



TRƯỜNG ĐẠI HỌC BÁCH KHOA – ĐẠI HỌC QUỐC GIA TP.HCM
TRUNG TÂM KỸ THUẬT ĐIỆN TOÁN

268 Lý Thường Kiệt, Phường 14, Quận 10, TP.Hồ Chí Minh
Điện thoại: 84-8-3864 7256 ext 5371 – Fax: 84-8-3865 8687
Website: www.cce.hcmut.edu.vn
E-mail: dientoan@hcmut.edu.vn

Lab 1

Giới thiệu Linux CLI và Bash Script

Ths.Hoàng Lê Hải Thanh
Phòng Thí nghiệm Tính toán Hiệu năng cao, Trường Đại học Bách Khoa, ĐHQG TP.HCM
thanhhoang@hcmut.edu.vn

I. MỤC ĐÍCH

- Làm quen việc tương tác với hệ điều hành Linux qua giao diện dòng lệnh
- Học cách lập trình các tác vụ Linux cơ bản với Bash Script

II. MỤC TIÊU

- Hiểu rõ lịch sử phát triển của giao diện dòng lệnh (CLI) trên Linux
- Sử dụng thông thạo các câu lệnh phổ biến trên Linux
- Biết cách xây dựng một chương trình dòng lệnh bằng Bash Script

III. CHUẨN BỊ

Để phục vụ cho bài thực hành này, học viên phải chuẩn bị cài đặt hệ điều hành Linux lên trên máy tính cá nhân (Windows/MacOS) thông qua 3 cách sau (Lưu ý: Chỉ sử dụng 1 trong 3 cách dưới đây):

1. Cài đặt máy ảo Ubuntu bằng VMWare Player trên Windows/MacOS. Phiên bản đề xuất: Ubuntu 16 trở lên
Hướng dẫn: <https://www.thegioididong.com/game-app/cach-cai-dat-ubuntu-tren-vmware-chi-tiet-day-du-nhat-1423306>
2. Cài đặt Windows Subsystem for Linux (WSL) trên Windows (Được khuyến nghị).
Hướng dẫn: <https://docs.hpcc.vn/pages/viewpage.action?pageId=65660>
3. Cài đặt Linux dual-boot với Windows (Không khuyến nghị).

IV. LÀM QUEN VỚI GIAO DIỆN DÒNG LỆNH – CLI VÀ LINUX SHELL

1. Giao diện dòng lệnh - CLI

a. Giới thiệu về CLI

CLI là viết tắt của **Command Line Interface** (giao diện dòng lệnh). CLI đơn giản là một chương trình cho phép người dùng gõ các lệnh dạng văn bản để hướng dẫn máy tính thực hiện các tác vụ cụ thể. Ví dụ lệnh `cal` dưới đây ra lệnh cho máy tính hiển thị lịch của tháng hiện tại kèm bôi đậm ngày hôm nay:

```
~$ cal
    September 2022
Su Mo Tu We Th Fr Sa
                1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30
```



TRƯỜNG ĐẠI HỌC BÁCH KHOA – ĐẠI HỌC QUỐC GIA TP.HCM TRUNG TÂM KỸ THUẬT ĐIỆN TOÁN

268 Lý Thường Kiệt, Phường 14, Quận 10, TP.Hồ Chí Minh
Điện thoại: 84-8-3864 7256 ext 5371 – Fax: 84-8-3865 8687
Website: www.cce.hcmut.edu.vn
E-mail: dientoan@hcmut.edu.vn

Trong những thế hệ máy tính đầu tiên, CLI được sử dụng như là phương pháp giao tiếp cơ bản giữa con người và máy móc, với một bàn phím là thiết bị đầu vào và màn hình máy tính chỉ có thể hiển thị thông tin dạng văn bản thuần túy. Sau khi nhập lệnh (lệnh *cal* phía trên), kết quả người dùng nhận được sẽ là thông tin dạng văn bản (trả về lịch tháng hiện tại) hoặc hành động cụ thể được máy tính thực hiện như xóa, sửa tập tin, tắt máy,...

Khi thao tác với CLI, việc gõ đúng dòng lệnh chính là điểm mấu chốt. Nếu người dùng gõ sai lệnh, rất có thể họ sẽ xóa nhầm tập tin hoặc vô tình đóng chương trình trước khi lưu, hoặc nếu may mắn hơn là dòng lệnh không được thực thi. Tuy nhiên, việc phát minh ra chuột máy tính đã đánh dấu sự bắt đầu của phương pháp trở và nhấp chuột, như là cách mới để tương tác với máy tính thông qua giao diện đồ họa người dùng (Graphic User Interface – GUI). Ngày nay, GUI đã trở thành một cách lập trình phổ biến. Tuy nhiên, hầu hết các hệ điều hành vẫn cung cấp lập trình kết hợp giữa CLI và GUI.

