Université Cadi Ayyad
Faculté des Sciences et Techniques
Guéliz-Marrakech
Département d'Informatique

FILIÈRE IRISI

TP6 : Programmation Shell (suite)

Module : Linux/Unix

Partie 1 : Rappel

Loops

Basic for loop

```
for i in /etc/rc.*; do
  echo $i
done
```

C-like for loop

```
for ((i = 0 ; i < 100 ; i++)); do
  echo $i
done
```

Ranges

```
for i in {1..5}; do
    echo "Welcome $i"
done
```

With step size

```
for i in {5..50..5}; do
    echo "Welcome $i"
done
```

Reading lines

```
cat file.txt | while read line; do
  echo $line
done
```

Forever

```
while true; do
  ...
done
```

Conditionals

Conditions

Note that `[[` is actually a command/program that returns either `0` (true) or `1` (false). Any program that obeys the same logic (like all base utils, such as `grep(1)` or `ping(1)`) can be used as condition, see examples.

`[[ -z STRING ]]`	Empty string
`[[ -n STRING ]]`	Not empty string
`[[ STRING == STRING ]]`	Equal
`[[ STRING != STRING ]]`	Not Equal
`[[ NUM -eq NUM ]]`	Equal
`[[ NUM -ne NUM ]]`	Not equal
`[[ NUM -lt NUM ]]`	Less than
`[[ NUM -le NUM ]]`	Less than or equal
`[[ NUM -gt NUM ]]`	Greater than
`[[ NUM -ge NUM ]]`	Greater than or equal
`[[ STRING =~ STRING ]]`	Regexp
`(( NUM < NUM ))`	Numeric conditions

More conditions

`[[ -o noclobber ]]`	If OPTIONNAME is enabled
`[[ ! EXPR ]]`	Not
`[[ X && Y ]]`	And
`[[ X || Y ]]`	Or

File conditions

`[[ -e FILE ]]`	Exists
`[[ -r FILE ]]`	Readable
`[[ -h FILE ]]`	Symlink
`[[ -d FILE ]]`	Directory
`[[ -w FILE ]]`	Writable
`[[ -s FILE ]]`	Size is > 0 bytes
`[[ -f FILE ]]`	File
`[[ -x FILE ]]`	Executable
`[[ FILE1 -nt FILE2 ]]`	1 is more recent than 2
`[[ FILE1 -ot FILE2 ]]`	2 is more recent than 1
`[[ FILE1 -ef FILE2 ]]`	Same files

Example

```
# String
if [[ -z "$string" ]]; then
  echo "String is empty"
elif [[ -n "$string" ]]; then
  echo "String is not empty"
else
  echo "This never happens"
fi
```

```
# Combinations
if [[ X && Y ]]; then
  ...
fi
```

```
# Equal
if [[ "$A" == "$B" ]]
```

```
# Regex
if [[ "A" =~ . ]]
```

```
if (( $a < $b )); then
   echo "$a is smaller than $b"
fi
```

```
if [[ -e "file.txt" ]]; then
  echo "file exists"
fi
```

Partie 2 : Scripts shell

Tâche 1 :

Écrivez un script bash qui enregistre dans un fichier les lignes saisies au clavier, et qui affiche le nombre de lignes qui ont été enregistrées.

Tâche 2 :

Créez la commande **copier.** La commande reçoit en argument deux noms de fichiers, la source et la destination. Le script se termine et affiche un message d'erreur si l'une des conditions suivantes est réalisée :

- Le nombre d'arguments est incorrect.
- Le fichier source n'existe pas ou il n'est pas copiable (pas d'accès en lecture).
- Le fichier source n'est pas un fichier ordinaire.
- Le fichier destination existe.
- Le répertoire de destination, que l'on peut connaître avec la commande **dirname,** n'est pas accessible en écriture.
- La copie a échoué.

Tâche 3 :

Écrivez un script bash permettant à partir du chemin d'un répertoire passé en argument de renommer tous les noms des fichiers de ce répertoire en minuscules.