CS230 - Programmation shell sous Unix

Cours 1 – Interprétation des commandes

Dr. Bassem DEBBABI

Objectif du cours 1

Utilisateur

variables
tests
Boucles
fonctions

Outils

Binaires exécutables

Appels système

Système de fichiers

Système d'exploitation UNIX

Plan du cours 1

- Forme générale des commandes
- Exécution des commandes
- Redirections élémentaires
- Tubes de communication
- Codes de retour
- Groupement de commandes
- Caractères génériques
- Quelques commandes de base

Forme générale des commandes

- Forme générale d'une commande :

  [chemin**/**] nom_cmd [option ...] [argument ...]

```
$ ls –l /home/debbabi
drwx------+  5  debbabi  staff   170 Jan 28 07:22  Desktop
drwx------+ 34 debbabi  staff  1156 Feb  3 06:47 Documents
```

- Une *commande interne* est une commande dont le code est implanté au sein de l'interpréteur de commande. Exemple : **cd** , **echo** , **pwd** …
- Une *commande externe* est une commande dont le code se trouve dans un fichier ordinaire. Exemple **ls**, **mkdir**, **vi**, **sleep** … et les *fichiers shell*.

```
$ type –a echo
echo is a shell builtin
echo is /bin/echo
```

- Remarque:
  Bash distingue les caractères majuscules des caractères minuscules.

Exécution des commandes

- Exécution séquentielle (en premier-plan)

```
$ sleep 3 ; date
Tue Feb  4 11:18:57 CET 2014
```

  - L'exécution de date débute après que le délai de 3 secondes se soit écoulé.
  - Pour terminer l'exécution d'une commande, on appuie simultanément sur les touches CTRL et C

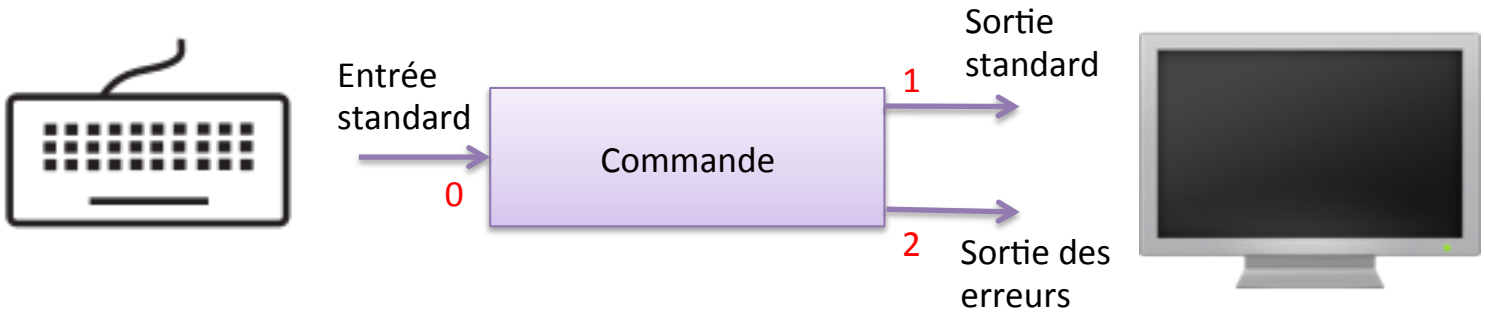
- Exécution en arrière-plan

```
$ sleep 3 &
[1] 4146
$ date
Tue Feb  4 11:18:57 CET 2014
```

  - La commande « sleep 3 » est lancée en arrière-plan.
  - Le système a affecté le numéro d'identification 4146 au processus, tandis que **bash** a affecté un numéro de travail égale à 1.

