

Corrigé du TP 7 Système

C. Pain-Barre

INFO - IUT Aix-en-Provence

version du 11/12/2012

1 Substitution de commandes et d'expressions arithmétiques

Corrigé de l'exercice 1

Étude des substitutions de commandes et d'expressions arithmétiques

1. `$ cd ~/tp/tpunix`
2. `$ ls`
acrostiche.txt cigale.txt essai1.txt fruit.price
a_decouper.txt ...
3. `$ liste="$(ls)"`
4. `$ echo "$liste"`
acrostiche.txt
a_decouper.txt
amphigouri.txt
...
5. `$ echo $liste`
acrostiche.txt a_decouper.txt amphigouri.txt ...
6. `$ shopt -os xtrace`
7. `$ liste="$(ls)"`
++ ls --color=auto
+ liste='acrostiche.txt
a_decouper.txt
amphigouri.txt
...'
8. `$ shopt -ou xtrace`
+ shopt -ou xtrace
9. `$ xyz=100`
10. `$ echo "Résultat : $((xyz * 7))"`
Résultat : 700
11. `$ mkdir essai ; cd essai`
12. `$ touch f1 f2 'avec_espace'`
13. `$ ls`
avec espace f1 f2
14. `$ t1s1=$(ls)`
 - a. `$ echo ${#t1s1[@]}`
4
 - b. `$ echo ${t1s1[0]}`
avec

15. \$ **tls2=(*)**

a. \$ **echo \${#tls2[@]}**
3

b. \$ **echo "\${tls2[0]}"**
avec espace

Corrigé de l'exercice 2

Étude de la décomposition en mots

1. ... pas besoin de corrigé pour cette question...

2. ... pas besoin de corrigé pour cette question...

3. \$ **ckmots ***
3 argument(s) :
-->avec espace<--
-->f1<--
-->f2<--

⇒ le motif ***** est remplacé par une liste de mots qui ne sont pas sujets à la décomposition en mots

\$ **ckmots \$(ls)**
4 argument(s) :
-->avec<--
-->espace<--
-->f1<--
-->f2<--

⇒ la substitution de la commande **ls** est sujette au découpage en mots

\$ **ckmots "\$ (ls) "**
1 argument(s) :
-->avec espace
f1
f2<--

⇒ les guillemets empêchent le découpage en mots de la substitution de **ls**

4. \$ **OIFS="\$IFS"**
\$ **IFS=\$'\t\n'**

5. \$ **ckmots \$(ls)**
3 argument(s) :
-->avec espace<--
-->f1<--
-->f2<--

6. \$ **ckmots \$(ls)**
5 argument(s) :
-->av<--
-->c <--
-->spac<--
-->f1<--
-->f2<--

```

7. $ IFS="$OIFS"
   $ ckmots $PATH
   1 argument(s) :
     -->/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games<--

8. $ IFS=':'
   $ ckmots $PATH
   5 argument(s) :
     -->/usr/local/bin<--
     -->/usr/bin<--
     -->/bin<--
     -->/usr/local/games<--
     -->/usr/games<--

9. $ tabpath=($PATH)
   $ echo ${tabpath[1]}
   /usr/bin

10. $ IFS="$OIFS"
   $ ckmots $((11 * 12))
   1 argument(s) :
     -->132<--

11. $ IFS=3
   $ ckmots $((11 * 12))
   2 argument(s) :
     -->1<--
     -->2<--

12. $ IFS=$'\_\\t\\n'

13. $ cd .. ; rm -rf essai

```

Corrigé de l'exercice 3

Utilisation des substitutions de commandes

```

1. $ ls -l $(tty)
2. $ chmod go+w $(tty)
3. $ ls -l $(tty)
4. $ chmod go-w $(tty) ; ls -l $(tty)
5. $ ls -l $(type -P passwd)
6. $ luid=$(grep cpb /etc/passwd | cut -d: -f 3)
   $ echo $luid
   1002
7. $ kiwi=$(grep kiwi fruit.price)
   $ echo "${kiwi[0]} -- ${kiwi[1]}"
   kiwi -- 3.89
8. $ kiwiprix=$(grep kiwi fruit.price | sed -e 's/^.*_//')
   ou pourquoi pas :
   $ kiwiprix=$(echo $(grep kiwi fruit.price) | cut -d'_' -f2)
   puis :
   $ echo $kiwiprix
   3.89

```

9. `$ f=trucmuche`
10. `mkdir $(echo $f | sed -e 'y|trucmhe|TRUCMHE|')`
11. `cat $(find ~/tp -type f -name *.txt -print) > TOUT.TXT`

2 Valeur de retour, test de condition et suite de commandes

Corrigé de l'exercice 4

Utilisation des tests de conditions et de la variable \$?

