

Prise en main de l'environnement Unix

Laurent Réveillère
laurent.reveillere@enseirb-matmeca.fr

2016-2017

1 Interprète de commandes

Le *shell* désigne un *interpréteur de commandes*. Son rôle est de faire l'interface entre l'utilisateur et le système Unix. Il permet d'exécuter les commandes de l'utilisateur et d'afficher le résultat à l'écran. Il existe plusieurs types de *shell*:

- le `/bin/sh` appelé *shell Bourne*
- le `/bin/bash` appelé *shell Bourne Again SHell*
- le `/bin/csh` appelé *C shell*
- le `/bin/ksh` appelé *Korn shell*
- le `/bin/tcsh` appelé *shell amélioré*

Une invite de commande primaire *Prompt String 1* (PS1) pour un utilisateur débute par le caractère `$` ou `>` (Bourne, Bash ou Korn shell) et `%` (C shell). En mode super utilisateur, le prompt (primaire) commence avec le caractère `#`. Un prompt secondaire *Prompt String 2* (PS2) est utilisé lorsque la commande en cours a besoin de plusieurs ligne.

Q1.1 Ouvrez un "TERMINAL" et testez la commande `exit`, puis réouvrez votre "TERMINAL".

Q1.2 Essayez les commandes suivantes `pwd`, `ls`, `who`, `date`, `cal`, `echo`. Commentez leur fonctionnement.

Historique des commandes. À l'aide des flèches haut et bas du clavier, l'utilisateur peut naviguer dans l'historique des commandes, les éditer pour les modifier de façon simple afin de lancer de nouvelles actions.

Q1.3 Essayez `man cal`. La commande `man` vous permet de consulter le manuel d'un exécutable. Essayez `man man`.

Q1.4 Il est parfois préférable de consulter le manuel en anglais qui peut être plus précis. Trouvez dans le manuel comment l'afficher en anglais (`man man`).

Q1.5 Essayer les combinaisons usuelles suivantes en mode édition.

C-p	commande précédente
C-n	commande suivante
C-a	début de ligne
C-e	fin de ligne
C-b	caractère précédent
C-f	caractère suivant
C-d	supprime le caractère sous le curseur ou exit si plus de caractère Δ
DEL	caractère précédent le curseur
C-k	couper jusqu'à la fin de ligne
C-u	couper à partir du début de la ligne
C-y	coller
RETURN	validation de la ligne
C-c	annulation de la ligne en cours

Combinaison de touches ou clé. La notation C-C représente l'appuyé simultané des touches CONTROL et C.

2 Visite de l'arborescence utilisateur

Q1.6 Essayez les commandes suivantes.

```
$ ls
$ pwd
$ cd_ . . (séparez la commande de ses paramètres par un espace représenté ici par _)
$ pwd
$ ls
```

Complétion (ou saisie rapide des commandes) : lors de la saisie d'une commande (ou d'un nom de fichier), la frappe de la touche Tab (tabulation) permet au shell de compléter automatiquement le nom de la commande (ou le nom du fichier). Cette reconnaissance automatique s'arrête dès qu'il peut y avoir ambiguïté et le shell attend la saisie de caractères supplémentaires pour continuer au prochain Tab. Cette méthode assure par ailleurs la validité des noms ainsi invoqués.

Q1.7 Exécutez la suite de commandes.s

```
$ cd
$ cd_ Desktop (n'oubliez pas le mécanisme de complétion)
$ pwd
```

Maintenant pour aller dans votre répertoire d'accueil plusieurs possibilités :

```
$ cd_ . . ou
$ cd_ ~ ou encore
```

```
$ cd
```

Pour afficher les fichiers cachés, utilisez l'option `-a` (fichiers dont le nom commence par un `.` tels que `.bashrc`).

Q1.8 Exécutez, en utilisant les mécanismes d'édition de bash, les commandes

```
$ man ls
$ ls Desktop
$ ls -a Desktop
$ ls -l Desktop
```

3 Création et suppression de répertoires

La commande `mkdir` (make directory) permet de créer un répertoire. Lors de la création de ce répertoire, deux fichiers spéciaux sont créés : `.` et `..`. Ces deux fichiers spéciaux sont des liens vers le répertoire courant et le répertoire "parent" utile pour l'exécution de certaine commande prenant un répertoire en argument.

