

Architecture des Ordinateurs & Systèmes d'Exploitation TP 8

Exercice 1 : Découverte du Shell

* Nous voulons visualiser les fichiers `.bash_profile` et `.bashrc`. Pour cela nous utilisons la commande `cat`.

```
cat .bash_profile
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs
PATH=$PATH:$HOME/bin:.$HOME/Office51/bin
BASH_ENV=$HOME/.bashrc
USERNAME=""
export USERNAME BASH_ENV PATH
```

Ce fichier s'exécute à chaque ouverture de fenêtre et exécute le fichier `.bashrc` donc le fichier `.bashrc` est exécuté tout le temps.

```
cat .bashrc
# .bashrc

# User specific aliases and functions

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi
PATH=$PATH:/applications/MATLAB5/bin
alias ll='ls -alh'
```

C'est ici que l'on rajoute les alias.

Ce fichier est exécuté à chaque fois que l'on se logue. Ce fichier est lu à chaque fois qu'on lance `bash` à condition que la variable `ENV=$HOME/.bashrc` soit bien définie dans le `.bash_profile`

* Nous voulons ensuite visualiser toute les variables d'environnement. Pour cela, on utilise la commande `set`. On obtient alors la liste de toute les variables d'environnement :

<code>BASH=/bin/bash</code>	Chemin d'accès au bash
<code>BASH_VERSINFO=([0]="2" [1]="04" [2]="11" [3]="1" [4]="release" [5]="i386-redhat-linux-gnu")</code>	Informations sur le bash utilisé
<code>BASH_VERSION='2.04.11(1)-release'</code>	Version du bash
<code>COLORS=/etc/DIR_COLORS</code>	Chemin du répertoire où sont répertoriés les couleurs
<code>COLORTERM=</code>	
<code>COLUMNS=142</code>	Largeur en caractère de l'écran ou de la fenêtre
<code>DIRSTACK=()</code>	Une variable-tableau représentant le contenu

Volatiana RALAMBONDRAINY – Samy FOUILLEUX – Maxime CHAMBREUIL
ASI 3 – Année 2001 / 2002

