Introduction to Linux Shell

Elements of Programming Languages

Emidio Capriotti
http://biofold.org/

Biomolecules Folding and Disease

Department of Pharmacy and Biotechnology (FaBiT)
University of Bologna

Shell and Terminal

In computing, a shell is a computer program that exposes an operating system's services to a human user or other programs.

The terminal is a program that opens a graphical window and lets you interact with the shell.

In this course we will consider terminals that run Unix-like shells available by default on Linux and Mac operating systems.

In terminals the interaction with operating system occurs mainly by command line. One the most common shell used in linux is bash.

If you have a machine running Windows, it is suggested to install ubuntu with Windows Subsystem for Linux (WSL).

Linux Basics

Important directories in the linux operating systems are the root directory (/) the home (/home). Executable programs are /bin and /usr/bin.

Commands to navigate and show the file content in the operating system are:

- cd: change directory
- pwd: current directory
- ls: display file list

Commands for manipulating files and directories are:

- mv: move file
- cp: create a copy of the file
- mkdir: make a new directory
- rm: remove file of directory
- cat/less: print the content of a file

Users and Packages

Regular linux user have read/write permissions only on specific directories such as /home.

To write file in different locations of the filesystem the user should have sudo (super user do) permissions.

```
emidio@S968-01D20-W01:~$ ls -ltr ~/
drwx------@  93 emidio  staff        2976 Jun 24 20:13 Library
drwx------@  61 emidio  staff        1952 Jun 27 11:30 Documents
drwx------@ 358 emidio  staff       11456 Jul  3 05:56 Downloads

emidio@S968-01D20-W01:~$ ls -ltr ~/
drwxr-xr-x  70 root  wheel  2240 May 30 22:45 Library
drwxrwxr-x  55 root  admin  1760 Jun 26 18:16 Applications
drwxr-xr-x   3 root  wheel    96 Jul  2 13:03 Volumes
```

To install specific tools on a linux operating system, specific package management utilities are available. On ubuntu it is apt (install/uninstall). On mac brew and macports are used.

Python packages

By default on a linux operating system only the basic python package is installed.

To install specific python package the pip command id used. pip is the package installer for Python.

If *numpy* package is not available with the following command it can be installed:

```
emidio@S968-01D20-W01:~$ pip install numpy
Defaulting to user installation because normal site-packages is not
writeable
Requirement already satisfied: numpy in ./Library/Python/3.9/lib/.
python/site-packages (1.24.1)
```

Resources and processes

To display the free disc space of a specific file system the df command is used while the command free allow to show the RAM available on your machine

```
emidio@S968-01D20-W01:~$ df -h
Filesystem       Size   Used  Avail Capacity iused       ifree    %iused  Mounted on
/dev/disk3s1s1  1.8Ti  8.5Gi  1.3Ti     1%  356093  4294025914    0%   /
devfs           208Ki  208Ki    0Bi   100%     720           0  100%   /dev
/dev/disk3s6    1.8Ti   30Gi  1.3Ti     3%      30 13588623200    0%   /System/Volumes/


emidio@S968-01D20-W01:~$ free -mh
               total        used        free      shared  buff/cache   available
Mem:            94G        8.0G        7.9G        5.6M         78G         85G
Swap:           31G        372M         31G
```

The top (table of processes) command shows a real-time view of running processes in Linux and displays kernel-managed tasks.

Process control

Processes are identified by a unique PID (process identification number) on a Linux or Unix-like operating system. A PID is automatically assigned to each process when it is created. A process is killed with the command kill that takes in input its PID.

```
emidio@S968-01D20-W01:~$ ps aux
USER               PID  %CPU %MEM      VSZ    RSS   TT  STAT STARTED      TIME COMMAND
emidio           59247   4.1  0.9 39216788 299272   ??  S     1:32PM   0:10.76 /Applicat

emidio@S968-01D20-W01:~$ kill 59247
```

Process can run in background adding the Ampersand (&) character at the end. The jobs command allow to visualise the process running in a shell.

```
emidio@S968-01D20-W01:~$ jobs
[1]-  Running                 /Applications/Zotero.app/Contents/MacOS/zotero &
[2]+  Running                 sleep 30 &
```

The combination of Ctrl+C and Ctrl+Z respectively kills and suspends a running job.

