

Système Le shell bash

Gilles Roussel
<http://igm.univ-mlv.fr/~rousseau/L22/>

Licence 2

6 février 2011

Les interpréteurs de commandes ou *shell*

- Interface non graphique entre le système et l'utilisateur : très proche des fonctionnalités systèmes
- Programme qui permet d'en lancer d'autres
- Disponible sur la plupart des systèmes
- Une syntaxe différentes pour chaque système
- Sous Unix/Linux au moins deux syntaxes `sh` et `csh`
 - `bash` utilise la syntaxe `sh`
- Attaché à un terminal en mode graphique ou à une console (`Ctrl+Alt+F1`)

La ligne de commande

- Chaque ligne entrée est interprétée par le shell
- Le premier mot de la ligne est interprété comme une commande
- Les mots suivants comme les arguments de la commande
- Chaque ligne est mémorisée dans un historique
 - Les flèches `↑` et `↓` permettent de naviguer dans l'historique
 - `Ctrl+r` permet de rechercher une commande dans l'historique

Le répertoire courant

- Un répertoire courant (*working directory*) est associé au shell
- Toutes les commandes lancées à partir du shell auront le même répertoire courant
- Il est accessible *via* la commande `pwd` (*print working directory*)
- La commande `cd` (*change working directory*) permet de changer de répertoire courant
- `mkdir` et `rmdir` permettent de modifier l'arborescence

Les variables

Système

Gilles Roussel

Shell

La ligne de commande

Environnement

Caractères spéciaux

Quelques commandes

Le chaînage

Ecrire un shell script

Entrées-sorties

Gestion des options

Il existe deux types de variables :

- les variables locales qui ne sont accessibles que dans le shell
- les variables d'environnement qui seront aussi accessibles par les programmes lancés depuis du shell

Gilles Roussel

Système

Les variables locales

Système

Gilles Roussel

Shell

La ligne de commande

Environnement

Caractères spéciaux

Quelques commandes

Le chaînage

Ecrire un shell script

Entrées-sorties

Gestion des options

- Elles sont définies et modifiées par **VAR=...**
TEST=pwd
- Leur valeur est obtenue par **\$VAR**
echo \$TEST
pwd
(la commande **echo** affiche les valeurs en arguments)
- La valeur d'une variable peut être utilisée n'importe où
\$TEST
affiche le répertoire courant (⇔ pwd)

Gilles Roussel

Système

Les variables d'environnement

Système

Gilles Roussel

Shell

La ligne de commande

Environnement

Caractères spéciaux

Quelques commandes

Le chaînage

Ecrire un shell script

Entrées-sorties

Gestion des options

- Une variable locale peut devenir d'environnement en l'exportant
- Exportation au moyen de la commande **export**
export TEST
- Les variables d'environnement sont passées aux programmes lancés *via* le shell
- La commande **env** affiche l'ensemble des variables d'environnement
env
...

Gilles Roussel

Système

Apostrophes

Système

Gilles Roussel

Shell

La ligne de commande

Environnement

Caractères spéciaux

Quelques commandes

Le chaînage

Ecrire un shell script

Entrées-sorties

Gestion des options

- Une chaîne de caractères entre apostrophes (**'**) n'est pas interprétée par le Shell
echo 'TEST=' \$TEST
\$TEST= pwd
- Une chaîne de caractères entre apostrophes à l'envers, *backquote* (**`**), est interprétée par le shell et sa valeur est égale au résultat de la commande
echo `TEST`
/home/roussel
 - Les *backquotes* peuvent être remplacées en bash par **\$()** pour plus de lisibilité
echo \$(TEST)
/home/roussel

Gilles Roussel

Système

Les caractères spéciaux du shell

Système

Gilles Roussel

Shell

La ligne de commande

Environnement

Caractères spéciaux

Quelques commandes

Le chaînage

Ecrire un shell script

Entrées-sorties

Gestion des options

- Certains caractères sont interprétés de façon particulière par le shell
 - `*` remplace n'importe quelle suite de caractères, éventuellement vide, dans les noms de fichiers
 - `t*` $\Leftrightarrow$ tous les noms de fichiers commençant par `t`
 - `?` remplace zéro ou un caractère dans les noms de fichiers
 - `nom.txt?` $\Leftrightarrow$ tous les noms de fichiers commençant par `nom.txt` suivi éventuellement d'un caractère
- Ces caractères ne sont pas interprétés s'ils sont placés entre apostrophes (`'`) ou guillemets (`"`)
 - Entre guillemets les variables sont remplacées par leur valeur

