

Partie shell : COURS-TD1

Partie 1 : Cours et exemples.

Présenter le chapitre 7 du poly : exercice 7 pas obligatoire, mais faire l'exo 9.

Parler aussi des redirections et du tube de commandes.

Pour la partie "quote-backquote" je propose de présenter la suite ci-dessous (on suppose le prompt égal à \$) et d'en tirer les conclusions :

```
$ a=46      écrire le = SANS ESPACE de part et d'autre
$ echo $a
    46
$ b=livre
$ echo $b
    livre
$ c='date'
$ echo $c
    date
$ d="le chien"
$ echo $d
    le chien
$ d='le chien'
$ echo $d
    le chien
$ echo "$a"
    46
$ echo '$a'
    $a
$ c='date'
$ echo $c
    Fri Feb 19 14:59:01 CET 2010
$ echo '$a'
    46: command not found
$ echo '$c'
    $c
$ c=le chien
    sh: chien: not found    (message d'erreur)
```

Moralité : on écrira correctement les quote et backquote et on évitera les traits droits...

et

- les guillemets " " sont perméables à l'interprétation,
- les quotes ' ' bloquent le mécanisme de substitution,
- les backquotes ` ` exécute ce qui se trouve entre-elles.

Présenter le chapitre 8 : instructions conditionnelles, itérations (for, while, shift), et l'arithmétique ($i = i + 1$ par exemple), avec les exemples du poly en guise d'exercices (mais à faire un peu soi-même pour montrer comment on écrit un script).

Partie 2 : Exercices.

1. Commande lecture.

Ecrire une commande lecture permettant de boucler sur une lecture au clavier jusqu'à obtention de la chaîne de caractères "oui" ou "non".

Solution possible :

```

#!/bin/sh
# lecture oui non : commande lecture

q="Repondez par oui ou non: "
echo $q
# dans echo -n $q l'option -n permet de ne pas passer à la ligne suivante

read reponse
echo $reponse

while [ "$reponse" != oui -a $reponse != "non" ]
# des espaces OBLIGATOIRES à cote de !=
do
    echo $q
    read reponse
done

```

Autre version possible: MAIS je n'arrive plus à le faire tourner, je pense que c'est à cause de \$<.

```

#!/bin/sh
# lecture oui non : commande lecture

q="Repondez par oui ou non: "
echo $q
# dans echo -n $q l'option -n permet de ne pas passer à la ligne suivante

while [ $< != "oui" -a $< != "non" ]
# des espaces OBLIGATOIRES à cote de !=
# $< recuperation sans traitement de l'entrée standard
do
    echo $q
    read reponse
done

# RAPPELS :
# comparaison de chaine de caracteres : == !=
# et ou not : -a -o !
# comparaison d'entiers: -gt -ge (>=) -lt (<) -le -eq (=) -ne (<>)

```

S'il reste du temps, traiter l'exercice suivant, sinon le proposer dans la séance de TP.

2. Commande enlever.

Ecrire une commande enlever nom liste_de_noms qui affiche sur la sortie standard la liste de noms privée de toutes les occurences de nom.

```
#!/bin/sh
# commande:  enlever nom liste_de_noms

if [ $# -le 1 ]
then
    echo "usage : $0 nom liste_de_noms"
    exit 1
fi
# $# est le nombre de vrais arguments (sans le nom de la cde)

nom=$1
shift

# $* désigne tous les arguments de la ligne de commande
for i in $*
do
    if [ $i != $nom ]
    then echo -n "$i "
    fi
done

# pour passer à la ligne avant de terminer
echo
exit 0
```