Mặc dù CLI khó hiểu hơn và không trực quan hơn GUI nhưng CLI vẫn được các lập trình viên ưa chuộng do các lý do sau:

- **Không phải phần mềm nào cũng dùng được GUI:** So với GUI, việc phát triển ứng dụng bằng CLI sẽ tiết kiệm được nhiều thời gian và công sức thực hiện hơn.
- **Hầu hết các tool cho lập trình viên đều có bản chất là CLI:** Trên thực tế, các tool dành cho developer đa phần đều là CLI. Ví dụ như là npm, yarn, composer, docker, git,... đều được ưu tiên sử dụng trên CLI hơn là trên GUI. Nguyên nhân chủ yếu là do các developer thì đều yêu thích sự đơn giản, nhanh chóng và linh hoạt. Để đáp ứng được các yêu cầu này thì CLI quả thực là sự lựa chọn tốt hơn GUI.
- **CLI hỗ trợ đa dạng hệ điều hành:** Bất kỳ hệ điều hành nào cũng có thể truy cập thông qua CLI, trong khi một số hệ điều hành như Ubuntu Server lại không có giao diện đồ họa.
- **Tính đơn giản của CLI:** Trái với sự rườm rà và khó tiếp cận bởi nhiều bước thực hiện của GUI thì CLI chỉ cần một lệnh là đã có thể nhanh chóng xem thông tin và tương tác với hệ thống. Việc sử dụng CLI cũng hầu như không tiêu tốn tài nguyên so với giao diện đồ họa.
- **CLI rất mạnh mẽ:** Với CLI, bạn sẽ có toàn quyền kiểm soát hệ thống của mình và có thể thực hiện được rất nhiều thao tác mà bạn không thể làm được trên GUI.

b. Khởi chạy Linux Terminal

Để sử dụng Linux CLI, chúng ta sẽ sử dụng **Terminal**. Terminal là một text-based interface (giao diện dòng lệnh) cho phép người dùng tương tác trực tiếp vào hệ thống UNIX. Terminal sẽ tiếp nhận các dòng lệnh do chúng ta nhập vào qua bàn phím và thực thi chúng, sau đó trả kết quả trên màn hình.

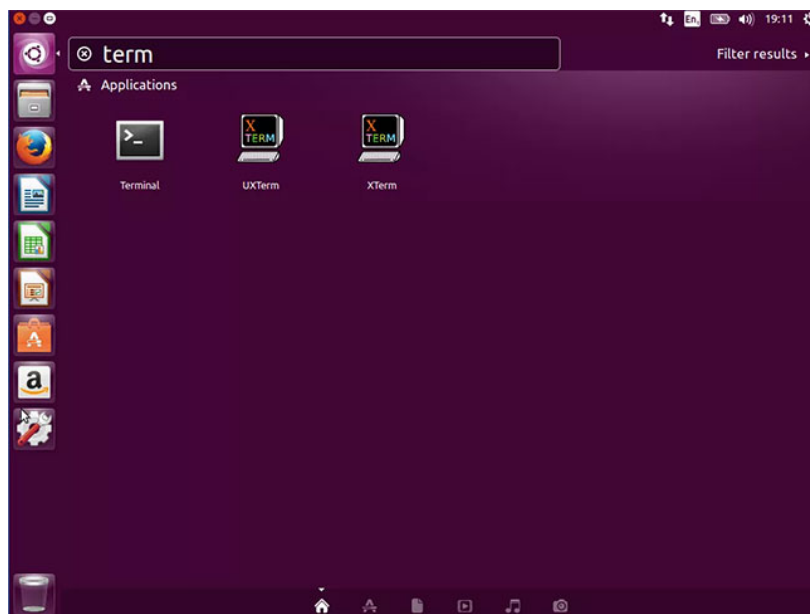
Để khởi chạy Terminal trên Ubuntu (Máy ảo/Máy vật lý), chúng ta có hai cách sau:

- **Cách 1 (Qua GUI):** Nhấn phím cửa sổ, gõ *Terminal*. Sau đó, biểu tượng Terminal hiện ra. Bạn click chuột vào để mở.