Gilles Roussel

Système

Les commandes

Système

Gilles Roussel

Shell

La ligne de commande

Environnement

Caractères spéciaux

Quelques commandes

Le chaînage

Ecrire un shell script

Entrées-sorties

Gestion des options

- Il existe de nombreuses commandes qui permettent, en particulier, de manipuler les fichiers
- Souvent plus simple et plus rapide que d'utiliser des outils graphiques

Gilles Roussel

Système

La commande `man`

Système

Gilles Roussel

Shell

La ligne de commande

Environnement

Caractères spéciaux

Quelques commandes

Le chaînage

Ecrire un shell script

Entrées-sorties

Gestion des options

- Permet, en particulier d'accéder au manuel des commandes
- Organisé en sections

1	commandes shell
2	appels systèmes
3	fonctions de bibliothèque
4	les fichiers spéciaux et périphériques
5	les formats de fichier
6	les jeux
7	divers
- Option `-k` permet de rechercher une commande à partir d'un mot clef
 - # `man -k image`

Gilles Roussel

Système

La commande `cat`

Système

Gilles Roussel

Shell

La ligne de commande

Environnement

Caractères spéciaux

Quelques commandes

Le chaînage

Ecrire un shell script

Entrées-sorties

Gestion des options

- Affiche le contenu des fichiers dont les noms sont passés en arguments
- Sans argument, affiche tout ce qui est lu sur l'entrée standard (le clavier) et l'affiche
- Les commandes `more` ou `less` ajoute une gestion de page

Gilles Roussel

Système

La commande ls

- Système**
- Gilles Roussel
- Shell
- La ligne de commande
- Environnement
- Caractères spéciaux
- Quelques commandes
- Le chaînage
- Ecrire un shell script
- Entrées-sorties
- Gestion des options
- Liste le contenu des répertoires passés en argument
 - Sans argument, liste le contenu du répertoire courant
 - Avec l'option **-l** affiche des informations sur les fichiers

```
# ls -l exemple
-rwxr-xr-x 1 roussel institut 244 mar 14 2003 exemple
```

Droits relatifs au **possesseur** du fichier **roussel** Droits relatifs au **groupe** du fichier **institut** Droits du **possesseur** du fichier (lecture, écriture, exécution) Droits des membres du **groupe** du fichier (lecture, exécution) Droits des **autres** utilisateur (lecture, exécution)
 - Avec l'option **-R** liste récursivement les fichiers dans les répertoires

Gilles Roussel Système

Les commandes grep et find

- Système**
- Gilles Roussel
- Shell
- La ligne de commande
- Environnement
- Caractères spéciaux
- Quelques commandes
- Le chaînage
- Ecrire un shell script
- Entrées-sorties
- Gestion des options
- **grep** affiche les lignes lues depuis l'entrée standard (le clavier) ou un fichier, qui contiennent une chaîne de caractères particulière
 - Option **-v** permet d'afficher celle qui ne la contienne pas
 - Caractères spéciaux dans la chaîne :
 - **^** : début de ligne
 - **\$** : fin de ligne

```
# grep '^$'
```
 - **find** recherche dans tous les fichiers à partir d'un répertoire particulier
 - **-name** précise la forme des noms recherchés
 - **-type d** précise que l'on recherche des répertoire

```
# find /home/roussel -type d -name 'LI*'
```

Gilles Roussel Système

Les commandes file, cut et wc

- Système**
- Gilles Roussel
- Shell
- La ligne de commande
- Environnement
- Caractères spéciaux
- Quelques commandes
- Le chaînage
- Ecrire un shell script
- Entrées-sorties
- Gestion des options
- **file** affiche le type du contenu d'un fichier
 - **cut** permet de récupérer un morceau d'une chaîne de caractères tabulée
 - l'option **-d** précise le délimiteur
 - l'option **-f** précise le numéro du champs
 - **wc** (*word count*) compte, entre autres, le nombre de caractères, de mots et de ligne dans un fichier
 - **-c** : nombre de caractères
 - **-w** : nombre de mots
 - **-l** : nombre de lignes
 - **-L** : longueur de la plus longue ligne