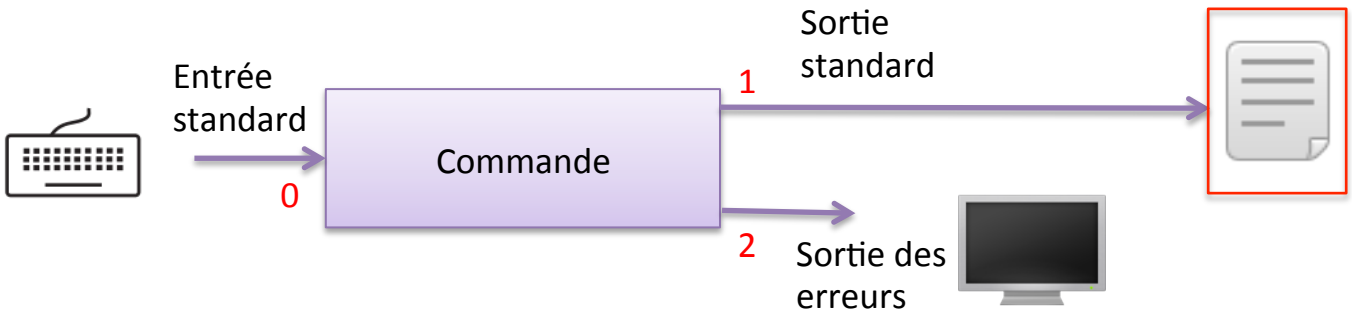
Redirections élémentaires

- Toute commande est exécutée par un processus Unix



Redirections élémentaires

- Redirection de la sortie standard



> *Fichier*

1> *fichier*

écrase l'ancien contenu du fichier

>> *fichier*

1>> *fichier*

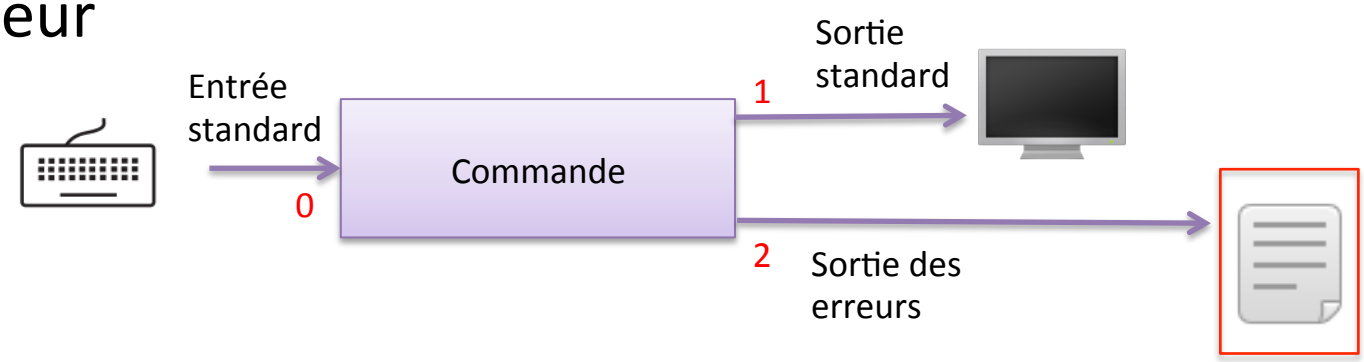
ajouter à la fin du fichier le contenu de la sortie standard

```
$ pwd
/home/debbabi
$ pwd > fichier
$
$ cat fichier
/home/sanchis $
```

=> aucun résultat affiché à l'écran !
=> le résultat a été enregistré dans le fichier *fichier*

Redirections élémentaires

- Redirection de la sortie standard pour les messages d'erreur



2> *fichier*

2>> *fichier*

On ne doit laisser aucun caractère **espace** entre le chiffre **2** et le symbole **>**.

2>&1

&>

Pour rediriger la sortie standard pour les messages d'erreur vers la sortie standard


```
$pwd
/home/debbabi
$ ls vi
ls: vi: Aucun fichier ou
répertoire de ce type
$ ls vi 2> /dev/null
$
```

=> l'éditeur de texte **vi** se trouve dans le répertoire **/usr/bin**

=> Le fichier spécial **/dev/null** est appelé « poubelle » ou « puits » car toute sortie qui y est redirigée, est perdue.

Redirections élémentaires

- Redirection de l'entrée standard

< *Fichier*

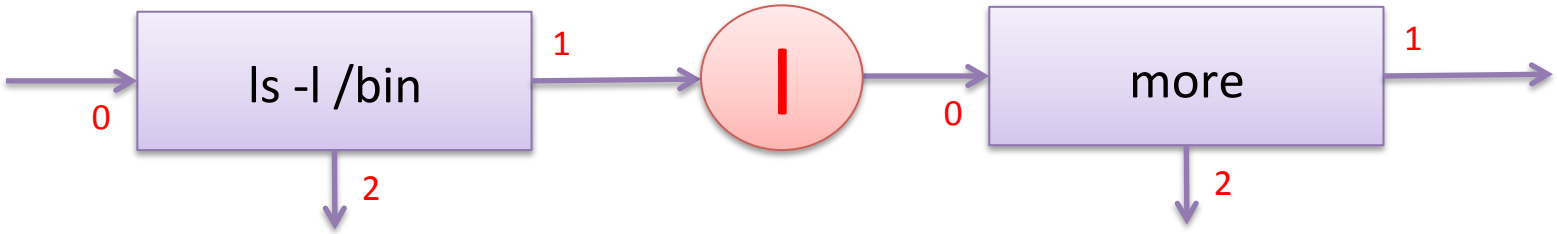
```
$ ls -l > fichier
$ wc -l  < fichier
     19
$
```

- Redirections séparées des entrées / sorties standard :

```
$ wc -l <fic_noms >fic_nblignes 2>err
```

Les tubes

- enchaîner l'exécution de commandes successives en connectant la sortie standard d'une commande à l'entrée standard de la commande suivante




```
$ ls -l | more
```