The grep command

Grep command can be used to find or search a regular expression or a string in a text file. Let's generate a file with two columns containing the name of the customers of a shop and the amount of their purchase (*purchase.tsv*). To search if someone is a customer of the shop we can use grep.

```
emidio@S968-01D20-W01:~$ grep -i emidio purchase.tsv
Emidio   12.0
Emidio   20.0
```

The caret (^) and dollar ($) symbols allow to match words at the beginning or the and of the line.

```
emidio@S968-01D20-W01:~$ grep ^P purchase.tsv
Paola    20.0
Piero    30.0
Piero    20.5

emidio@S968-01D20-W01:~$ grep ^5$ purchase.tsv
Piero    20.5
```

The sort command

The sort command arranges the records in a particular order. By default, the sort command sorts file assuming the content is ASCII. Using options in the sort command can also be used to sort numerically. The option *-k* allow to sort based on the elements present on a specific column

```
emidio@S968-01D20-W01:~$ sort -k 1 purchase.tsv
Emidio   12.0
Emidio   20.0
Kashaf   5.0
Nicola   30.0
Paola    20.0
Piero    20.5
Piero    30.0
Young    40.0

emidio@S968-01D20-W01:~$ sort -nk 2 purchase.tsv
Kashaf   5.0
Emidio   12.0
Emidio   20.0
Paola    20.0
Piero    20.5
Nicola   30.0
Piero    30.0
Young    40.0
```

Piping commands

A pipe (|) is a form of redirection that is used in Linux and other Unix-like operating systems to send the output of one command/program/process to another command/program/process for further processing.

To count the number of purchases of the customer *Emidio* a combination of the *grep* and *wc* commands can be used

```
emidio@S968-01D20-W01:~$ grep Emidio purchase.tsv | wc -l
      2       4      24
```

To list the unique customers of the shop, a combination of the *cut* and *sort* commands can be used.

```
emidio@S968-01D20-W01:~$ cut -f 1 purchase.tsv | sort -u
Emidio
Kashaf
Nicola
Paola
Piero
Young
```

wget and curl

Curl and wget are both command-line tools used to retrieve data from internet. Wget, primarily supports HTTP and FTP protocols while curl a larger number of protocols.

```
emidio@S968-01D20-W01:~$ wget https://rest.uniprot.org/uniprotkb/Q92624.fasta
--2023-07-03 21:32:17--  https://rest.uniprot.org/uniprotkb/Q92624.fasta
Resolving rest.uniprot.org (rest.uniprot.org)... 193.62.193.81
Connecting to rest.uniprot.org (rest.uniprot.org)|193.62.193.81|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/plain]
Saving to: ‘Q92624.fasta’

Q92624.fasta              [ <=>                 ]     696  --.-KB/s    in 0s

2023-07-03 21:32:18 (94,8 MB/s) - ‘Q92624.fasta’ saved [696]

emidio@S968-01D20-W01:~$ curl https://folding.biofold.org/k-pro/api/pdb/1imq
[
  {
     "PROTEIN": "Im9*",
     "SOURCE": "Escherichia coli",
     "LENGTH": 86.0,
     "UniProt": "P13479",
     "PDB_wild": "1imq",
```

Bash script

A bash script is a series of commands written in a file. These are read and executed by the bash program. The script is characterised by the shebang (#!/bin/bash) which represents the absolute path to the bash interpreter.

A bash script can be written for downloading protein sequences from a specific repository.

```
emidio@S968-01D20-W01:~$ cat ./uniprot_seq.sh
#!/bin/bash
pid=$1
wget https://rest.uniprot.org/uniprotkb/$pid.fasta

emidio@S968-01D20-W01:~$ ./uniprot_seq.sh Q92624
Resolving rest.uniprot.org (rest.uniprot.org)... 193.62.193.81
Connecting to rest.uniprot.org (rest.uniprot.org)|193.62.193.81|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 696 [text/plain]
Saving to: ‘Q92624.fasta’

100%[=====================================================================>] 696
–.–K/s   in 0s

2023-07-04 13:11:11 (32.2 MB/s) – ‘Q92624.fasta’ saved [696/696]
```

For loop

A bash for loop is a bash programming language statement which allows code to be repeatedly executed. A for loop is classified as an iteration statement i.e. it is the repetition of a process within a bash script.

A for loop can be used to download the sequence of a set of proteins from the same repository.

```
emidio@S968-01D20-W01:~$ cat list_pids.txt
 P05067
 P51693
 Q06481
 Q92624

 emidio@S968-01D20-W01:~$ for i in `cat list_pids.txt`
 > do
 > wget https://rest.uniprot.org/uniprotkb/$i.fasta
 > done
 --2023-07-04 13:29:41--  https://rest.uniprot.org/uniprotkb/P05067.fasta
 Resolving rest.uniprot.org (rest.uniprot.org)... 193.62.193.81
 Connecting to rest.uniprot.org (rest.uniprot.org)|193.62.193.81|:443... connected.
 HTTP request sent, awaiting response... 200 OK
 Length: unspecified [text/plain]
 Saving to: ‘P05067.fasta’
```