

Corrigé du TP 9 Système

C. Pain-Barre

INFO - IUT Aix-en-Provence

version du 16/12/2012

1 Environnement des processus

Corrigé de l'exercice 1

Étudier l'héritage de l'environnement d'un processus bash par ses fils.

1. Mise en place d'un environnement.

```
$ echo "$MAVARNOM"
normale
$ echo "$MAVARENV"
environnement
$ MAFONCNORM
message affiché par MAFONCNORM
$ MAFONCENV
message affiché par MAFONCENV
$ umask
0022
$ alias
alias ls='ls --color=auto'

↔ au moins cet alias devrait exister
```

```
$ ls -l /proc/$$/fd
total 0
lrwx----- 1 toto toto 64 16 déc. 15:06 0 -> /dev/pts/9
lrwx----- 1 toto toto 64 16 déc. 15:06 1 -> /dev/pts/9
lrwx----- 1 toto toto 64 16 déc. 15:06 2 -> /dev/pts/9
lrwx----- 1 toto toto 64 16 déc. 15:06 255 -> /dev/pts/9
```

2. Vérification de l'environnement hérité par un sous-shell.

```
$ ( echo "$MAVARNORM" ; echo "$MAVARENV"; MAFONCNORM ; MAFONCENV ;
    umask ; alias; ls -l /proc/$BASHPID/fd )
MAVARNORM
environnement
message affiché par MAFONCNORM
message affiché par MAFONCENV
0022
alias ls='ls --color=auto'
total 0
lrwx----- 1 toto toto 64 16 déc. 15:06 0 -> /dev/pts/9
lrwx----- 1 toto toto 64 16 déc. 15:06 1 -> /dev/pts/9
lrwx----- 1 toto toto 64 16 déc. 15:06 2 -> /dev/pts/9
lrwx----- 1 toto toto 64 16 déc. 15:06 255 -> /dev/pts/9
```

3. Indépendance de l'environnement du fils.

```
$ ( cd / ; MAVARENV="fils"; unalias ls )
$ pwd ; echo "$MAVARENV" ; alias ls
/home/toto/tp/tpunix
environnement
alias ls='ls --color=auto'
```

4. Vérification de l'environnement d'un fils exécutant une commande externe.

```
$ ./verifenv.bash
Verif de l'environnement de ce processus (PID 1241)
Valeur de 'MAVARNORM' : <inconnue>
Valeur de 'MAVARENV' : environnement
Appel de MAFONCENORM
./verif_env.bash: line 9: MAFONCENORM : commande introuvable
Appel de MAFONCENV
message affiché par MAFONCENV
Répertoire de travail : /home/toto/tp/tpunix
Masque (umask) : 0022
Liste des alias :
Fin des alias.
Liste des entrées-sorties :
total 0
lrwx----- 1 toto toto 64 16 déc. 16:13 0 -> /dev/pts/9
lrwx----- 1 toto toto 64 16 déc. 16:13 1 -> /dev/pts/9
lrwx----- 1 toto toto 64 16 déc. 16:13 2 -> /dev/pts/9
lr-x----- 1 toto toto 64 16 déc. 16:13 255 -> /home/toto/tp/tpunix/verif_env.bash
Fin des entrées-sorties
```

5. Variables d'environnement dans un programme (C/C++).

6. Création de variables d'environnement pour un processus.

```
$ echo "XXX=$XXX et YYY=$YYY"
XXX= et YYY=
```

7. \$ (export -n MAVARENV ; ls -l /proc/\$BASHPID/fd; ./verifenv.bash > ficout) < cigale.txt

```
total 0
lr-x----- 1 cyril users 64 16 déc. 18:21 0 -> /home/toto/tp/tpunix/cigale.txt
lrwx----- 1 cyril users 64 16 déc. 18:21 1 -> /dev/pts/9
lrwx----- 1 cyril users 64 16 déc. 18:21 2 -> /dev/pts/9
lrwx----- 1 cyril users 64 16 déc. 18:21 255 -> /dev/pts/9
./verifenv.bash: line 7: MAFONCENORM : commande introuvable
$ cat ficout
Verif de l'environnement de ce processus (PID 1666)
Valeur de 'MAVARNORM' : <inconnue>
Valeur de 'MAVARENV' : <inconnue>
Appel de MAFONCENORM
Appel de MAFONCENV
message affiché par MAFONCENV
Répertoire de travail : /home/toto/tp/tpunix
Masque (umask) : 0022
Liste des alias :
Fin des alias.
Liste des entrées-sorties :
total 0
lr-x----- 1 cyril users 64 16 déc. 18:21 0 -> /home/toto/tp/tpunix/cigale.txt
l-wx----- 1 cyril users 64 16 déc. 18:21 1 -> /home/toto/tp/tpunix/ficout
```

```
lrwx----- 1 cyril users 64 16 déc. 18:21 2 -> /dev/pts/9
lr-x----- 1 cyril users 64 16 déc. 18:21 255 -> /home/toto/tp/tpunix/verifenv.bash
Fin des entrées-sorties
```

Corrigé de l'exercice 2

Manipuler des variables d'environnement.

1. `$ echo $LANG`
fr_FR.utf8
2. `$ date`
dimanche 16 décembre 2012, 17:21:23 (UTC+0100)
3. `$ LANG=C`
`$ date`
Sun Dec 16 17:21:30 CET 2012
4. `$ rm -iv cigale.txt`
rm : supprimer fichier « cigale.txt » ? **n**
5. `$ LANG=C rm -iv cigale.txt`
rm: remove regular file `cigale.txt'? **o**


Corrigé de l'exercice 3 (Facultatif)

*Modification des couleurs d'affichage de ls par modification de la variable (d'environnement) **LS_COLORS***

... Il ne devrait pas y avoir besoin de corrigé pour cet exercice...

Corrigé de l'exercice 4

Script bouclant selon les saisies de l'utilisateur. Cet exercice sert aussi de prétexte pour illustrer un usage des variables d'environnement ainsi que la gestion de valeurs par défaut.

 Ce code a été volontairement condensé en exploitant les facilités de bash pour que celui-ci tienne sur une page. En espérant néanmoins qu'il reste suffisamment compréhensible.

```
#!/bin/bash

# Script : boucle.bash

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

function lire_nombre {
    local carlus;
    while true; do
        read -p "nombre $1 (entre $liminf et $limsup) ? " carlus
        if isnum "$carlus" && ((liminf <= carlus && carlus <= limsup)); then
            echo "$carlus"
            return
        fi
        echo "nombre incorrect" >&2
    done
}

liminf=0
limsup=99

if isnum "${LESLIMITESINF:-inconnue}"; then
    liminf="${LESLIMITESINF}"
fi
if isnum "${LESLIMITESSUP:-inconnue}"; then
    limsup="${LESLIMITESSUP}"
fi
if ((liminf > limsup)); then tmp=$liminf ; liminf=$limsup ; limsup=$tmp; fi

continuer='o'
while [ $continuer == 'o' -o $continuer == 'O' ]; do
    nb1=$(lire_nombre 1)
    nb2=$(lire_nombre 2)

    while true; do
        read -p "somme ? " somme
        isnum "$somme" && break
        echo "nombre incorrect" >&2
    done

    ((somme != nb1 + nb2)) && break

    read -p "Continuer (o/O/n/N) ? " continuer
    until [ "$continuer" == 'o' -o "$continuer" == 'O' \
        -o "$continuer" == 'n' -o "$continuer" == 'N' ]
    do
        echo "Saisie incorrecte" >&2
        read -p "Continuer (o/O/n/N) ? " continuer
    done
done
```