dans un répertoire.

La modification de la variable `PWD` est autorisée, mais elle ne change pas le répertoire de travail du processus.

Configuration du shell*set option*

-a	-o allexport	exporter automatiquement les variables lors des affectations
-b	-o notify	signaler les terminaisons des jobs en arrière-plan
-C	-o noclobber	ne pas écraser de fichier lors des redirections
-e	-o errexit	arrêter le script en cas d'échec de commande simple
-f	-o noglob	ne pas développer les noms de fichiers
-m	-o monitor	afficher les codes de terminaison des jobs
-n	-o noexec	vérifier la syntaxe sans exécuter les commandes
-u	-o nounset	refuser la lecture d'une variable indéfinie
-v	-o verbose	afficher les lignes de commande avant exécution
-x	-o xtrace	afficher les développements avant exécution
	-o vi	utiliser le mode vi d'édition des commandes
	-o emacs	utiliser le mode emacs d'édition des commandes



Commandes Unix standards

Manipulation des fichiers

ls [*options*] [*fichiers...*]
Affiche noms et informations sur les fichiers indiqués.

- **-a** : aussi les fichiers commençant par un point
- **-d** : noms des répertoires, pas le contenu
- **-i** : affiche les numéros d'i-noeud
- **-l** : format d'affichage long
- **-R** : affiche récursivement les sous-répertoires

basename *fichier* [*suffixe*]
Elimine le chemin d'accès et le suffixe éventuel du fichier.

dirname *fichier*
Supprime le nom de fichier d'un chemin d'accès complet.

chown [-R] *nom[:groupe]* *fichiers...*
Modification du groupe et du propriétaire d'un fichier.

- **-R** : modifie récursivement les sous-répertoires

chgrp [-R] *groupe* *fichiers...*
Modification du groupe d'un fichier.

- **-R** : modifie récursivement les sous-répertoires

chmod [-R] *mode* *fichiers...*
Modification les permissions d'un fichier.

- **-R** : modifie récursivement les sous-répertoires

cp [*options*] *fichier nouveau*
cp [*options*] *fichiers... répertoire*
Copie de fichier.

- **-p** : conserve horodatages, propriétaire, permissions
- **-d** : copie les liens symboliques en tant que tels
- **-R** : copie récursivement les sous-répertoires

mv [*options*] *fichier nouveau*
mv [*options*] *fichiers... répertoire*
Renomme ou déplace des fichiers.

- **-f** : force l'écrasement du fichier destination

rm [*options*] *fichiers...*
Suppression de fichiers.

- **-f** : pas de confirmation
- **-i** : confirmation avant effacement
- **-r** : effacement récursif des sous-répertoires

touch [*options*] *fichiers...*
Accès en écriture sur des fichiers.

- **-t** *MMJJhhmm* : utiliser l'horodatage indiqué
- **-r** *fichier* : utiliser l'horodatage du fichier

ln [*options*] *fichier nouveau*
ln [*options*] *fichiers... répertoire*
Création de liens sur des fichiers.

- **-s** : liens symboliques
- **-f** : force l'écrasement si la destination existe

```
mkdir [options] répertoires...  
Création de répertoires.  
- -p : créer si besoin les répertoires parents  
- -m mode : fixe les autorisation d'accès
```

```
rmdir [options] répertoires...  
Suppression de répertoires vides.  
- -p : efface le répertoire parent s'il devient vide
```

```
df [options] fichiers...  
Affiche la place occupée sur un système de fichiers.  
- -P : affiche une ligne d'en-tête  
- -k : affiche les tailles en kilo-octets
```

```
du [options] fichiers...  
Affiche la place occupée par des fichiers  
- -a : affiche les statistiques pour les fichiers  
- -c : affiche seulement le total  
- -x : ignore sous-répertoires sur un autre système de fichiers
```

```
dd [options]  
Copie générique et conversion de fichiers  
- if= : nom du fichier d'entrée  
- of= : nom du fichier de sortie  
- bs= : taille des blocs  
- count= : nombre de blocs à copier  
- skip= : position de début de lecture  
- seek= : position de début d'écriture
```

```
compress fichiers...  
Compression au format .Z
```

```
zip fichiers...  
Compression au format .zip
```

```
gzip fichiers...  
Compression au format .gz
```

```
bzip2 fichiers...  
Compression au format .bz2
```

```
tar [options] [fichiers...]  
Création et extraction d'archives .tar  
- -f fic : nom de l'archive  
- -c : créer une archive  
- -t : afficher le contenu d'une archive  
- -x : extrait le contenu d'une archive  
- -z : invoque gzip pour les (dé)compressions  
- -j : invoque bzip2 pour les (dé)compressions  
- -v : mode volubile
```