	courant de la pile
<i>DISPLAY=:0.0</i>	
<i>DOSEMUDIR=/home/etud/asi01/mchambre/mydos/dosemu</i>	Chemin du répertoire de dosemu, émulateur dos
<i>EUID=596</i>	Contient l'UID effectif de l'utilisateur, initialisé au démarrage du shell
<i>GROUPS=()</i>	Le groupe auquel on appartient
<i>GTK_RC_FILES=/etc/gtk/gtkrc:/home/etud/asi01/mchambre/.gtkrc</i>	
<i>HISTFILE=/home/etud/asi01/mchambre/.bash_history</i>	Le fichier d'historique
<i>HISTFILESIZE=500</i>	La taille du fichier d'historique
<i>HISTSIZE=500</i>	La taille de l'historique
<i>HOME=/home/etud/asi01/mchambre</i>	Le répertoire de l'utilisateur
<i>HOSTNAME=asi1248.insa-rouen.fr</i>	Le nom du serveur duquel on dépend
<i>HOSTTYPE=i386</i>	Le Type de la machine où l'on s'est connecté .
<i>IFS=' '</i>	Séparateur de champ interne
<i>KDEDIR=/usr</i>	Prompt principal ou prompt de shell primaire
<i>KDE_DISPLAY=:0.0</i>	
<i>KDE_INITIAL_DESKTOP=2</i>	
<i>KDE_MULTIHED=faalse</i>	
<i>LANG=fr_FR</i>	Langue utilisée, ici français
<i>LD_LIBRARY_PATH=/usr/lib</i>	
<i>LESSOPEN= /usr/bin/lesspipe.sh %s'</i>	Filtre
<i>LINES=48</i>	Longueur en caractère de l'écran ou de la fenêtre
<i>LOGNAME=mchambre</i>	Login de l'utilisateur
<i>LS_COLORS='no=00:fi=00:di=01;34:ln=01;36:pi=40;33:so=01;35:bd=40;33;01:cd=40;33;01:or=01;05;37;41:mi=01;05;37;41:ex=01;32:*.cmd=01;32:*.exe=01;32:*.com=01;32:*.bat=01;32:*.sh=01;32:*.csh=01;32:*.tar=01;31:*.tgz=01;31:*.arj=01;31:*.taz=01;31:*.lzh=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.gz=01;31:*.bz2=01;31:*.bz=01;31:*.tz=01;31:*.rpm=01;31:*.cpio=01;31:*.jpg=01;35:*.gif=01;35:*.bmp=01;35:*.xbm=01;35:*.xpm=01;35:*.png=01;35:*.tif=01;35:'</i>	Couleurs utilisées
<i>LTDL_LIBRARY_PATH=/usr/lib</i>	
<i>MACHTYPE=i386-redhat-linux-gnu</i>	Type de machine – SE se trouvant sur la machine
<i>MAILCHECK=60</i>	Définit la période en seconde de vérification des mail
<i>OPTERR=1</i>	Variable boléenne. Si cette variable contient la valeur 1, bash affiche les messages d'erreur
<i>OPTIND=1</i>	A chaque fois que le shell ou une procédure du shell est appelé(e) cette variable est initialisée à 1
<i>OSTYPE=linux-gnu</i>	Système d'exploitation par défaut
<i>PATH=/bin:/usr/bin:/usr/X11R6/bin:/usr/local/bin:/applications/bin:/home/etud/asi01/mchambre/mydos/dosemu</i>	Liste des répertoires à parcourir pour trouver un programme
<i>PIPESTATUS=([0]="")</i>	Variable-tableau contenant une liste des codes de retour des derniers processus exécutés en avant-plan
<i>PPID=1778</i>	ID du processus parent du shell
<i>PROMPT_COMMAND='echo -ne "\033]0;\${USER}@\${HOSTNAME}: \${PWD}\007"'</i>	

<code>PS1='[\u@h \W]\\$ '</code>	Prompt de shell primaire
<code>PS2='> '</code>	Prompt de shell secondaire
<code>PS4='+ '</code>	Prompt utilisé par l'option de trace
<code>PWD=/home/etud/asi01/mchambre</code>	C'est le répertoire de travail courant
<code>QTDIR=/usr/lib/qt-2.2.1</code>	
<code>QT_XFT=0</code>	
<code>SESSION_MANAGER=local/asi1248.insa-rouen.fr:/tmp/.ICE-unix/1595</code>	
<code>SHELL=/bin/bash</code>	Chemin d'accès au shell courant
<code>SHELLOPTS=braceexpand:hashall:histexpand:monitor:history:interactive-comments:emacs</code>	Liste des options activées du shell, séparées par des virgules
<code>SHLVL=2</code>	Incrémenté de 1 à chaque invocation d'une instance de bash
<code>SSH_ASKPASS=/usr/libexec/openssh/gnome-ssh-askpass</code>	
<code>TERM=xterm</code>	Type de terminal
<code>UID=596</code>	Contient l'U-ID (user identifiant) de l'utilisateur, initialisé au démarrage du shell
<code>USER=mchambre</code>	Login de l'utilisateur
<code>_ =ls</code>	
<code>i=/etc/profile.d/which-2.sh</code>	
<code>kdepath=/usr/bin</code>	Chemin pour lancer KDE
<code>langfile=/home/etud/asi01/mchambre/.i18n</code>	Répertoire où se trouvent les fichiers de langues
<code>sourced=1</code>	
<code>mc=()</code>	
<pre>{ mkdir -p ~/.mc/tmp 2>/dev/null; chmod 700 ~/.mc/tmp; MC=~/.mc/tmp/mc-\$\$; /usr/bin/mc -P "\$@" >"\$MC"; cd "`cat \$MC`"; /bin/rm "\$MC"; unset MC }</pre>	