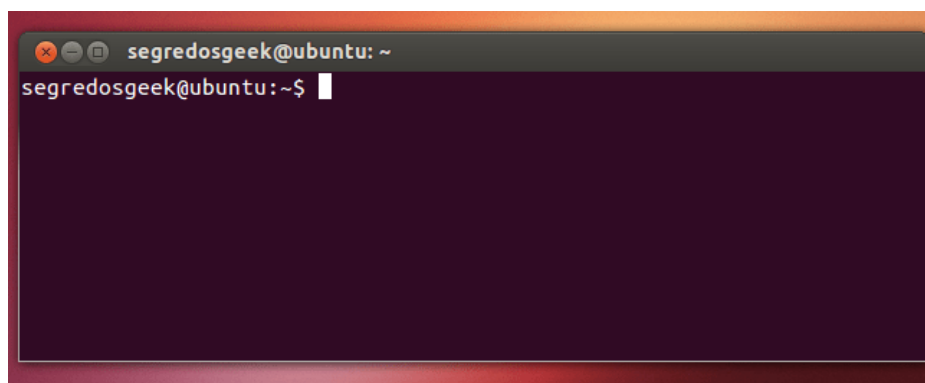


TRƯỜNG ĐẠI HỌC BÁCH KHOA – ĐẠI HỌC QUỐC GIA TP.HCM
TRUNG TÂM KỸ THUẬT ĐIỆN TOÁN

268 Lý Thường Kiệt, Phường 14, Quận 10, TP.Hồ Chí Minh
Điện thoại: 84-8-3864 7256 ext 5371 – Fax: 84-8-3865 8687
Website: www.cce.hcmut.edu.vn
E-mail: dientoan@hcmut.edu.vn



- Dùng tổ hợp phím tắt Ctrl Alt T để mở Terminal

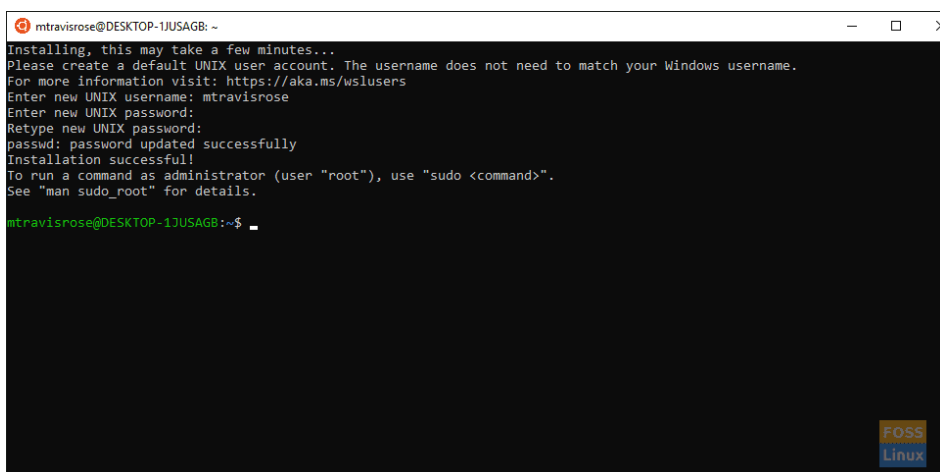
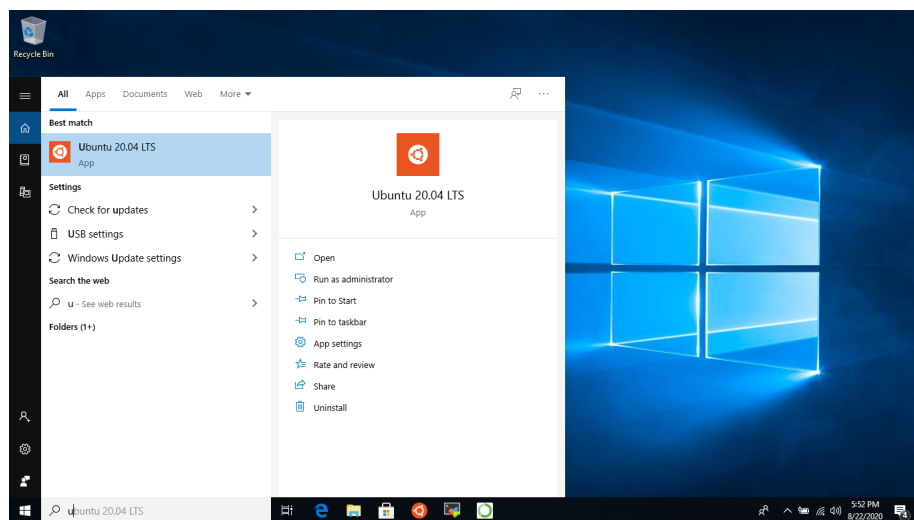