Gilles Roussel Système

La commande ps

La commande **ps** avec les options **ax** permet de connaître les programmes qui s'exécutent et leur identité

```
# ps ax
PID TTY          STAT       TIME COMMAND
  1 ?            S          0:00 init [5]
...
2874 ?          S          0:00 [pccardd]
...
7285 ?          Sl         0:14 gnome-terminal
7287 pts/1      Ss         0:00 bash
...
```

Gilles Roussel Système

Le chaînage de commandes

Système

Gilles Roussel

Shell

La ligne de commande

Environnement

Caractères spéciaux

Quelques commandes

Le chaînage

Écrire un shell script

Entrées-sorties

Gestion des options

- Le point virgule (;) chaîne deux commandes, la commande à droite commence uniquement quand celle de gauche à terminée
- le esperluette (&) chaîne deux commandes, la commande à droite commence même si celle à gauche n'est pas terminée
- Le *pipe* (|) chaîne deux commandes, la sortie de la commande à gauche est envoyée sur l'entrée de la commande à droite

```
# ls -l | cut -d ' ' -f 2
```

Gilles Roussel

Système

Écrire un shell script

Système

Gilles Roussel

Shell

La ligne de commande

Environnement

Caractères spéciaux

Quelques commandes

Le chaînage

Écrire un shell script

Entrées-sorties

Gestion des options

- Si le fichier commence par **#!**, le nom du fichier est passé au programme dont le nom suit pour être interprété

```
#!/bin/grep test
```

```
test de programme idiot  
qui fait une recherche sur son contenu
```

- Il suffit donc de préciser le nom d'une commande shell à utiliser pour que les lignes du fichier soient exécutées comme si elles étaient tapées

```
#!/bin/bash
```

```
ls -l  
echo $PWD  
exit 0
```

Gilles Roussel

Système

Changer les droits

Système

Gilles Roussel

Shell

La ligne de commande

Environnement

Caractères spéciaux

Quelques commandes

Le chaînage

Écrire un shell script

Entrées-sorties

Gestion des options

La commande chmod (man chmod)

```
# chmod go-x exemple  
# ls -l exemple  
-rwxr--r-- 1 roussel institut 244 mar 14 2003 exemple  
# chmod g+wx exemple  
# ls -l exemple  
-rwxrwxr-- 1 roussel institut 244 mar 14 2003 exemple  
# chmod u=rw exemple  
# ls -l exemple  
-rw-rwxr-- 1 roussel institut 244 mar 14 2003 exemple
```

Retire les droits d'**exécution** pour le **groupe** Retire les droits d'**exécution** pour les **autres** Ajoute les droits d'**écriture** pour le **groupe** Ajoute les droits d'**exécution** pour le **groupe** Affecte **exactement** les droits de **lecture** et **écriture** pour le **possesseur**

Gilles Roussel

Système

Changer les droits

Système

Gilles Roussel

Shell

La ligne de commande

Environnement

Caractères spéciaux

Quelques commandes

Le chaînage

Écrire un shell script

Entrées-sorties

Gestion des options

u	possesseur (<i>user</i>)	+	ajout	r	lecture	
g	groupe	-	suppression	w	exécution	
o	autre (<i>other</i>)	=	affectation	x	exécution	
a	tout le monde (<i>all</i>)					

existe aussi une notation en hexadécimal pour les droits :

Notation hexadécimale

r=4 w=2 x=1

Par exemple :

```
chmod 0666 ⇔ chmod a=rwa chmod 0750 ⇔ chmod a=rx;  
chmod g-w; chmod o-rwx
```

Gilles Roussel

Système

Arguments, terminaison et options en Shell

Système

Gilles Roussel

Shell

La ligne de commande

Environnement

Caractères spéciaux

Quelques commandes

Le chaînage

Écrire un shell script

Entrées-sorties

Gestion des options

- Arguments de la ligne de commande disponibles dans `$0`, `$1`, ...
 - `$0` contient le nom du script
- `$#` contient le nombre d'arguments
- `$*` et `@$` contiennent tous les arguments à partir de `$1`
 `$*` les regroupe dans une unique chaîne de caractères
- `exit` permet de terminer
- `$?` retourne la valeur de retour de la dernière commande lancée

