Basic Tools for System Management

Frederic.Mallet@unice.fr

Université Nice Sophia Antipolis
Membre de Université Cote d’Azur

Objectives

- Learn how to manage your own computer
  - Using the shell
    - Linux / Windows
  - Version control and git
  - Installing a Virtual Machine
    - Linux Mint 18.3
  - **Using bash**
  - Eclipse: JDT and Modeling

The “shell” processes

- Launched when
  - Successful login (see /etc/passwd)
  - When opening a terminal window
  - => bash
- Lots of variables
  - USER: user login
  - PS1: prompt
  - HOME: user home directory
  - HOSTNAME: the computer name
  - PATH: list of path to find executable files, separator is ‘:’
    - which man

Variables

- Display value:
  - echo $<variable-name>
  - echo ${<variable-name>:-value}
  - echo ${<variable-name>:=value}
  - echo ${<variable-name>:offset} (note negative offset)
  - echo ${<variable-name>:offset:length}
- Change value:
  - <variable-name>=<value>
- Make variables global:
  - EXPORT <variable-name>

Configuration files

- When login
  - Execute ~/.profile if exists
  - Execute ~/.bashrc if exists
- Alias
  - alias ll=’ls -l’
- Expr evaluates arithmetic expressions
  - expr 2 + 3

Regular expressions

- In a shell
- Replace a full string by a shorter expression
  - “*”: replace any string of characters
  - “?”: replace any character
  - [oai]: list set of possible characters

Standard regular expression

- In commands like sed, expr, grep
  - ‘.’ any character
  - [aiu] list of characters
  - [A-Z] ou [A-Z0-9] sequence of letters
  - [^0-9] characters that do not match
  - ‘*’ repeat a pattern several times (including 0)
  - ‘\’ escape character .*\.jpg
  - Sub-part: expr ‘foo123.jpg’ : ‘foo\(.*\)\.jpg’
- Sed
  - sed -e ‘s/pattern/replacement/g’ -e ‘s/pattern/replacement/1’
  - Use \(\) and \1 \2 to identify sub parts of a pattern

Shell scripts

- Sequence of bash instructions in ‘.sh’ file
  - Start with “#! /bin/bash”
  - Update PATH
  - Must be executable (chmod)
- Parameters
  - $n : n th parameter
  - $# : number of parameters
  - $* : “$1 $2 ...”
  - Shift: shift the parameters

Control flow

- # test
  - if […]
    - then …
    - else …
  - fi
- #repeat
  - while […]
  - do
    - …
  - done

- # repeat
  - for variable in list
  - do
    - …
  - done
- #case
  - case expression in
  -   pattern1) #simple
    - …
    - ;;
  - esac

See man test

Function

```
#! /bin/bash
foo () {
  if [ "$1" = "$2" ]
    then echo "$1"
    else echo "$2$1"
  fi
}
case "$#" in
   0|1) echo "need more than one !!"
          exit 1 ;;
   *) echo "prefix is \"$1\""
          prefix="$1"
          shift
esac
```

```
for suffix in $*
do
    name=‘foo $suffix $prefix‘
    if [ -f $name ]
    then echo $name
    fi
done
```