- Équivalent à

```
$ ls -l /bin >temporaire ; more < temporaire ; rm temporaire
```

Code de retour

- Principe : Un *code de retour* (0..255) est fourni par le shell après exécution d'une commande.
- **0** signifie que la commande s'est exécutée correctement

- **?** (*à ne pas confondre avec le caractère générique ?)* contient le code de retour de la dernière commande exécutée de manière séquentielle

- Exemple
  - ls -l vi ; echo $?
  - La commande unix grep positionne un code de retour
    - égal à 0 pour indiquer qu'une ou plusieurs lignes ont été trouvées
    - égal à 1 pour indiquer qu'aucune ligne n'a été trouvée
    - égal à 2 pour indiquer la présence d'une erreur de syntaxe ou qu'un fichier mentionné en argument est inaccessible.
- Remarque :
  - Le code de retour d'un programme shell est le code de retour de la dernière commande qu'il a exécutée.
  - exit n : provoque l'arrêt du programme shell avec un code de retour égal à *n*.

Groupement de commandes

- Principe : c'est l'exécution d'un ensemble de commandes en une fois sans pour cela affecter un nom à ce groupement.

- Exemples
  - { cd /bin ; pwd ;}
  - (cd /bin ; pwd)
  - { cd /bin ; pwd ;} &
  - (pwd ; date ; echo FIN ;) > fin
  - { read ligne1 ; read ligne2 ; } < fich
  - { read lig1 ; read -p "Entrez votre ligne : " lig2 </dev/tty read lig3 ; } < fich

Caractères génériques

- Utilisés pour spécifier un modèle de noms d'entrées
  - Ce modèle est ensuite interprété par le shell pour créer une liste triée de noms d'entrées
  - À noter : Par défaut, le shell traite uniquement les entrées non cachées du répertoire courant

- Un caractère générique désigne un ou plusieurs caractères
  - ? désigne un et un seul caractère
  - * désigne n'importe quelle suite de caractères, même la chaîne vide
  - [] désignent un seul caractère parmi un groupe de caractères

Caractères génériques

```
$ ls
1       En      a       an      emirat  minuit  zoulou
Arbre   _a      ami     e       etat    zaza
```

```
$ echo a*
$ echo *a*
$ echo Z*t
$ echo ?
$ echo ?n
$ echo ?mi*
$ echo [ame]
$ echo ?[ame]*
```

```
$ echo [a-z]*
$ echo [A-Z]*
$ echo [[:lower:]]
$ echo [[:upper:]]*
$ echo [[:upper:]a]*
$ echo ?[!n]*
$ echo [![:lower:]]
$ echo [![:upper:]]*[!at]
```

[:lower:] (minuscules)

[:upper:] (majuscules)

[:digit:] (chiffres, 0 à 9)

[:alnum:] (caractères alphanumériques)

Quelques commandes de base

- La commande d'aide « man »
  - Une commande UNIX permettant d'accéder aux pages de manuel installées sur le système
  - Utilisation : $ man [section] commande
    - Les sections :
      - 1) Programmes exécutables ou commandes du shell
      - 2) Appels système
      - 3) Appels de bibliothèque
      - 4) Fichiers spéciaux
      - 5) Formats des fichiers et conventions
      - 6) Jeux
      - 7) Divers …

$ man 1 echo

$ man 3 echo

Quelques commandes de base

- Manipulation des dossiers et fichiers

Commande	Role
pwd	Chemin du répertoire courant
cd	Changer le dossier courant
ls	Contenu d'un répertoire (list)
touch	Création d'un fichier
mkdir	Création d'un répertoire
rmdir	Suppression d'un répertoire vide
rm	Suppression d'un fichier
mv	Déplace ou renomme un fichier
cp	Copie physique d'un fichier
ln	Lien physique

Commande	Role
whereis	Recherche un fichier
find	Recherche de fichiers selon des critères
cat	Affiche le contenu d'un fichier
more	Affiche sur plusieurs pages
tar	Archive d'un fichier

Note : Voir la page de documentation de chaque commande pour savoir les options possibles.

Quelques commandes de base

Droits d'accès

Commande	Role
chmod	Change les droits d'accès
umask	Modifie les droits d'accès par défaut
chown	Change le propriétaire
chgrp	Change le groupe

Exécution d'un processus

Commande	Role
ps	Liste des processus (process)
kill	Envoie un signal au processus
which	Affiche le chemin d'une commande

Filtres

Commande	Role
sort	Trie les lignes d'un fichier
grep	Recherche ligne dans un texte
wc	Compte le nombre de lignes, de mots ou de caractères
head	Affiche n 1ère lignes d'un fichier
tail	Affiche de la ligne *n* à la fin/ début
cut	Extraction de colonnes
tr	Remplace *txt1* par *txt2*