1. `test -d /tmp` ou `[-d /tmp]` puis `echo $?`
2. `VAR=50`
3. `((VAR < 7))` puis `echo $?`
4. `echo $?` écrit 0 car il s'agit de la valeur de retour de la commande précédente (**echo**)
5. `test $VAR -lt 7` ou `[$VAR -lt 7]` puis `echo $?`
 ⇨ le test `[$VAR < 7]` est une comparaison de chaîne renvoyant 0 (vrai) !
6. `((30 < VAR && VAR < 100)) ; echo $?`
 ⇨ on écrit le tout sur une seule ligne en séparant les deux commandes par ;
7. `((VAR < 40 || VAR % 10 == 0)) ; echo $?`

Corrigé de l'exercice 5

Utilisation des opérateurs && et || dans les suites de commandes

1. `((VAR < 40 || VAR % 10 == 0)) && echo "c'est ma foi vrai"`
2. `[-f /tmp] || echo "ben il n'existe pas le fichier ordinaire /tmp ?"`
3. `((VAR % 10 == 0)) && [-x /tmp] && echo 'yes : tout est vrai !'`
4. `((VAR > 100)) || [-x /tmp] && echo "OK"`
 ⇨ ou encore : `[$VAR -gt 100 -o -x /tmp] && echo "condition vraie"`

3 Premiers scripts



Pour les corrigés des scripts, l'accent est mis sur la lisibilité plutôt que sur la performance...

Corrigé de l'exercice 6

Script affichant un message dépendant du nombre de ses arguments

```
#!/bin/bash

# Script : argtest.bash

if (($# == 1)); then
    echo "un seul argument"
elif (($# > 1)); then
    echo "plusieurs arguments"
else
    echo "aucun argument"
fi
```

Corrigé de l'exercice 7

Scripts comparant ses arguments

```
#!/bin/bash

# Script : in.bash

function isnum {
    echo "$1" | grep -q "^[+-]\?[0-9]\+$"
}

if (($# != 3)); then
    echo "${0##*/}: 3 arguments numériques requis" >&2
    exit 2
elif ! isnum "$1" || ! isnum "$2" || ! isnum "$3"; then
    echo "${0##*/}: 3 arguments numériques requis" >&2
    exit 2
fi

if (($1 <= $2 && $2 <= $3)); then
    exit 0
else
    exit 1
fi
```

4 Personnalisation de l'environnement bash

Corrigé de l'exercice 8

Copie des fichiers de personnalisation.

```
1. $ ls -la ~
total 36
drwx---r-x 6 toto vboxusers 4096 28 nov. 17:09 .
drwxr-xr-x 5 root root      4096  2 déc. 02:40 ..
lrwxrwxrwx 1 toto vboxusers  22 28 nov. 15:21 .bash_aliases -> net-home/.bash_aliases
lrwxrwxrwx 1 toto vboxusers  24 28 nov. 15:21 .bash_functions -> net-home/.bash_functions
-rw----- 1 toto vboxusers 1065  3 déc. 17:22 .bash_history
lrwxrwxrwx 1 toto vboxusers  21 28 nov. 15:21 .bash_logout -> net-home/.bash_logout
lrwxrwxrwx 1 toto vboxusers  22 28 nov. 15:21 .bash_profile -> net-home/.bash_profile
```

```
lrwxrwxrwx 1 toto vboxusers 16 28 nov. 15:21 .bashrc -> net-home/.bashrc
drwx---r-x 4 toto vboxusers 0 3 déc. 16:00 net-home
...
```

2. \$ ls -laL ~

```
ls: impossible d'accéder à /home/toto/.bash_aliases: Aucun fichier ou dossier de ce type
ls: impossible d'accéder à /home/toto/.bash_functions: Aucun fichier ou dossier de ce type
ls: impossible d'accéder à /home/toto/.bash_logout: Aucun fichier ou dossier de ce type
ls: impossible d'accéder à /home/toto/.bash_profile: Aucun fichier ou dossier de ce type
ls: impossible d'accéder à /home/toto/.bashrc: Aucun fichier ou dossier de ce type
total 48
drwx---r-x 6 toto vboxusers 4096 28 nov. 17:09 .
drwxr-xr-x 5 root root 4096 2 déc. 02:40 ..
l???????? ? ? ? ? ? .bash_aliases
l???????? ? ? ? ? ? .bash_functions
-rw----- 1 toto vboxusers 1065 3 déc. 17:22 .bash_history
l???????? ? ? ? ? ? .bash_logout
l???????? ? ? ? ? ? .bash_profile
l???????? ? ? ? ? ? .bashrc
drwx---r-x 4 toto vboxusers 0 3 déc. 16:00 net-home
...
```