Q1.9 Créez un répertoire `tmp` et un répertoire `Projets` dans son répertoire d'accueil. Plusieurs possibilités, par exemple :

```
$ mkdir ~/tmp
$ cd
$ mkdir Projets
$ cd Projets
$ ls -als
$ pwd
$ cd ..
$ pwd
```

Si le chemin est celui d'un fichier (ou répertoire) déjà présent, cela provoque une erreur : le nom de base doit être nouveau dans le répertoire de référence.

```
$ mkdir eclair
$ ls -F
$ cd eclair
$ rmdir ../eclair
$ cd ..
$ rmdir eclair
$ ls -F
```

4 Premier pas avec Emacs

Exécuter la commande `emacs`. Emacs est un éditeur de texte très largement utilisé dans le monde Unix et hautement configurable. Il existe de nombreux raccourcis clavier qui facilite l'utilisation de l'éditeur. Voici quelques raccourcis commun :

Ctrl-x Ctrl-f: ouvrir un fichier
Ctrl-x Ctrl-s: sauver
Ctrl-x Ctrl-w: sauver sous
Ctrl-x Ctrl-c: quitter emacs
Ctrl-K : supprime tous les caractères jusqu'à la fin de la ligne
Sélection à la souris puis Ctrl-W : supprime la zone sélectionnée.
Ctrl-space: Pour activer le debut d'une sélection
Ctrl-s: recherche en avant
Ctrl-r: recherche en arrière
Ctrl-g: annuler la dernière commande
Meta-/: complétion automatique
Ctrl-_: undo circulaire (défaire)
Ctrl-x b: passer d'un buffer à un autre
Ctrl-x k: tuer un buffer

La combinaison M-x permet d'exécuter une commande. Tous les raccourcis précédents sont associés à des commandes. Ainsi, Ctrl-x Ctrl-f correspond en fait à la commande *find_file*

Q1.10 Utiliser emacs pour créer un fichier LISEZ_MOI dans votre répertoire principal contenant un petit texte de votre choix.

5 Manipulation de fichiers

5.1 Copies

Q1.11 Exécuter et commentez les commandes suivantes :

```
$ mkdir test
$ cd test
$ touch doc1.html
$ touch doc2.html
$ ls
$ cd
$ cd test
$ cp doc1.html doc2.html ~/tmp
```

Copie de fichier. Lorsque le dernier argument de cp est un chemin valide ne correspondant à aucun fichier ou répertoire, un nouveau fichier est créé. Sinon, le fichier qui existe est écrasé.

5.2 Suppressions

Q1.12 Essayez la suite de commandes :

```
$ rm ~/tmp/README.html
```

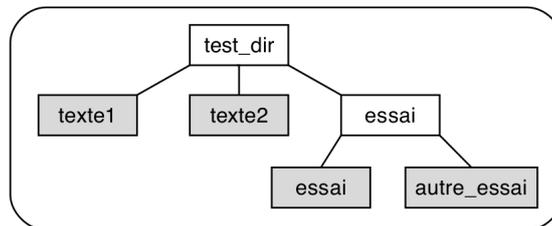
```
$ cd ~/tmp
```

⚠ Certaines options de la commande peuvent avoir des conséquences importantes. Utiliser par exemple la commande `rm -fr` avec précaution. Regarder le manuel pour plus d'information.

5.3 Déplacements et renommages

```
$ cd ~/test
$ mv doc1.html doc1.html
$ ls -F
$ mv doc1.html ../doc1.html
$ ls -F
$ ls -F ..
$ mv ../doc1.html .
$ ls -F ..
$ ls -F
$ mv *.html ~/tmp
$ ls -F
$ mv ~/tmp/*.html .
```

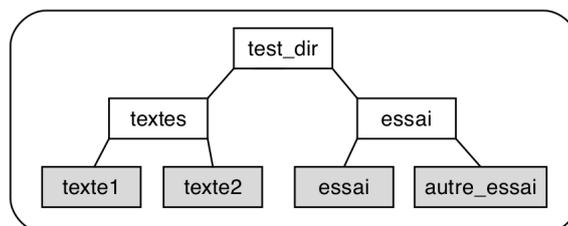
Q1.13 Créez à partir de votre répertoire d'accueil l'arborescence :



Les cases blanches sont des répertoires, tandis que les cases grisées sont des fichiers texte à créer. La marche à suivre pour cet exercice est la suivante :

- créez les répertoires;
- lancez emacs à partir du répertoire d'accueil;
- éditez les fichiers et sauvegardez-les.

Q1.14 Placez-vous dans le répertoire `test_dir`, et ramenez-vous en deux commandes à :



Q1.15 Depuis `test_dir` essayez `cp essai textes`; commentez le résultat. Que fait dans ce même

cas cp * Essayez avec l'option -R. Commentez les erreurs possibles lors de l'exercice précédent :
\$ mv t* textes
\$ cp t* textes

6 Sessions en mode texte sous linux

Dans certaines situations, il peut s'avérer nécessaire d'ouvrir une session linux en mode texte. Ce type de session donne accès aux mêmes outils que la console mais ne permet pas de lancer d'application graphique.

