Linux Command Line

Abdullah Zubaer, University of Manitoba. zubaera@myumanitoba.ca

Redirecting output

- There are three ways we can redirect output

  “>” redirects the output to a file (that also overwrite the file)

  “>>” redirects and append the output to a file

  “|” redirects output of one command and use it as an input of the next command

Example of Redirecting output

- “ls” shows the content of a directory in the terminal (standard output), but it can be written into a file with “>” or “>>”

  such as, ls /lib > my_text_file.txt

  or, ls /etc >> my_text_file.txt

  “>>” appends the output to the file, where “>” wipes the previous content of the file and overwrite it.

Example of Redirecting output

- A simple example of “|” would be

  ls /etc | less

  - Here the output of the “ls” command is used as an input of the “less” command
  - You can be creative to use this. It can be a powerful tool if you can use it creatively

Wild-card

- Symbol: *
- Function: it means “any character” and is used for matching a string (e.g., file name)
- Example:
  - ls *.pdf

    that is to list all the pdf files in a directory

Searching texts

- Command name: grep
- Global Regular Expression Print (grep)
- Function: search for a string
- Usage:
  - grep pattern file_name
  - grep -option pattern file_name
- Example:
  - grep ‘atg’ myfile
  - grep -c ‘atg’ myfile [-c is for counting matches]
  - grep -v ‘atg’ myfile [-v is for inverse match]

More commands to check data

- cut – extract character of field from a file
  - cut -option file-name
- sort – alphabetically sort a lines of a file
  - sort file-name.txt
- diff – show differences between files
  - diff file-1 file-1
- wc – word count of a text file
  - wc file-name.txt

Concatenation (once more!)

- “cat” can concatenate text into a file
- It can also concatenate multiple files together
- Usage:
  - cat file_1 file_2 file_n > combined_file

Writing scripts

- Shell script is just a set of commands in a file
- Example of a simple shell script:

```
#!/bin/bash
date
echo “This is my first shell script”
```

Zipping and archiving

- Zipping
  - Formats: gz, bz, bz2
  - gzip and gunzip commands
  - Syntax: gzip folder-name

- Archiving
  - tar
  - Example: tar -zxvf file-name.tar.gz

File permission

- Every file is secured by its permission specific to a user or its usage (e.g., read, write or executable)
- File permission can be checked with “ls -l” command

- Changing mode of a regular file to an executable file -

  chmod +x file-name

- Running an executable file -

  ./file-name

Installing a program

- Installing a program from its source code
  - ./configure
  - Make
  - sudo make install

- sudo = superuser do
- sudo gives a user the administrative/root authorization

Executing a program

- If a program in not install that can be executed by running its executable file from the program’s directory
- If the program is installed in the system (that means it has a copy in the “/bin” or other “bin” folder), can be run by typing its name
- Syntax:
  - program-name
  - program-name file-name
- Example:
  - gedit
  - gedit newfile

Accessing servers

- Command name: ssh
- Function: Connect to a remote server through a secure shell
- Syntax: ssh username@serveraddress
- Example:
  - ssh user@grex.computecanada.ca

Getting help

- To know more about the available options and other information of a command, “-h” or “--help” option is available
  - Syntax: command --help
- Another way to get more information about a command is to access its manual or man page
  - Syntax: man command
- Example:
  - cut --help
  - man cut