3. \$ cp /commun/pain-barre/bashskel/.bash* ~/net-home

Corrigé de l'exercice 9

Étude des fichiers /etc/profile et ~/.bash_profile, puis personnalisation de ~/.bash_profile

1.

```
# /etc/profile: system-wide .profile file for the Bourne shell (sh(1))
# and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).
```

```
if [ "`id -u`" -eq 0 ]; then
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
else
PATH="/usr/local/bin:/usr/bin:/bin:/usr/games"
fi
```

```
if [ "$PS1" ]; then
if [ "$BASH" ]; then
PS1='\u@\h:\w\$ '
else
if [ "`id -u`" -eq 0 ]; then
PS1=#
else
PS1=$
fi
fi
fi
PATH=$PATH:/usr/local/eclipse-C++:/usr/local/eclipse-java
export PATH
```

```
umask 022
```

```
export TWO_TASK=orcl
export EDITOR=/usr/bin/gvim
```

```
PT5HOME=/usr/local/PacketTracer5
export PT5HOME
```

```
mesg n
```

2. `$ cat ~/.bash_profile`

⇒ le `~/` n'est pas nécessaire car on devrait se trouver dans le répertoire d'accueil

3. `$ mkdir ~/net-home/bin`

`$ cp /commun/pain-barre/unix/sorties ~/net-home/bin`

4. `$ sorties`

`-bash: sorties: command not found`

5. `$ echo $PATH`

`/usr/local/bin:/usr/bin:/bin:/usr/games (...)`

⇒ il y a probablement encore d'autres chemins à la suite

6.

7. `$ mv ~/net-home/bin/sorties ~/net-home`

8. `$ sorties`

`-bash: sorties: command not found`

9. ... *pas besoin de corrigé pour cette question...*

10. ... *pas besoin de corrigé pour cette question...*

11. ... *pas besoin de corrigé pour cette question...*

12. ... *pas besoin de corrigé pour cette question...*

13. ... *pas besoin de corrigé pour cette question...*

Corrigé de l'exercice 10

Étude du fichier `~/.bashrc` et création d'alias persistants dans `~/.bash_aliases`

1. `$ cat ~/.bashrc`

2. `$ vi ~/.bash_aliases`

(a) `alias clean='rm -fv *~ .*~ *.sw?'`

(b) `alias sl='ls'`

(c) `alias up='cd ..'`

(d) `alias ssha='ssh etxxxx@allegro'`

(e) Sortie du mode insertion avec Esc puis `:wq`

3. `$ alias`

`alias cp='cp -i'`

`alias ll='ls -l'`

`alias ls='ls --color=auto'`

`alias mv='mv -i'`

`alias rm='rm -i'`

`alias vi='vim'`

4. `$ source ~/.bash_aliases`

5. `$ alias`

`alias clean='rm -fv *~ .*~ *.sw?'`

`alias cp='cp -i'`

`alias ll='ls -l'`

`alias ls='ls --color=auto'`

```
alias mv='mv -i'
alias rm='rm -i'
alias sl='ls'
alias ssh='ssh etxxxx@allegro'
alias up='cd ..'
alias vi='vim'
```

6. ... pas besoin de corrigé pour cette question...

Corrigé de l'exercice 11

Modification de ~/.bashrc et création de fonctions persistantes dans ~/.bash_functions

1. \$ **vi ~/.bashrc**
2. **if [-f ~/.bash_functions]; then**
 . ~/.bash_functions
fi
3. Sortie du mode insertion avec Esc puis **:wq**
4. \$ **vi ~/.bash_functions**
 - (a) ... pas besoin de corrigé pour cette question...
 - (b) Sortie du mode insertion avec Esc puis **:wq**
5. \$ **source ~/.bash_functions**
6. \$ **cp /commun/pain-barre/unix/sorties.cxx ~/net-home**
 \$ **rm -f ~/net-home/sorties**
7. \$ **cd ~/net-home ; mk sorties**
8. \$ **sorties**
Ce texte a été écrit sur la sortie standard
Ce texte a été écrit sur la sortie d'erreur
Ce texte a été écrit sur la sortie standard
Ce texte a été écrit sur la sortie d'erreur

Corrigé de l'exercice 12 (en local sur le PC)

Ajout facultatif de divers alias et fonctions

1. **function xvi {**
 gnome-terminal --hide-menubar -t "\$1" -x vi "\$1" &
}

 ➡ selon l'installation, **vi** est le vrai (ancien) **vi**. Dans ce cas, il vaut mieux écrire **vim** à la place pour utiliser la version améliorée.
2. ... pas besoin de corrigé pour cette question...
3. **alias aliases='vi ~/.bash_aliases ; source ~/.bash_aliases'**
4. **alias functions='vi ~/.bash_functions ; source ~/.bash_functions'**