

Cours : les structures conditionnelles

Les structures conditionnelles ressemblent en bash à toutes les autres syntaxes conditionnelles :

<mot-clef> <condition> <bloc conditionnel>

Structure conditionnelle simple

La fin de ligne étant un marqueur de fin d'instruction pour bash le début de bloc est marqué par le mot-clef "then".

```
if [ condition ]
then
  commande1
  commande2
  ...
fi
```

Le marquage de fin e bloc reste cependant nécessaire. Le shell choisit un mot clef inverse du mot-clef de début de structure : "fi" dans ce cas.

Alternative conditionnelle

La deuxième forme est l'alternative comme dans les langages de programmation classiques.

```
if [ condition ]
then
  commande1
  commande2
  ...
else
  commandej
  commandej+1
  ...
fi
```

Choix multiples

Par extension, le choix entre de multiples possibilités peut être réalisé par une nouvelle extension de la syntaxe :

```
if [ condition ]
then
  commande1
  commande2
  ...
elif [ condition2 ]
then
  commandej
  commandej+1
  ...
else
  commandek
  commandek+1
  ...
```

```
fi
```

On note donc une très grande similitude de construction avec le C ou d'autres langages procéduraux.

On note également une susceptibilité de l'analyseur syntaxique qui demande un respect assez pointu de l'écriture et de l'agencement syntaxique.

Le branchement

Le branchement permet de brancher à un certain nombre de cas suivant la valeur d'une variable :

```
case ${var##*.} in # extrait l'extension
gz)
    echo archive gz
    gzunpack ${SROOT}/${$x}
    ;;
bz)
    echo archive bz
    bzunpack ${SROOT}/${$x}
    ;;
zip)
    echo archive zip
    pkunzip ${SROOT}/${$x}
    ;;
*)
    echo archive non reconnue
    exit
    ;;
esac
```