

---

**TD 9 : Scripts UNIX (corrigé)**  
semaine du 31 mai 1999

---

**Exercice 1 : Mon premier script**

```
#!/bin/csh
# Mon premier script
clear
while (1)
    echo "Menu"
    echo "    Affichage repertoire courant          1"
    echo "    Liste des fichiers du repertoire      2"
    echo "    Informations sur un fichier                3"
    echo "    Changement de repertoire                  4"
    echo "    n premieres lignes d'un fichier          5"
    echo "    Sortie                                     0"
    echo -n "Choix: "
    set choix=$<
    switch ($choix)
    case 0:
        exit(0)
    case 1:
        pwd ; breaksw
    case 2:
        ls ; breaksw
    case 3:
        echo -n "Nom du fichier: " ; set file=$<
        ls -l $file ;breaksw
    case 4:
        echo -n "Nouveau repertoire: " ; set rep= $<
        cd $rep ; breaksw
    case 5:
        echo -n "Nom du fichier: " ; set file=$<
        echo -n "Nb de lignes a afficher: " ; set n=$<
        head -$n $file ; breaksw
    default:
        echo "Choix non propose"
    endsw
end
```

**Exercice 2 : Script de concatenation + tri de 2 fichiers**

a. Ecrire un script qui concatène puis trie deux fichiers `file1` et `file2` dans un nouveau fichier `file3` et qui affiche le nombre total de lignes. Les noms des trois fichiers doivent être passés en paramètre.

```
#!/bin/csh
# Script 2a.
# les arguments sont récupérés dans $1, $2 et $3

cat $1 $2 | sort > $3

# nblig reçoit le résultat de l'interprétation de la
# commande cat $3|wc -l (^ cause du `, back-quote)

set nblig=`cat $3|wc -l` # comme nblig est une variable numérique, on peut remplacer set par @

echo "Le fichier $3 contient $nblig lignes"
```

b. Modifier le script précédent pour demander à l'utilisateur de saisir au clavier le (ou les) nom(s) de fichiers qu'il aurait oublié d'indiquer en lançant le script

```
#!/bin/csh
# Script 2b.

# les arguments récupérés dans $1, $2 et $3
# servent d'initialisation aux variables f1, f2 et f3.

set f1=$1; set f2=$2 ; set f3=$3

# Attention : ne pas oublier les espaces ^ gauche et ^ droite du ==
while ($f1 == "")
  echo -n "fichier1:" ; set f1=$<
end

while ($f2 == "")
  echo -n "fichier2:" ; set f2=$<
end

while ($f3 == "")
  echo -n "fichier3:" ; set f3=$<
end

cat $f1 $f2 | sort > $f3
set nblig=`cat $f3 |wc -l`
echo "Le fichier $f3 contient $nblig lignes"
```

### Exercice 3 : Test sur les fichiers ou les répertoires

a. Ecrire un script qui vérifie l'existence dans le répertoire courant du fichier dont le nom est passé en paramètre.

```
#!/bin/csh
# script 3a.

# Pour utiliser le test d'existence (-e $file), il faut que $file != ""

if ($1 == "") then
  echo "aucun nom de fichier en parametre"
else if !(-e $1) then
  echo "$1 n'existe pas"
endif
```

b. Ecrire un script qui vérifie que le nom passé en paramètre est un fichier ou un répertoire.

```
#!/bin/csh
# script 3b.

# Pour utiliser le test de fichier (-f $file) ou de repertoire (-d $file),
# il faut que $file != ""
# Ces 2 tests marchent aussi (et répondent faux) si $file n'existe pas ...

if ($1 == "") then
  echo "aucun nom de fichier en parametre"
else if (-d $1) then
  echo "$1 est un repertoire"
else if (-f $1) then
  echo "$1 est un fichier"
else
  echo "$1 n'est ni un fichier ni un paramètre"
```

```
endif
```

#### Exercice 4 : Vérifier les noms de fichiers et de répertoires dans l'ex. 1

(Commentaire : à mon avis, vérifier que le nom de fichier est non-vide a un sens, mais bloquer l'utilisateur jusqu'à ce qu'il ait donné un nom de fichier ou de répertoire existant est une maladresse, parce que ls -l ou cd traitent l'erreur, et que l'utilisateur a sans doute besoin de l'option 2 pour rectifier. J'ai mis \$< entre quotes '\$<' pour autoriser les noms de fichier avec des espaces ou des caractères jocker ? et \* — fl)

```
#!/bin/csh
# Mon premier script + verification des noms de fichiers et de repertoire
clear
while (1)
  echo "Menu"
  echo "      Affichage repertoire courant          1"
  echo "      Liste des fichiers du repertoire      2"
  echo "      Informations sur un fichier            3"
  echo "      Changement de repertoire              4"
  echo "      n premieres lignes d'un fichier       5"
  echo "      Sortie                                0"
  echo -n "Choix: "
  set choix=$<
  switch ($choix)
  case 0:
    exit(0)
  case 1:
    pwd ; breaksw
  case 2:
    ls ; breaksw
  case 3:
    #===== Info fichier =====
    @ OK=1
    # repeter tant que $file n'est pas un fichier
    while ($OK == 1)
      # saisie d'un nom de fichier non vide !
      while ($file == "")
        echo -n "Nom du fichier: " ; set file='$<'
      end
      if !(-f $file) then
        echo "$file n'est pas un fichier"
      else
        @ OK=0
      endif
    end
    ls -l $file ;breaksw
  case 4:
    #===== Chgt repertoire =====
    @ OK=1
    while ($OK == 1)
      while ($rep == "")
        echo -n "Nouveau repertoire: " ; set rep= '$<'
      end
      if !(-d $rep) then
        echo "$rep n'est pas un repertoire "
      else
        @ OK =0
      endif
    end
    cd $rep ; breaksw
  case 5:
    #===== n premieres lignes =====
    @ OK=1
    while ($OK == 1)
      while ($file == "")
        echo -n "Nom du fichier: " ; set file='$<'
      end
      if !(-f $file) then
        echo "$file n'est pas un fichier"
      else
```

```
                @ OK = 0
            endif
        end
        echo -n "Nb de lignes a afficher: " ; set n=$<
        head -$n $file ; breaksw
    default:
        echo "Choix non propose"
    endsw
end
```