

TD1 - ISS

Aura Leirós García
S4 2023/2024

Exercice 1.

1.1

Un langage interprété utilise un logiciel appelé interpréteur, qui utilise un processus dit « pas à pas » en exécutant les lignes du code une par une, ce qui permet d'avoir une grande compatibilité avec différents systèmes d'exploitation, mais avec un rendement inférieur.

Dans un langage compilé, le programme s'exécute directement sur l'ordinateur après avoir été traduit en binaire par un programme appelé compilateur, ce qui réduit fortement sa compatibilité, mais améliore la performance.

1.2

Deux exemples de langages compilés sont, par exemple, C ou OCaml.

1.3

Des langages interprétés sont par exemple Python ou Java.

1.4

Un langage de script est un langage de programmation, normalement interprété, qui permet de manipuler les fonctionnalités d'un système informatique.

Exercice 2.

2.1

Le terminal est simplement un programme d'entrée/sortie basée sur du texte qui permet à un utilisateur d'envoyer des commandes à un shell et de recevoir sa réponse.

2.2

C'est parce qu'avant la popularisation des ordinateurs actuels, le terminal était une machine physique.

2.3

Termius, Terminator, GNOME Terminal, ...

2.4

Le rôle d'un shell Unix est d'interpréter les commandes et de renvoyer leur sortie (output), mais aussi de gérer les processus, etc.

2.5

Il y a deux grandes familles de shells : les Bourne shells (comme sh, bash, ...) et les C-shells, dérivés de la syntaxe de C (csh, tcsh, ...).

Exercice 3.

3.1

1. Parsing: Bash parses the input into tokens, usually delimiting by whitespace (spaces, tabs, newlines).
2. Alias expansion: If the first word of a command is an alias, bash replaces it with the alias definition.
3. Brace expansion.
4. Variables expansion: Bash replaces the variables with their values.
5. Executing commands that are enclosed with `$()` and replacing that with the output of the command.
6. Bash expands the wildcard characters `*`, `?` and `[]` to match filenames
7. Quote removal: Bash removes the quotes `"`, `'` used to prevent the interpretation of special characters in the previous steps.
8. Redirection: Bash processes redirection and pipes `>`, `<`, `>>` and `|`
9. Execution of the command.

3.2

Yes it does it from scratch.

Exercice 4.

4.1

1.

```
Hello World
```

2.

```
Hello World
Hello World
```

3.

```
Hello world echo Hello world
```

4.

```
Hello world
-bash: Hello command not found
```

5.

```
Hello World
Hello World
```

6.

```
Hello World
Hello World
```

4.2

Yes, we can, the advantages are the simplicity, the portability and the efficiency.

4.3

Only the semicolon `;` and the newline.

Exercise 5.

5.1

In the sake of avoiding to print all the outputs, the goal of this exercise is explaining how the braces expansion works.

Before starting, for the expansion to work the elements inside the braces `{a,b}` must be separated only by a comma (not spaces).

If we want to use it combined with a prefix, the prefix and the braces must be connected by a character like in `echo fichier_{a,b}`

If we want to use nested expansions the inner braces must be immediately connected with an element like `{a, b, c{1,2}}` if not all the expression (outer and inner) will fail.

5.2

1.

```
$ echo libreoffice-{en-US,fr}
```

2.

```
$ echo python{-{scipy,matplotlib}}
```

3.

```
$ echo vim-{runtime,spell-{fr,en}}
```

5.3

1.

```
$ rm -f {list,tab}{c,h,o}
```

5.4

1.

```
$ echo fichier_{1..9}
```

2.

```
$ echo fichier_{0..8..2}
```

3.

Celui-ci avec un space !

```
$ echo {1..3} {6..12..2}
```

Exercice 6.

6.1

```
IDD td1 - 2019  
partiel 18 et examen:
```

6.2

String, l'unique type de variable en Shell est string.

6.3

Au `#define` qui est la substitution de texte.

6.4

```
$ annee=2010  
$ nom="Aloe Blacc"  
$ echo $nom $annee : I need a $
```

6.5

```
$ echo La variable '$myvar' contient la chaine $myvar qui a ${#myvar} caracteres
```

Exercice 7

7.1

`cat [fichier]` donne le contenu de chaque fichier, donc le contenu de `cv.txt` et un erreur pour la deuxième execution, parce que le `ma lettre.txt` comporte un espace, et donc, essaie de lire le fichier `ma`.

7.2

Juste renommer le fichier "ma lettre.txt" pour ne pas avoir des spaces.

7.3

Le `do` est hérite de sh et est necessaire pour que bash sait où se termine la liste de conditions.

7.4

Oui, mais on doit mettre un `;` à chaque commande.

```
for x in zero un ; do echo $x; echo "hehe"; done
```

Exercice 8.

8.1

1. All the PDF files.
2. All the elements inside the folder `?` if exists, error if not.
3. If `td1`, `td2` and / or `td3` exists then outputs the elements inside these folders.
4. If `td1` and / or `td3` exists, then outputs the elements present in the folder(s).
5. Matches `td1`, `td2` and / or `td3`, the `[charlist]` only matches a single-digit.
6. Matches all the elements that start with a lowercase letter and followed by any sequence of characters.

8.2

Parce que faire ça outputs toutes les subdirectoires et fichiers.

8.3

Le découpage en mots qui effectue bash. Car la variable ne se expande pas.

8.4

La substitution post-découpage.

8.5

Oui.