Contenu des fichiers

cat [*options*] [*fichiers...*]
Concatène et affiche les fichiers indiqués.
– **-n** : numérotter les lignes en sortie
– **-v** : afficher symboliquement les caractères spéciaux

more [*options*] [*fichier*]
Affiche page par page son entrée standard (**less** est plus puissant).

head [*options*] [*fichier*]
Affiche le début d'un fichier.
– **-c n** : afficher les *n* premiers octets
– **-n n** : afficher les *n* premières lignes

tail [*options*] [*fichier*]
Affiche la fin d'un fichier.
– **-n** : afficher les *n* dernières lignes
– **-f** : affiche en continu les ajouts

wc [*options*] [*fichier*]
Affiche le nombre de caractères, de mots et de lignes d'un fichier.

cmp [*options*] *fichier1 fichier2*
Compare des fichiers.
– **-l** : afficher le rang des octets différents
– **-s** : n'affiche rien, renvoie vrai ou faux

diff [*options*] *fichier1 fichier2*
Détermine les différences entre des fichiers.
– **-i** : ignorer les différences majuscules/minuscules
– **-b** : ignorer les différences sur les blancs
– **-r** : analyse récursivement les sous-répertoires
– **-u** : fournit un format compatible avec **patch**

patch [*options*] [*fichier*]
Applique un fichier de différences obtenu avec **diff**.

sort [*options*] [*fichier*]
Trier les lignes d'un fichier de texte.

od [*options*] [*fichier*]
Affiche le contenu d'un fichier binaire.
– **-c** : affiche les caractères Ascii imprimables
– **-x** : affiche les codes en hexa-décimal

strings [*options*] [*fichier*]
Cherche les chaînes Ascii dans un fichier.

file [*fichier*]
Détermine le type d'un fichier.

uniq [*options*] [*fichier*]
Recherche les répétitions de lignes dans un fichier de texte.
– **-u** : n'affiche que les lignes uniques
– **-d** : n'affiche que les lignes dupliquées
– **-c** : affiche le nombre d'occurrences des lignes

tr [*options*] [*fichier*]
Transpose ou élimine les caractères d'un ensemble.

col [*options*] [*fichier*]
Élimine les séquences de surimpression.
– **-b** : enlève tous les retours en arrière

cut [*options*] [*fichier*]
Extrait des portions de lignes dans un fichier.
– **-b** : affiche les caractères indiqués
– **-f** : affiche les champs indiqués
– **-d** : caractère séparateur de champs

csplit [*options*] *fichier motif*
Découpe un fichier de texte en plusieurs parties selon le contexte
– **-f** : préfixe pour les fichiers créés

split [*options*] *fichier*
Découpe un fichier en plusieurs parties.
– **-l n** : fichiers de *n* lignes
– **-b n** : fichiers de taille *n*

pr [*options*] *fichiers...*
Prépare des fichiers de texte pour l'impression.
– **-h texte** : indique l'en-tête de chaque page
– **-l n** : *n* lignes par page
– **-t** : supprime en-têtes et pieds-de-pages

lp [*options*] *fichiers...*
Envoie des fichiers au système d'impression.
– **-d imprimante** : sélectionne l'imprimante
– **-n** : nombre de copies

mail [*options*] *destinataire* < *fichier* >
Envoie un message par courrier électronique.
– **-s texte** : sujet du message
– **-c liste** : destinataires en CC
– **-b liste** : destinataires en BCC

Interactions avec le système**at** [*options*]

Programme un job à exécuter ultérieurement.
– **-f script** : lit les commandes dans un fichier

batch [*options*]

Programme un job à exécuter lorsque la charge système sera faible.

crontab [*options*]

Edite le fichier des commandes à exécuter régulièrement.