Exercice 2 : Programmation en shell

1 -

```
#!/bin/bash
# mon premier script
clear
while :
do
echo 'Menu'
echo '[1] Affichage repertoire courant'
echo '[2] Liste des fichiers du repertoire'
echo '[3] Information sur un fichier'
echo '[4] Changement de repertoire'
echo '[5] n premieres lignes d'un fichier'
echo '[0] Fin'
echo 'Choix:'
read ch
case $ch in
```

```
0) exit 0
;;
1) pwd
;;
2) ls
;;
3) echo -n 'Nom du fichier:' ; read file ; ls -l $file
;;
4) echo -n 'Nouveau repertoire:' ; read rep ; cd $rep
;;
5) echo -n 'Nom du fichier:' ; read file
echo -n 'Nb de lignes a afficher:' ; read n
head -$n $file
;;
*) echo 'choix non propose'
esac
done
```

Commentaires :

Les actions effectués par le script sont assez claires et n'ont pas besoin d'être expliquées.
Voyons plutôt la syntaxe :

<i>echo</i>	sert à afficher du texte
<i>clear</i>	efface l'ecran
<i>while</i>	effectue une boucle while sans argument donc infinie
<i>do</i>	identificateur de début de boucle
<i>done</i>	identificateur de fin de boucle
<i>read</i>	lit la valeur entrée au clavier
<i>case</i>	effectue une action suivant la valeur du paramètre qu'on lui donne
<i>#) action</i>	effectue "action" lorsque le cas vaut #
<i>*) actionbis</i>	effectue "actionbis" lorsque que l'on a aucun des cas spécifiés au dessus
<i>esac</i>	identificateur de fin de case

Nous savons que (\$var) équivaut à (nom de variable "var") et que dans un script \$1 correspond au premier argument en entrée. On en déduit que :

<i>ls -l file</i>	nous donne les caractéristiques du fichier appelé file
<i>ls -l \$1 file</i>	nous affiche les caractéristiques du fichier appelé file avec l'option d'affichage contenue dans l et qui est le premier argument donné au script
<i>ls -l \$file</i>	nous affiche les caractéristiques du fichier contenu dans la variable file
<i>ls -l \$1 \$file</i>	nous affiche les caractéristiques du fichier contenu dans la variable file avec l'option d'affichage qui est le premier argument donné au script

2 –

On veut écrire un script qui concatène puis trie deux fichiers f1 et f2 dans un nouveau fichier f3, les noms des 3 fichiers étant passés en paramètre du script.

```
#!/bin/bash
# notre deuxieme script
clear
```

<i>cat \$1>\$3</i>	On envoie le texte de f1 dans f3
<i>cat \$2>>\$3</i>	On envoie le texte de f2 dans f3 à la suite du texte déjà existant
<i>sort \$3 -o \$3</i>	On trie les lignes du fichier f3 en envoyant la sortie de la commande sort dans le fichier lui-même (d'où la présence du -o)

3 –

* On veut écrire un script qui vérifie l'existence dans le répertoire courant du fichier dont le nom est passé en paramètre.

```
#!/bin/bash
# notre deuxieme script
clear
if (test -e $1);
then echo 'oui';
else echo 'non';
fi
```

* On veut écrire script qui indique si le nom passé en paramètre est un fichier ou un répertoire.

```
#!/bin/bash
# notre deuxieme script
clear
if (test -d $1);
then echo 'repertoire';
elif (test -f $1);
then echo 'fichier';
else echo 'c est rien'
fi
```

Les 3 scripts que nous avons réalisés fonctionnent mais ne sont pas optimisés. On peut en effet faire une vérification au début de chaque programme (que le nombre de paramètres est correct ou que les paramètres sont corrects par exemple).