Unix : Utilisation et programmation

Pr. Hajar IGUER,
hajar.iguer@uic.ac.ma

Unix®
Operating System

Editeur de texte

- Un éditeur de texte est un programme qui permet de modifier des fichiers de texte brut sans mise en forme particulière.
- Exemples:
  - Emacs
  - Vim (Vi Improved)
  - Gedit
  - GNU Nano

Commandes de base pour Emacs

Crtl a	début de la ligne
Ctrl e	fin de la ligne
Ctrl d	supprime le caractère sous le curseur
Ctrl k	supprime les caractères jusqu'à la fin de la ligne
Ctrl r	recherche vers le début du fichier
Ctrl s	recherche vers la fin du fichier
Ctrl y	copie le buffer
Ctrl space	marque le début d'un buffer
Ctrl w	marque la fin d'un buffer et le supprime
Esc w	marque la fin d'un buffer et le copie
Esc x	commande ligne
Ctrl x Ctrl s	sauve le fichier
Ctrl x Ctrl w	sauve le fichier sous un autre nom
Ctrl g	annule la commande en cours

Commandes de base pour Vim

Cursor movement

h - move left
j - move down
k - move up
l - move right
w - jump by start of words (punctuation considered words)
W - jump by words (spaces separate words)
e - jump to end of words (punctuation considered words)
E - jump to end of words (no punctuation)
b - jump backward by words (punctuation considered words)
B - jump backward by words (no punctuation)
0 - (zero) start of line
^ - first non-blank character of line
$ - end of line
G - Go To command (prefix with number - 5G goes to line 5)
Note: Prefix a cursor movement command with a number to repeat it. For example, 4j moves down 4 lines.

Insert Mode - Inserting/Appending text

i - start insert mode at cursor
I - insert at the beginning of the line
a - append after the cursor
A - append at the end of the line
o - open (append) blank line below current line
(no need to press return)
O - open blank line above current line
ea - append at end of word
Esc - exit insert mode

Editing

r - replace a single character (does not use insert mode)
J - join line below to the current one
cc - change (replace) an entire line
cw - change (replace) to the end of word
c$ - change (replace) to the end of line
s - delete character at cursor and subsitute text
S - delete line at cursor and substitute text (same as cc)
xp - transpose two letters (delete and paste, technically)
u - undo
. - repeat last command

Marking text (visual mode)

v - start visual mode, mark lines, then do command (such as y-yank)
V - start Linewise visual mode
o - move to other end of marked area
Ctrl+v - start visual block mode
O - move to Other corner of block
aw - mark a word
ab - a () block (with braces)
aB - a {} block (with brackets)
ib - inner () block
iB - inner {} block
Esc - exit visual mode

Visual commands

> - shift right
< - shift left
y - yank (copy) marked text
d - delete marked text
~ - switch case

Cut and Paste

yy - yank (copy) a line
2yy - yank 2 lines
yw - yank word
y$ - yank to end of line
p - put (paste) the clipboard after cursor
P - put (paste) before cursor
dd - delete (cut) a line
dw - delete (cut) the current word
x - delete (cut) current character

Exiting

:w - write (save) the file, but don't exit
:wq - write (save) and quit
:q - quit (fails if anything has changed)
:q! - quit and throw away changes

Search/Replace

/pattern - search for pattern
?pattern - search backward for pattern
n - repeat search in same direction
N - repeat search in opposite direction
:%s/old/new/g - replace all old with new throughout file
:%s/old/new/gc - replace all old with new throughout file with confirmations

Working with multiple files

:e filename - Edit a file in a new buffer
:bnext (or :bn) - go to next buffer
:bprev (of :bp) - go to previous buffer
:bd - delete a buffer (close a file)
:sp filename - Open a file in a new buffer and split window
ctrl+ws - Split windows
ctrl+ww - switch between windows
ctrl+wq - Quit a window
ctrl+wv - Split windows vertically

Installation d'un éditeur de texte

- Pour installer un éditeur de texte:
  - sudo apt-get install vim s'il existe au niveau de votre distribution
- Pour ouvrir un fichier:
  - nano fich1.txt
  - vim fich1 , il sera créé s'il n'existe pas
  - emacs nomdufichier
- Pour démarrer l'éditeur de texte:
  - nano

Lien physique et symbolique

- ln <fich1> <fich2> : (LiNk) crée un lien physique du fichier <fich1> vers <fich2>.

- ln –s : crée un lien symbolique au lieu d'un lien physique. On peut créer un lien symbolique d'un répertoire vers un autre.