Gilles Roussel

Système

Un peu de syntaxe

Système

Gilles Roussel

Shell

La ligne de commande

Environnement

Caractères spéciaux

Quelques commandes

Le chaînage

Écrire un shell script

Entrées-sorties

Gestion des options

- `if commande; then commandes; else commandes; fi`
exécute *commande* et si elle termine (`exit`) en renvoyant 0 les commandes de `then` sont exécutées, sinon les autres
- `for var do commandes; done`
exécute *commandes* avec la variables *var* prenant les valeurs des arguments (`$1` , `$2` , ...)
- `for var in values; do commandes; done`
exécute *commandes* avec la variables *var* prenant les valeurs de *values*
- `case value in val) commandes;; ... esac`
exécute *commandes* si *value* est égale à *val*
- `help` permet d'obtenir une documentation sur la syntaxe

Gilles Roussel

Système

Faire des tests

Système

Gilles Roussel

Shell

La ligne de commande

Environnement

Caractères spéciaux

Quelques commandes

Le chaînage

Écrire un shell script

Entrées-sorties

Gestion des options

- La commande `test` ou les symboles `[]` et `[[]]` permettent d'effectuer des tests utilisés avec `if`
 - Si un fichier existe ou s'il est exécutable

```
# if test -f cours.pdf; then echo oui; else echo non; fi
# if [[ -x cours.pdf ]]; then echo oui; else echo non; fi
```
 - Si deux chaîne de caractères est égale à une autre

```
# if [ $A == "abc" ]; then echo oui; else echo non; fi
```
 - Si deux valeurs entières sont égales

```
# if [[ $A -eq 2-1 ]]; then echo oui; else echo non; fi
```

Gilles Roussel

Système

Variables d'environnement et shell script

Système

Gilles Roussel

Shell

La ligne de commande

Environnement

Caractères spéciaux

Quelques commandes

Le chaînage

Écrire un shell script

Entrées-sorties

Gestion des options

- Un script ne peut pas servir à définir des variables d'environnement dans le shell
- Possibilité de faire exécuter une suite de commandes d'un fichier directement par le shell
 - Le fichier n'a pas besoin d'être exécutable
- Utilisation de la commande `source` ou point `(.)` devant le nom du fichier

```
# . .bashrc
```

Gilles Roussel

Système

Définir des fonctions

Système

Gilles Roussel

Shell

La ligne de commande

Environnement

Caractères spéciaux

Quelques commandes

Le chaînage

Écrire un shell script

Entrées-sorties

Gestion des options

- Il est possible de définir des fonctions avec le mot clef **function**
- Une fonction se comporte comme un shell script, en particulier concernant les arguments, si ce n'est que les variables d'environnement sont visibles

```
# function test () { for i;do echo "$i".jpeg; export V=1; done; }
# test a b
a
b
# echo $V
1
```

Gilles Roussel

Système

Lecture du clavier

Système

Gilles Roussel

Shell

La ligne de commande

Environnement

Caractères spéciaux

Quelques commandes

Le chaînage

Écrire un shell script

Entrées-sorties

Gestion des options

- Il est possible de lire une ligne entrée par un utilisateur avec la commande interne **read**
- Place les valeurs des mots lus dans les variables en argument
 - Si plus de mots que de variables la dernière contient la fin de la ligne
 - Si moins de mots que de variables les dernières variables sont vides
- ```
read a b
test test1 test2
echo $a
test
echo $b
test1 test2
```

Gilles Roussel

Système

## Manipulation du contenu d'une variable

Système

Gilles Roussel

Shell

La ligne de commande

Environnement

Caractères spéciaux

Quelques commandes

Le chaînage

Écrire un shell script

Entrées-sorties

Gestion des options

- **\${i#mot}** supprime **mot** s'il apparaît au début de la valeur de la variable **i**
- **\${i%mot}** supprime **mot** s'il apparaît à la fin de la valeur de la variable **i**
- Utile pour manipuler les noms de fichiers :

```
for i in *.jpeg; do mv $i ${i%.jpeg}.jpg; done
```

