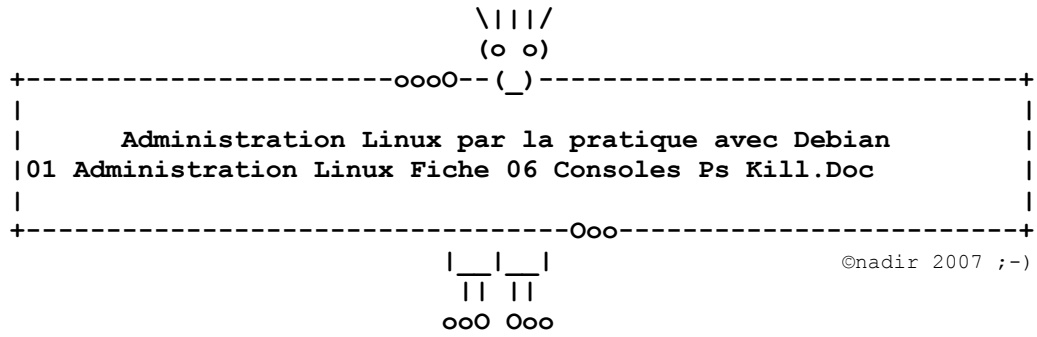

Sommaire

1. Shells

- **sh** ("Bourne shell", Steve Bourne,1976) shell historique (disponible sur tous les Unix) peu interactif (pas d'historique, pas de rappel de commandes, pas d'alias..)
- **csh** ("C-shell", Bill Joy, 1979) shell BSD (Berkeley,disponible sur tous les Unix) pallie les manques du sh, et reprend une syntaxe C.
- **ksh** ("Korn shell",David Korn,1986) shell normalisé POSIX (P1003.2) et ISO (HP, Solaris)- réécriture de sh en reprenant des fonctionnalités pratiques de csh
- **bash** ("Bourne again shell",Brian Fox et Chet Ramey,1989) le Ksh de Linux
- **zsh** ("Zero shell") Ksh intelligent (complétion de commandes)
- **tcsh** ("Toronto C-shell")

```
Ksh
Sh puis exit
```

Execution (shell est un interpréteur)
Par le shell courant : . fichier (symbole point)
Par un nouveau shell : « ksh fichier » ou « fichier » (si fichier est executable)

Shell par defaut:
Dans le fichier /etc/passwd le dernier champ contient le chemin du fichier exécutable
Modifier /etc/passwd ou chsh

2. Processus initialisation du shell

Init avant shell
Man login

Exec /etc/environnment	commun à tous les shells / vars d'environnement
Exec /etc/profile	commun à tous les shells hérités de bourne sh (donc tous les utilisateurs y compris root)
Exec /etc/bash.bashrc	commun à tous les shells bash
~/.bashrc ou .profile	spécifique utilisateur

3. Environnement

Variables d'environnement
Var=valeur
Ex : TOTO=mickey
puis echo $TOTO
echo permet d'afficher une variable ou une chaine.

Exemples
$PATH : liste des répertoires d'exécutables
$USER : nom de l'utilisateur
$HOME : repertoire de base de l'utilisateur

Liste:
env ou set : La liste des variables systèmes et personnalisées
essayer **echo** $PATH, **echo** $USER

Manipulation:
TOTO=$TOTO' mickey'
PATH=$PATH:$HOME/bin:.

Alias
Syntaxe: alias mot=commande ou alias mot='commande'

Historique des commandes
history
!! repeter la dernière commande
!n relancer la commande numéro **n**
!debut relancer la commande qui commence par « debut »

TP

1/ouvrir .bashrc. les lignes commencant par # sont des commentaires
Trouver les alias « ll » et « la » et les decommenter.

2/Ajouter l'alias « h » pour executer la commande history

3/
Modifier le ***.bashrc pour executer***
echo -n " bienvenue $USER ! Nous sommes le "; date
(-n : ne pas faire de retour à la ligne)

4/
Créer un fichier texte dans le home de mickey. Lui donner le droit d'execution.
Que faut il entrer pour l'executer ?
ajouter de manière permanente le répertoire courant aux chemins de recherches PATH

5/
Curiosité : le fichier .bash_logout. Modifier le pour qu'il affiche « ciao » au logout.
Cela marche t'il pour un su ?

4. User, process et jobs

users
Id
who whoami w

Su puis Exit

Process
Ps permet de connaître les processus actifs à un moment donné, associés à un terminal et lancés par l'utilisateur courant.
Exemple :

Ps –ef
Ps –ef |grep bash

Description des colonnes de la commande ps –ef :

- "USER" à quel utilisateur appartient le processus.
- "PID" est le numéro qui identifie le processus
- "%CPU" en % les ressources du microprocesseur utilisées par le processus.
- "%MEM" en % les ressources en mémoire vive utilisées par le processus.
- "RSS" mémoire réellement utilisée en ko par le processus.
- "START" l'heure à laquelle le processus a été lancé.

Kill permet d'envoyer un « signal » à un processus. Un signal n'est pas nécessairement un arrêt du programme.
La plupart du temps on se sert de « kill pid » pour terminer un programme.

Arreter un processus par ordre de gentillesse (faire un « man kill » ou « man signal » pour avoir la liste des signaux):
Kill pid
Kill -3 pid (QUIT)
Kill -9 pid (KILL)
Kill -15 pid (TERM)

Kill -1 ou –HUP pid demander au programme de relire sa configuration

1/
Essayer la commande pkill, et pgrep.
2/
Essayer la commande pstree

5. Jobs

Jobs: jobs associés à la console en cours
ctrl+c: tuer un process qui utilise la console
ctrl+z: susprendre un process qui utilise la console
fg: "foreground": repasser au premier plan le dernier programme suspendu
bg: "background": passer en arrière plan le dernier programme suspendu (il s'exécute)
&: "commande&" permet de lancer une commande directement en arrière plan, équivaut donc à ctrl+z + bg
Dans ces 2 derniers cas, il peut etre necessaire de « nettoyer l'ecran » : ctrl+l

Fg et bg peuvent prendre un parametre : %x où x est le numéro de job (donné par jobs)

1/
Essayer de lancer plusieurs vi en edition sur plusieurs fichiers, en les suspendant à la suite, en utilisant jobs et en reveillant celui de votre choix