Đối với WSL, sau khi đã cài đặt thành công Ubuntu từ cửa hàng Windows Store theo hướng dẫn, chúng ta có thể truy cập giao diện dòng lệnh bằng cách khởi chạy ứng dụng Ubuntu trên máy (thông qua tìm kiếm/ Start Menu):



TRƯỜNG ĐẠI HỌC BÁCH KHOA – ĐẠI HỌC QUỐC GIA TP.HCM TRUNG TÂM KỸ THUẬT ĐIỆN TOÁN

268 Lý Thường Kiệt, Phường 14, Quận 10, TP.Hồ Chí Minh
Điện thoại: 84-8-3864 7256 ext 5371 – Fax: 84-8-3865 8687
Website: www.cce.hcmut.edu.vn
E-mail: dientoan@hcmut.edu.vn



2. Giới thiệu về Linux Shell

Như vậy chúng ta đã thử nghiệm một số câu lệnh cơ bản trên Linux CLI. Vậy bản chất các câu lệnh trên được thực thi như thế nào? Đây chính là lúc chúng ta nói về Linux Shell.

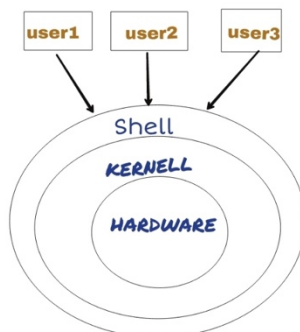
Shell là chương trình giữa bạn và Linux (hay nói chính xác hơn là giữa bạn với nhân Linux). Mỗi lệnh bạn gõ ra sẽ được Shell diễn dịch rồi chuyển tới nhân Linux. Nói một cách dễ hiểu Shell là bộ diễn dịch ngôn ngữ lệnh, ngoài ra nó còn tận dụng triệt để các trình tiện ích và chương trình ứng dụng có trên hệ thống...

Bash (Bourne-Again SHell) là shell được sử dụng gần như mặc định trên các bản phân phối của Linux.



TRƯỜNG ĐẠI HỌC BÁCH KHOA – ĐẠI HỌC QUỐC GIA TP.HCM
TRUNG TÂM KỸ THUẬT ĐIỆN TOÁN

268 Lý Thường Kiệt, Phường 14, Quận 10, TP.Hồ Chí Minh
Điện thoại: 84-8-3864 7256 ext 5371 – Fax: 84-8-3865 8687
Website: www.cce.hcmut.edu.vn
E-mail: dientoan@hcmut.edu.vn



linux architecture

3. Cấu trúc dòng lệnh Linux

Sau khi khởi chạy thành công Terminal, chúng ta sẽ bắt đầu tìm hiểu giao diện và cấu trúc các câu lệnh cơ bản trên Linux

a. Giao diện tương tác dòng lệnh trên Linux

Sau khi chúng ta mở thành công Terminal lên, giao diện dòng lệnh trên Ubuntu sẽ chờ chúng ta nhập lệnh vào:

```
Student01@lab-computer:~$
```

Các kí tự cố định bên trái của mỗi dòng lệnh được gọi là *shell prompt*, được hiển thị theo cấu trúc sau:

`<username>@<hostname>:<location>$`

Trong đó:

- *username*: Tên người dùng hiện tại (*student01*)
- *hostname*: Tên của máy tính (*lab-computer*)
- *location*: Thư mục đang thao tác (*Thư mục HOME (~) của student01*)
- *\$*: Dấu phân cách kết thúc shell prompt

Khi gõ lệnh vào từ bàn phím thì kí tự nhập sẽ được hiển thị sau dấu \$ và ấn phím Enter để thực thi lệnh. Ví dụ dưới đây chúng ta sẽ gõ lệnh `whoami` để hiển thị tên người dùng hiện tại đang đăng nhập vào Linux:

```
student01@lab-computer:~$ whoami  
student01
```

Chúng ta cũng có thể dùng phím mũi tên lên để quay lại các câu lệnh đã được thực thi trước đó.

