

Introduction aux systèmes d'exploitation (IS1)

TP 7 – Commandes conditionnelles et tests

Dans le TP précédent, nous vous indiquions que le shell était un véritable langage de programmation. Nous allons découvrir aujourd'hui une nouvelle fonctionnalité : les commandes conditionnelles. Voici leur syntaxe :

```
if cmd ; then cmd1 ; cmd2 ; ... ; else cmd3 ; cmd4 ; ... ; fi
```

Ce qui correspond au comportement suivant :

- exécution de la commande *cmd*
- exécution des commandes *cmd1 ; cmd2 ; ... ;* si la valeur de retour de *cmd* est 0
- exécution des commandes *cmd3 ; cmd4 ; ... ;* dans les autres cas

Il existe trois commandes particulières très utiles pour effectuer des tests :

true : cette commande renvoie toujours la valeur de retour 0.

false : cette commande renvoie toujours la valeur de retour 1.

test : cette commande permet d'effectuer de nombreux tests sur des fichiers, des chaînes de caractères, des entiers, etc... Ces tests sont accessibles via différentes options spéciales, que l'on appelle *primitives*. Ces *primitives* sont détaillées, *comme toujours*, dans la page de manue. Elles ont pour particularité de prendre des arguments, comme vous pouvez le voir dans cet exemple :

```
test -e fic -a \(! -x fic \)
```

Cette commande renvoie la valeur de retour 0 lorsque le fichier *fic* existe (primitive *-e*) mais (primitive *-a* pour la conjonction) n'est pas exécutable (primitive *-x* pour exécutable, primitive *!* pour la négation, et les parenthèses pour délimiter une sous-expression). On peut également écrire, de façon abrégée :

```
[ -e fic -a \(! -x fic \) ]
```

Exercice 1 – A la recherche du fichier `.bashrc`

1. Ecrivez une commande qui affiche le contenu de votre fichier `.bashrc` s'il existe, et la phrase « Vous n'avez pas de fichier `.bashrc`. », sinon.
2. Modifiez cette commande pour que dans le cas où le fichier `.bashrc` n'existe pas, elle le crée en y incluant vos `alias` (voir TP1) et la valeur de votre variable `PATH` (voir TP6).
3. Modifiez cette commande pour vérifier, non plus seulement que `.bashrc` existe, mais qu'en plus, c'est bien un fichier régulier.
4. Modifiez cette commande pour vérifier, non plus seulement que `.bashrc` existe et est un fichier régulier, mais qu'en plus, vous en êtes bien le propriétaire.

Exercice 2 – Tests de comparaison

1. Refaites l'exercice 5 du TP 4, mais cette fois-ci avec une structure `if then else`. L'objectif de cet exercice était d'obtenir une ligne de commande qui compare deux fichiers et affiche un message indiquant le résultat de la comparaison (identiques ou différents).
2. En utilisant une primitive de test, modifiez cette commande pour qu'elle affiche un message d'erreur lorsqu'un des deux fichiers n'existe pas.
3. Redirigez le résultat de cette commande vers un fichier tout en ayant, au préalable, vérifié que vous possédez bien les droits pour le faire.

Exercice 3 – Question de date

1. Écrivez une commande qui salue l'utilisateur en fonction de l'heure qu'il est : « Bonjour » avant 18 heures et « Bonsoir » après 18 heures et avant minuit.
2. Ajoutez cette salutation personnalisée à votre fichier de configuration `.bashrc`. Testez le résultat en ouvrant de nouveaux terminaux.
3. Modifiez cette salutation pour qu'elle indique si le jour courant est un jour de week-end ou non.

Exercice 4 – Arithmétique

Téléchargez, en utilisant la commande `wget`, l'archive suivante :

```
http://www.denis.cousineau.eu/Partage/IS108/tp7_arch.tar.gz
```

Puis extrayez-en son contenu. Les fichiers `qxy` contiennent des *questions* arithmétiques simples, les fichiers `rxxy` contiennent les *réponses* associées.

1. En utilisant la commande `read`, écrivez, sur une seule ligne, une suite de commandes qui demande à l'utilisateur numéro `xy`, compris entre 00 et 99, et qui affiche alors :

```
3859 + 2773 = 6632
```

si `qxy` contient `3859 + 2773` et `rxxy` contient `6632`.

2. Ecrivez une ligne de commande qui repère la question, parmi les cent, dont la réponse est incorrecte.

Exercice 5 – Problème

Ecrivez une commande qui affiche « Ça bosse pas depuis bien longtemps... ou alors vous connaissez votre arborescence sur le bout des doigts ! », si vous avez tapé moins de dix fois la commande `ls` dans le terminal courant, et affiche « Beau score ! la commande `ls` a été exécutée *n* fois », dans le cas contraire (où *n* est bien le nombre de commandes `ls` exécutées depuis le terminal courant).