PROGRAMMATION SHELL

Shell: Définition

- Un Shell est un programme exécuté par le système.

- Il a un double rôle:
  - Interpréteur de commandes
  - Langage de programmation

Shell: Définition

- L’interpréteur de commandes Shell :
  - initialise l’environnement de l ’utilisateur
  - affiche un prompt,
  - lit et exécute les instructions saisies par l’utilisateur

Shell: Définition

- Lorsqu'une commande est saisie, le Shell:
  - interprète les variables et les métacaractères
  - gère les redirections et les tubes (« pipe »)
  - interprète la commande
  - exécute la commande

Types de Shell

- Il existe plusieurs Shell, sur les systèmes de type Unix, dont les plus connus sont les suivants :
  - Bash (le Bourne Again SHell).
  - Sh (le bourne SHell) : Ce Shell fut le premier présent dans les systèmes Unix et est à l'origine de tout les autres.
  - Ksh (Korn SHell).
  - Csh (le C SHell) : dont la syntaxe est inspirée du langage C).
  - Tcsh (le Tenex C Shell) : C'est une évolution du Csh.
  - Zsh (le Z Shell) : C'est un Shell récent basé sur Bash, Ksh et Tcsh.

Script

- Un script est un fichier contenant un ensemble de commandes exécutées séquentiellement
  - Sous forme de fichier texte contenant les commandes
- Le script ne peut être exécuté que par un interpréteur
  - « /bin/bash » pour le bash
- Le langage de script Shell est un langage évolué offrant de nombreuses possibilités
  - Boucles, variables, tests avec if, création de fonctions, …

Script

- Dans quels cas utilise-t-on les scripts ?

– Pour effectuer un travail répétitif
– Pour des tâches d'administration système
– Pour installer des programmes
– Au démarrage du système pour démarrer les services et applications

Structure d'un script

- Toutes les instructions et commandes sont regroupées au sein d'un script.
- Lors de son exécution, chaque ligne sera lue et exécutée par ordre séquentiel.
- Une ligne peut se composer de commandes internes ou externes, de commentaires ou être vide.
- Par convention les Shell scripts se terminent généralement (pas obligatoirement) par:
  - «.sh» pour le Bourne Shell et le Bourne Again Shell,
  - « .ksh » pour le Korn Shell
  - « .csh » pour le C Shell.

Structure d'un script

- Une ligne de commentaire commence toujours par le caractère « # ». Un commentaire peut être placé en fin d'une ligne comportant déjà des commandes.
  - # La ligne suivante effectue un ls
  - ls # La ligne en question
- La première ligne a une importance particulière car elle permet de préciser quel shell va exécuter le script
  - #!/usr/bin/sh
  - #!/usr/bin/ksh
  - Dans le premier cas c'est un script Bourne, dans l'autre un script Korn.

Exécution d'un script

- Quand un script est lancé, un nouveau shell « fils » est créé qui va exécuter chacune des commandes.
- Si c'est une commande interne, elle est directement exécutée par le nouveau shell.
- Si c'est une commande externe, dans le cas d'un binaire un nouveau fils sera créé pour l'exécuter, dans le cas d'un shell script un nouveau shell fils est lancé pour lire ce nouveau shell ligne à ligne.

Exécution d'un script

- Pour rendre un script exécutable directement
  - $ chmod u+x monscript
- Pour exécuter le script, il faut appeler l'interpréteur
  - $ bash monscript
- Possibilité de simplifier l'appel en ajoutant la ligne suivante en tête du script
  - #!/bin/bash
- L'appel est alors plus simple
  - $ ./monscript
  - $ ./home/ginfo/scripts/fich.sh

Variables

- Une variable du Shell est l'assignation d'un nom à un contenu un nom associé à une valeur
- Par défaut, toutes les variables sont initialisées à NULL
- Certaines variables sont prédéfinies dans le SE
- les utilisateurs peuvent définir et utiliser leurs propres variables

Variables prédéfinies

- Quelques variables prédéfinies:
  - HOME: répertoire de connexion
  - PATH: répertoire à inspecter pour trouver les commandes saisies par l'utilisateur
  - TERM: type du terminal utilisé
  - PWD: chemin courant
  - SHELL: nom du Shell utilisé
  - USER: nom de connexion
- En Shell, pour désigner le contenu d'une variable, on écrit le nom de la variable précédé du signe dollar.
  - echo $HOME

Exercice d'application

- Ecrivez un script qui permet de:
  - Afficher le nom de l'utilisateur de la session ouverte
  - Afficher le chemin complet au dossier personnel
  - Afficher le message suivant : « Bienvenue »