b. Cấu trúc câu lệnh



TRƯỜNG ĐẠI HỌC BÁCH KHOA – ĐẠI HỌC QUỐC GIA TP.HCM
TRUNG TÂM KỸ THUẬT ĐIỆN TOÁN

268 Lý Thường Kiệt, Phường 14, Quận 10, TP.Hồ Chí Minh
Điện thoại: 84-8-3864 7256 ext 5371 – Fax: 84-8-3865 8687
Website: www.cce.hcmut.edu.vn
E-mail: dientoan@hcmut.edu.vn

Các câu lệnh Linux có thể tự chạy hoặc chấp nhận các tùy chọn và thông số để thay đổi hành vi của chúng. Cú pháp điển hình của các câu lệnh có thể tóm gọn như sau:

câu lệnh [tùy chọn] [thông số]

Trong đó các tùy chọn thường được định nghĩa bằng hai cách như sau:

- *Dạng rút gọn*: Bắt đầu bằng 1 dấu gạch ngang (-) và theo sau là một kí tự. Ví dụ: -l
- *Dạng đầy đủ*: Bắt đầu bằng 2 dấu gạch ngang (--) và theo sau là một từ. Ví dụ --list

Thông thường mỗi tùy chọn sẽ cung cấp cả 2 dạng. Ví dụ: -a và --all đều diễn giải cho tùy chọn hiển thị tất cả.

Trong khi đó, thông số đầu vào có thể là số (1), kí tự (“Hello World”) hoặc đường dẫn tới một tập tin (/home/user/abc.txt).

Có thể có nhiều tùy chọn và thông số khác nhau được cung cấp khi chạy các câu lệnh, hoặc chỉ cung cấp mỗi tùy chọn hoặc thông số, hoặc thậm chí không cung cấp bất kì tùy chọn hay thông số gì.

Cùng xét ví dụ dưới đây khi ta chạy lệnh `ls` để liệt kê các tập tin và thư mục con của một thư mục:

```
student01@lab-computer:~$ ls
```

Khi chúng ta không cung cấp bất kì tùy chọn hay thông số nào thì câu lệnh sẽ sử dụng thiết lập mặc định như ở đây là hiển thị danh sách đơn giản các tập tin và thư mục tại thư mục đang làm việc hiện tại (~).

Để liệt kê chi tiết hơn thông tin, ta sử dụng tùy chọn `-l` và có thể thêm thông số thư mục muốn kiểm tra như `/etc`

```
student01@lab-computer:~$ ls -l /etc
total 1376
drwxr-xr-x  2 root root      4096 Jun 29  2017 ImageMagick-6
drwxr-xr-x  8 root root      4096 May 18  2017 NetworkManager
drwxr-xr-x  2 root root      4096 Nov 13  2016 Upower
...
```

So sánh kết quả khi bỏ đi tùy chọn `-l`:

```
student01@lab-computer:~$ ls /etc
ImageMagick-6      firefox          logrotate.conf    rc5.d
NetworkManager     fonts           logrotate.d       rc6.d
...
```

Hoặc kết hợp nhiều tùy chọn khác nhau như thêm tùy chọn `-a` để hiển thị tất cả các tập tin ẩn:

```
student01@lab-computer:~$ ls -l -a /etc
```



TRƯỜNG ĐẠI HỌC BÁCH KHOA – ĐẠI HỌC QUỐC GIA TP.HCM
TRUNG TÂM KỸ THUẬT ĐIỆN TOÁN

268 Lý Thường Kiệt, Phường 14, Quận 10, TP.Hồ Chí Minh
Điện thoại: 84-8-3864 7256 ext 5371 – Fax: 84-8-3865 8687
Website: www.cce.hcmut.edu.vn
E-mail: dientoan@hcmut.edu.vn

```
total 1396
drwxr-xr-x 154 root root      12288 Jun 21 10:34 .
drwxr-xr-x  24 root root       4096 Jun 29  2017 ..
drwxr-xr-x   3 root root       4096 Nov 18  2016 .java
-rw-----   1 root root         0 Jul 23  2014 .pwd.lock
drwxr-xr-x   2 root root       4096 Jun 29  2017 ImageMagick-6
drwxr-xr-x   8 root root       4096 May 18  2017 NetworkManager
...
```