- **-e** : édite le fichier **crontab** personnel
- **-l** : affiche son contenu actuel
- **-r** : supprime le fichier **crontab** personnel

ps [*options*]

Affiche l'état des processus en cours.

- **-ax** : tous les processus (*BSD*)
- **-u** : informations complètes (*BSD*)
- **-e** : tous les processus (*SysV*)
- **-f** : informations complètes (*SysV*)

sleep *duree*

Attend une durée déterminée.

uname [*options*]

Affiche des informations sur le système.

- **-m** : type de machine
- **-n** : nom d'hôte
- **-r** : version du système
- **-s** : système d'exploitation
- **-a** : toutes les informations

who [*options*]

Affiche la liste des utilisateurs connectés.

id [*options*] *nom*

Affiche les identifiants d'un utilisateur appelant.

date [*options*]

Affiche ou configure la date système.

cal [*options*]

Affiche un calendrier.

Travaux pratiques**Exercices****Exercice numéro 1**

Saisie d'une réponse de l'utilisateur

Exercice numéro 2

Choix d'une option

Exercice numéro 3

Traitement d'une liste de fichiers

Exercice numéro 4

Minuscules et majuscules

Exercice numéro 5

Lecture ligne à ligne du contenu d'un fichier

Exercice numéro 6

Analyse d'une chaîne de caractères

Exercice numéro 7

Automatisation de transfert FTP

Exercice numéro 8

Processus créés par les commandes Unix

Exercice numéro 9

Arithmétique avancée avec le shell

Index

\$0 (paramètre)	17	ln(1)	18
\$1 (paramètre)	17	local (commande)	4
alias (commande)	4	lp(1)	21
at(1)	22	ls(1)	5, 18
basename(1)	18	mail(1)	21
batch(1)	22	mkdir(1)	5, 19
bg (commande)	4	more(1)	20
bind (commande)	4	mv(1)	18
break (commande)	4	od(1)	20
builtin (commande)	4	patch(1)	20
bzip2(1)	19	pr(1)	21
cal(1)	22	print (commande)	4
cat(1)	10, 12, 20	printf (commande)	4
cd (commande)	4, 15	ps(1)	22
chgrp(1)	18	pwd (commande)	4, 15
chmod(1)	18	PWD (variable)	15
chown(1)	18	read (commande)	4, 10, 13
cmp(1)	20	readonly (commande)	4
col(1)	21	REPLY (variable)	13
command (commande)	4	return (commande)	4
compress(1)	19	rm(1)	18
continue (commande)	4	rmdir(1)	19
cp(1)	18	set (commande)	4, 16
crontab(1)	22	shift (commande)	4, 17
csplit(1)	21	sleep(1)	22
cut(1)	21	sort(1)	20
date(1)	22	source (commande)	4
dd(1)	19	split(1)	21
declare (commande)	4	strings(1)	20
df(1)	19	tail(1)	20
diff(1)	20	tar(1)	19
dirname(1)	18	test (commande)	4
du(1)	19	times (commande)	4
echo (commande)	4, 5, 10, 11	touch(1)	18
enable (commande)	4	tr(1)	21
eval (commande)	4	trap (commande)	4
exec (commande)	4, 7, 14	true (commande)	4
exit (commande)	4	type (commande)	4, 5
export (commande)	4	typeset (commande)	4
false (commande)	4	ulimit (commande)	4
fc (commande)	4	umask (commande)	4
fg (commande)	4	unalias (commande)	4
file(1)	20	uname(1)	22
getopts (commande)	4	uniq(1)	20
gzip(1)	19	unset (commande)	4
hash (commande)	4	wait (commande)	4
head(1)	20	wc(1)	20
id(1)	22	whence (commande)	4
IFS (variable)	13	who(1)	22
jobs (commande)	4	zip(1)	19
kill (commande)	4		
let (commande)	4		