Gilles Roussel

Système

## Entrée et sorties standards

Système

Gilles Roussel

Shell

La ligne de commande

Environnement

Caractères spéciaux

Quelques commandes

Le chaînage

Écrire un shell script

Entrées-sorties

Gestion des options

- Chaque processus démarre avec certains **descripteurs standards** ouverts
- Descripteurs repérés par des numéros ou un nom symbolique
  - Entrée standard : **stdin** et **0**
  - Sortie standard : **stdout** et **1**
  - Sortie d'erreur : **stderr** et **2**
- Descripteurs standards, en général, associés au terminal courant (fenêtre Shell)
  - Tout ce qui est tapé dans la fenêtre est envoyé sur l'entrée standard du programme
  - Tout ce qui est écrit sur les sorties par le programme est affiché dans la fenêtre

Gilles Roussel

Système

## Redirections

Système

Gilles Roussel

Shell

La ligne de commande

Environnement

Caractères spéciaux

Quelques commandes

Le chaînage

Ecrire un shell script

Entrées-sorties

Gestion des options

- Il est possible de rediriger les entrées-sorties
- Vers des fichiers ou un autre descripteur avec `>` et `<`
  - `# ls / >fichier`  
Redirige la sortie standard de `ls -l` vers le fichier `fichier`
  - `# grep tmp <fichier`  
Envoie le contenu du fichier `fichier` sur l'entrée standard de la commande `grep tmp`
  - `# grep -r tmp / >fichier 2>/dev/null`  
Redirige la sortie standard de la commande `grep -r tmp` vers le fichier `fichier` et la sortie d'erreur vers le fichier `/dev/null` (trou noir)
  - `# grep -r tmp / >fichier 2>&1`  
Redirige la sortie standard vers le fichier `fichier` et la sortie d'erreur vers la sortie standard (le fichier)
- Entre commandes avec `|` (*pipe*)

Gilles Roussel

Système

## Gestion des options

Système

Gilles Roussel

Shell

La ligne de commande

Environnement

Caractères spéciaux

Quelques commandes

Le chaînage

Ecrire un shell script

Entrées-sorties

Gestion des options

- `shift` permet de décaler d'un argument (sauf `$0`)
- `set -- $VAR` permet de remplacer les arguments par le contenu de `VAR`
- `getopt` permet la gestion des options sur la ligne de commande

Gilles Roussel

Système

## Gestion des options

Système

Gilles Roussel

Shell

La ligne de commande

Environnement

Caractères spéciaux

Quelques commandes

Le chaînage

Ecrire un shell script

Entrées-sorties

Gestion des options

```
#!/bin/bash

function usage {
 echo "usage: $(basename $0) [-a] [-b arg] arg"
 exit 1
}

args=$(getopt ab: $*)
if ["$?" != 0]; then
 usage
fi
echo "$*"
set -- $args
echo "$*"
opts="Options:"
for i do
 case "$i" in
 -a) shift;opts="$opts -a";;
 -b) shift;opts="$opts -b avec argument $1";shift;;
 --) shift;break;
 esac
done
if ["$#" != 1]; then
 usage
else
 echo "$opts"
 echo "Arguments: $*"
fi
exit 0
```

Gilles Roussel

Système

## Arguments, terminaison et options en Shell (bash)

Système

Gilles Roussel

Shell

La ligne de commande

Environnement

Caractères spéciaux

Quelques commandes

Le chaînage

Ecrire un shell script

Entrées-sorties

Gestion des options

```
./getopt_test.sh -a -barg1 arg
-a -barg1 arg
-a -b arg1 -- arg
Options: -a -b avec argument arg1
Arguments: arg
./getopt_test.sh arg1 arg2 -a -b arg
arg1 arg2 -a -b toto
-a -b arg -- arg1 arg2
Options: -a -b avec argument arg
Arguments: arg1 arg2
./getopt_test.sh -a -barg1
-a -barg1
-a -b arg1 --
usage: getopt_test.sh [-a] [-b arg] arg
```

Gilles Roussel

Système