Hoặc gộp hai tùy chọn lại:

```
student01@lab-computer:~$ ls -la /etc
total 1396
drwxr-xr-x 154 root root      12288 Jun 21 10:34 .
...
```

c. Chi tiết câu lệnh

Trong trường hợp muốn biết thông tin chi tiết thông tin về một lệnh, ta sử dụng cú pháp:

man <lệnh cần tra cứu>

“Man” là từ viết tắt của “manual”, được coi là tài liệu trực tuyến trong Linux đã lưu trữ toàn bộ các lệnh có sẵn với các thông tin tham khảo khá đầy đủ cho phép người dùng có thể mở ra để nhận được trợ giúp. Ví dụ tra cứu đầy đủ thông tin về lệnh *ls*:

```
student01@lab-computer:~$ man ls
NAME
    ls - list directory contents
...
(Ấn q để thoát)
```

Ngoài ra, chúng ta cũng có thể sử dụng tùy chọn *--help* hoặc *-h* để xem trợ giúp do chính câu lệnh cung cấp (với điều kiện chương trình đó hỗ trợ):

```
student01@lab-computer:~$ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILES (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.
  -a, --all                do not ignore entries starting with .
...
```

4. Các câu lệnh với tập tin và thư mục

Trong Linux, tất cả mọi thứ đều được coi là tập tin, kể cả thư mục cũng là một loại tập tin có các đặc điểm riêng. Linux có cấu trúc thư mục không phân chia theo ổ đĩa(C, D, ...) như Windows mà có một thư mục gốc gọi là thư mục root (thường viết là /). Từ / sẽ phân chia thành nhiều loại thư mục có mục đích dùng khác nhau:

Một số thư mục hay dùng đến là:

- /home: chứa nội dung của người dùng user



TRƯỜNG ĐẠI HỌC BÁCH KHOA – ĐẠI HỌC QUỐC GIA TP.HCM
TRUNG TÂM KỸ THUẬT ĐIỆN TOÁN

268 Lý Thường Kiệt, Phường 14, Quận 10, TP.Hồ Chí Minh
Điện thoại: 84-8-3864 7256 ext 5371 – Fax: 84-8-3865 8687
Website: www.cce.hcmut.edu.vn
E-mail: dientoan@hcmut.edu.vn

- /root: chứa nội dung của người dùng root
- /bin, /usr/bin: chứa các chương trình thực thi, hầu hết các câu lệnh hệ thống được chạy. Ví dụ `ls` như chúng ta đã thực thi ở các ví dụ trên
- /sbin, /usr/sbin: chứa các chương trình thực thi giành cho admin
- .. : Thư mục cha của thư mục đang làm việc
- . : Thư mục đang làm việc

Đường dẫn thể hiện cách chúng ta có thể tham chiếu với cấu trúc thư mục. Dấu / được dùng trong đường dẫn để phân cách từng bậc của cấu trúc này. Có 2 loại đường dẫn chúng ta sẽ gặp là:

- **Đường dẫn tuyệt đối (absolute path):** đường dẫn này thể hiện vị trí của tệp liên quan đến thư mục root, do đó luôn bắt đầu với /. Ví dụ: /usr/bin
- **Đường dẫn tương đối (relative path) :** đường dẫn này thể hiện vị trí của tệp từ thư mục hiện tại, do đó sẽ bắt đầu từ thư mục hiện tại . Ví dụ: ../usr/bin hay ./test

Vậy là có thể dễ dàng xác định loại đường dẫn bằng cách quan sát, nếu đường dẫn bắt đầu bằng / thì là tuyệt đối vì bắt đầu ở root, còn đường dẫn có chứa . hay .. là tương đối vì sử dụng thư mục hiện tại. Và thêm nữa, đường dẫn có thể giúp chúng ta đến thư mục bin có thể là /usr/bin hoặc ../user/bin.

a. Hiển thị thư mục đang làm việc

Chúng ta sử dụng lệnh `pwd` (print working directory) để hiển thị đường dẫn tuyệt đối của thư mục hiện tại (tính từ /):

```
student01@lab-computer:~$ pwd
/home/student01
```

b. Thay đổi thư mục làm việc

Để thay đổi thư mục làm việc, chúng ta sử dụng lệnh:
`cd <đường dẫn thư mục muốn đến>`

```
student01@lab-computer:~$ cd /home
student01@lab-computer:/home$
```

Để di chuyển về thư mục cha hoặc con, chúng ta có thể sử dụng đường dẫn tương đối:

```
student01@lab-computer:/etc/bin$ cd ..
student01@lab-computer:/etc$
```

