

Introduction au système d'exploitation

1^{re} année

M. Bastregghi (mba) M. Paquot (apa)

Haute École de Bruxelles-Brabant — École Supérieure d'Informatique

Année académique 2019 / 2020

Séance 6

Shell et Scripts

- Shell
- Scripts et automatisations

bash - GNU Born Again SHell

Le shell est un interpréteur de langage de commandes

- 1 Affiche le PROMPT
- 2 Lit les commandes depuis clavier
- 3 Exécute les commandes

Le shell de linux s'appelle **bash**

analyse d'une commande

ls -ail dev1 int1 :

- ▶ ls - nom de la commande
- ▶ ail, dev1, int1 - paramètres de la commande

Commande Externe

Commande Interne

?

commande interne - externe

- ▶ externe - exécutée dans un processus séparé
- ▶ interne - réalisée par le code du bash

Une commande externe est documentée par sa page de manuel :

man ls

Une commande interne est documentée par :

man bash

ls ou /bin/ls ?

Où se trouve l'exécutable lié à
ma commande ?

commande externe

- ▶ **which ls**
- ▶ **PATH** : chemins où le shell cherche les commandes

Et pour une commande interne : **which cd** ?

entrée - sortie

Une commande peut

- ▶ Lire le clavier (entrée standard)
- ▶ Afficher à l'écran (sortie standard)

Les commandes qui font les deux sont appelées **filtres**

que fait

```
ls -l /dev|cut -c1-10|sort|uniq ?
```

commande filtre et pipe

- ▶ Filtre le flot de caractères provenant de l'entrée
- ▶ Le pipe est représenté par une barre verticale
- ▶ Le pipe connecte la sortie de la commande de gauche avec l'entrée de la commande à droite

Un filtre peut se trouver à droite d'un pipe ou entre deux pipes

ls | grep java

```
>ls *
```

```
>echo $PATH
```

expansion préalable de * et \$

Le shell réalise certaines **expansions** de ligne de commande

- ▶ * - noms des fichiers et répertoires du dossier courant
- ▶ \$NOM_VAR - contenu de la variable d'environnement NOM_VAR
- ▶ \$(comande) - remplacé par ce qu'affiche commande

Une expansion remplace en partie la commande

```
if mkdir DIR ; then cd DIR ; fi
```

statut de fin d'une commande

Une commande a un statut de fin qui peut être testé

```
if mkdir DIR ; then cd DIR ; fi
```

ou encore :

```
javac maclasse.java && java maclasse
```

Si la compilation est sans erreurs on exécute

```
java maclasse
```

script = fichier de commandes

scripts et automatisation

La première ligne d'un script bash est

!/bin/bash

- ▶ Le shebang **# !** est suivi du chemin de l'interpréteur
- ▶ Il indique au noyau qu'il s'agit d'un fichier de commandes

scripts et automatisation

Un script doit pouvoir être exécuté

```
>chmod a+x monscript
```

```
>./monscript
```

exemple 1

```
#!/bin/bash
echo
echo -n "quelle classe compiler ? "
read SOURCE
javac $SOURCE.java && java $SOURCE
echo
echo -n enter pour quitter
read
```

exemple 2

```
#!/bin/bash
# ceci est un commentaire
# $(whoami) est remplacé par l'affichage produit par la commande whoami
echo hello $(whoami)
echo -n "comment allez vous [OK NOK] ? "
read etat
if [ $etat == "OK" ]
then
echo ':)'
else
echo ':('
fi
```

il y a plein d'autres possibilités à
découvrir ...
soyez curieux ;)

Crédits

Ce document a été produit avec les outils suivants

- ▶ Les distributions **Ubuntu** et/ou **debian** et/ou **Open Suse** du système d'exploitation **Linux**
- ▶ **LaTeX/Beamer** comme système d'édition
- ▶ **Git** et **GitHub** pour la gestion des versions et le suivi des corrections
- ▶ Les outils **make**, **rubber**, **pdflatex**...