Q1.16 Ouvrir une session en mode texte par la combinaison de touches CONTROL-ALT-F1. Saisir son identifiant puis son mot de passe.

Q1.17 Créer un nouveau fichier dans votre répertoire principal. Repasser en mode graphique en utilisant la combinaison de touches CONTROL-ALT-F7. Il est possible d'avoir plusieurs sessions en mode texte différentes en utilisant CONTROL-ALT-F2, CONTROL-ALT-F3, etc. Ne pas oublier de quitter votre session en utilisant la commande `exit`.

7 Connexion à une machine distante

Le système unix permet de se connecter à des serveurs distants au moyen de la commande en ligne `ssh` (ou Secure Shell) qui est un protocole de communication sécurisé. La seule machine de l'école accessible depuis l'extérieure s'appelle `ssh.enseirb-matmeca.fr`

Q1.18 Ouvrir un terminal et se connecter à la machine `ssh` en utilisant la commande :

```
ssh login@ssh.enseirb-matmeca.fr.
```

Utiliser la commande `hostname` pour afficher le nom de la machine sur laquelle vous êtes connecté.

Q1.19 Il est possible de copier des fichiers entre deux machines distantes grâce à la commande `scp` qui s'utilise comme la commande `cp`. Identifier une autre machine de la salle et essayer de copier des fichiers localement dans le répertoire `/tmp` de la machine distante.

Dans l'utilisation de la commande `scp`, les noms de fichiers ou répertoires source ou destination sont préfixés par le nom de la machine distante suivie du signe `'.'`. Par exemple, la commande :
`scp foo.txt reveille@ssh.enseirb-matmeca.fr:/tmp/` copie le fichier `foo.txt` du répertoire courant dans le répertoire `/tmp` sur la machine distante `ssh` de l'enseirb.

8 Personnalisation de l'environnement

8.1 Bash

Le fichier `.bashrc` qui se trouve dans votre répertoire personnel est exécutée avant chaque lancement de `bash`. Il contient les informations pour configurer votre shell `bash` selon votre gout. Ajoutez par exemple les lignes suivantes dans votre fichier `.bashrc`

```
alias c=clear
alias m=more
alias cl='clear; more'
```

Lancez un nouveau shell bash et essayez les nouvelles commandes `c`, `m` et `cl`. Si vous voulez, par exemple, que les fichiers qui sont dans le répertoire `bin` de votre répertoire d'accueil soient directement accessibles, ajoutez le chemin dans la variable `PATH` comme suit :

```
export PATH=$PATH:$HOME/bin
```

8.2 Emacs

Le fichier `.emacs` qui se trouve dans votre répertoire personnel est exécutable avant chaque lancement d'`emacs`. Il contient les informations pour configurer votre `emacs` selon votre goût. Voici quelques commandes que vous pouvez ajouter et qui rendent la vie plus simple.

```
;; afficher la selection
(transient-mark-mode 1)

;; pas de pub...
(setq inhibit-startup-message t)

;; Pour avoir les couleurs
(global-font-lock-mode t)
(turn-on-font-lock)

;; Pour voir ou sont les debut et fin de parenthese
(show-paren-mode)

;; Quelque raccourcis clavier personnel
(global-set-key [insert] 'overwrite-mode)

(global-set-key [delete] 'delete-char)
(global-set-key [C-delete] 'kill-word)
(global-set-key [M-delete] 'kill-paragraph)

(global-set-key [backspace] 'delete-backward-char)
(global-set-key [C-backspace] 'kill-word)
(global-set-key [M-backspace] 'kill-paragraph)

(global-set-key [home] 'beginning-of-line)
(global-set-key [C-home] 'beginning-of-buffer)
(global-set-key [end] 'end-of-line)
(global-set-key [C-end] 'end-of-buffer)

(global-set-key [prior] 'scroll-down)
(global-set-key [C-prior] 'beginning-of-buffer)
(global-set-key [next] 'scroll-up)
```

```
(global-set-key [C-next] 'end-of-buffer)

(global-set-key [left] 'backward-char)
(global-set-key [C-S-left] 'backward-word)
(global-set-key [M-left] 'backward-kill-word)
(global-set-key [right] 'forward-char)
(global-set-key [C-S-right] 'forward-word)
(global-set-key [M-right] 'kill-word)

(global-set-key [up] 'previous-line)
(global-set-key [C-S-up] 'backward-paragraph)
(global-set-key [down] 'next-line)
(global-set-key [C-S-down] 'forward-paragraph)

(global-set-key [f2] 'save-buffer)
(global-set-key [f3] 'find-file)
```