Đặc biệt: Hãy thử `cd ~` hoặc `cd` (không có đường dẫn)

c. Liệt kê nội dung của thư mục

Như đã giới thiệu trong phần trước, chúng ta sử dụng lệnh `ls` để liệt kê nội dung tại một thư mục:

- Mặc định lệnh `ls` sẽ hiển thị tất cả các file trừ những file ẩn
- File ẩn là những file môi trường có bắt đầu bằng . (ví dụ: ., .., .env, .gitignore)
- Sử dụng tùy chọn `-a` để liệt kê tất cả các file (bao gồm các file bị ẩn)



TRƯỜNG ĐẠI HỌC BÁCH KHOA – ĐẠI HỌC QUỐC GIA TP.HCM
TRUNG TÂM KỸ THUẬT ĐIỆN TOÁN

268 Lý Thường Kiệt, Phường 14, Quận 10, TP.Hồ Chí Minh
Điện thoại: 84-8-3864 7256 ext 5371 – Fax: 84-8-3865 8687
Website: www.cce.hcmut.edu.vn
E-mail: dientoan@hcmut.edu.vn

- Sử dụng tùy chọn `-l` để liệt kê ở long format
- Sử dụng `-la` để kết hợp cả hai tùy chọn trên lại, hiển thị tất cả các file ở long format

```
[thanhnguyen@Chicken-house] - [~/Desktop/example-linux] - [2022-02-01 05:20:53]
[0] ls
ls-example test.txt text.txt
[thanhnguyen@Chicken-house] - [~/Desktop/example-linux] - [2022-02-01 05:20:56]
[0] ls -a
.      ..      .env      .gitignore ls-example test.txt text.txt
[thanhnguyen@Chicken-house] - [~/Desktop/example-linux] - [2022-02-01 05:21:06]
[0] ls -l
total 8
drwxr-xr-x  3 thanhnguyen  staff  96 Feb  1 17:20 ls-example
-rw-r--r--  1 thanhnguyen  staff   0 Feb  1 16:26 test.txt
-rw-r--r--@ 1 thanhnguyen  staff   4 Dec 15 16:34 text.txt
[thanhnguyen@Chicken-house] - [~/Desktop/example-linux] - [2022-02-01 05:21:12]
[0] ls -la
total 8
drwxr-xr-x  7 thanhnguyen  staff  224 Feb  1 17:20 .
drwx-----@ 41 thanhnguyen  staff 1312 Feb  1 16:55 ..
-rw-r--r--  1 thanhnguyen  staff   0 Feb  1 17:20 .env
-rw-r--r--  1 thanhnguyen  staff   0 Feb  1 17:19 .gitignore
drwxr-xr-x  3 thanhnguyen  staff  96 Feb  1 17:20 ls-example
-rw-r--r--  1 thanhnguyen  staff   0 Feb  1 16:26 test.txt
```

filetype
d is folder
- is file

admin group user
permission

hard link count

user owner group owner file size modification time file name

d. Tạo thư mục mới

`mkdir <tên thư mục mới>`

```
student01@lab-computer:~$ mkdir lab
student01@lab-computer:~$ ls
lab
```

Xoá thư mục hoặc tập tin

`rm <tên thư mục/tập tin cần xoá>`

Có thể kèm tùy chọn:

- `-f` để xoá ngay lập tức mà không cần xác nhận
- `-r` để xoá toàn bộ nội dung trong thư mục.

```
student01@lab-computer:~$ rm -rf lab
student01@lab-computer:~$ ls
```

e. Tạo tập tin trống

`touch <tên tập tin mới>`

```
student01@lab-computer:~$ touch ex.txt
student01@lab-computer:~$ ls
ex.txt
```



TRƯỜNG ĐẠI HỌC BÁCH KHOA – ĐẠI HỌC QUỐC GIA TP.HCM
TRUNG TÂM KỸ THUẬT ĐIỆN TOÁN

268 Lý Thường Kiệt, Phường 14, Quận 10, TP.Hồ Chí Minh
Điện thoại: 84-8-3864 7256 ext 5371 – Fax: 84-8-3865 8687
Website: www.cce.hcmut.edu.vn
E-mail: dientoan@hcmut.edu.vn

f. Sao chép tập tin:

cp <file-cần-copy> <nơi-copy-đến>

```
student01@lab-computer:~$ cp ex.txt /home/student02/ex.txt
student01@lab-computer:~$ ls /home/student02
ex.txt
```

Có thể sử dụng thêm option *-i* để vào chế độ cho phép tương tác(interactive) tức là được lựa chọn có ghi đè file cùng tên hay không.

g. Di chuyển tập tin

mv <file-cần-copy> <nơi-copy-đến>

```
student01@lab-computer:~$ mv ex.txt ex2.txt
student01@lab-computer:~$ ls
ex2.txt
```

Nếu nơi đến trùng với nơi ban đầu của tập tin, thì thao tác di chuyển trở thành thao tác đổi tên tập tin

h. Sửa nội dung tập tin

nano <Tên tập tin>

```
student01@lab-computer:~$ nano ex2.txt
```

```
GNU nano 2.5.3 File: ex2.txt

[ Read 0 lines ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text   ^J Justify    ^C Cur Pos   ^Y Prev Page  M-^ First Line
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell   ^_ Go To Line  ^V Next Page  M-^ Last Line
```

GNU nano là một trình soạn thảo văn bản dòng lệnh để sử dụng cho các hệ điều hành Unix và Linux. Sau khi trình soạn thảo nano hiện ra, chúng ta có thể gõ trực tiếp vào CLI và dùng các phím điều hướng để di chuyển.

- Để tìm kiếm văn bản, nhấn Ctrl + W, nhập cụm từ tìm kiếm và nhấn Enter. Con trỏ sẽ di chuyển đến từ mà chúng ta tìm kiếm. Để di đến vị trí tiếp theo, nhấn Alt + W.
- Để lưu các thay đổi chúng ta đã thực hiện vào tệp, nhấn Ctrl + O. Nếu tập tin không tồn tại, nó sẽ được tạo khi chúng ta lưu nó.
- Để thoát nano, nhấn Ctrl + X. Nếu có những thay đổi chưa được lưu, nano sẽ được hỏi liệu chúng ta có muốn lưu các thay đổi đó không (nhập y và ấn Enter).

i. Xem nội dung file

Lệnh *cat* giúp xem nội dung của một file. Nếu muốn hiển thị số dòng, chúng ta có thể sử dụng thêm lựa chọn *-n <số dòng>*



TRƯỜNG ĐẠI HỌC BÁCH KHOA – ĐẠI HỌC QUỐC GIA TP.HCM
TRUNG TÂM KỸ THUẬT ĐIỆN TOÁN

268 Lý Thường Kiệt, Phường 14, Quận 10, TP.Hồ Chí Minh
Điện thoại: 84-8-3864 7256 ext 5371 – Fax: 84-8-3865 8687
Website: www.cce.hcmut.edu.vn
E-mail: dientoan@hcmut.edu.vn

Ngoài ra chúng ta có thể kết hợp với *more* hoặc *less* để xem các tập tin lớn bằng cách chuyển hướng sẽ được giới thiệu dưới đây.

j. Một số lệnh khác:

- *locate*: tìm kiếm file trong một database có sẵn.
- *wc*: hiển thị các thông tin thống kê của nội dung file, như số dòng, số từ, số ký tự
- *clear*: xoá nội dung đang hiển thị trên Terminal
- *history*: xem lại lịch sử các lệnh đã nhập
- *diff*: so sánh và hiển thị sự khác nhau giữa hai tập tin
- *df*: nhận báo cáo về việc sử dụng dung lượng ổ đĩa của hệ thống
- *du*: kiểm tra xem một tập hoặc một thư mục chiếm bao nhiêu dung lượng

5. Chuyển hướng câu lệnh

a. Luồng dữ liệu

Quy trình làm việc cơ bản của bất cứ câu lệnh nào chính là nó sẽ nhận **đầu vào**(input) và trả về một **đầu ra**(output).

Một câu lệnh sẽ có 3 luồng dữ liệu gồm:

- **standard input(stdin)** : Là dữ liệu được truyền vào câu lệnh. Stdin thường là từ bàn phím, ngoài ra còn có thể từ file hoặc một process khác
- **standard out(stdout)**: Là kết quả được trả về sau khi thực thi câu lệnh thành công.
- **standard error(stderr)**: Là lỗi được trả về sau khi thực thi câu lệnh và có lỗi gì đó xảy ra. Stdout thường xuất ra trên màn hình, ngoài ra còn có thể xuất ra file hoặc một process khác

Ví dụ: khi bạn gõ *ls* thì đây chính là stdin, còn stdout là kết quả bạn thấy trên màn hình

```
..example-linux (-zsh) 261 +
~/Desktop/example-linux >
~/Desktop/example-linux > ls
conf          config-3-chars manage-dir
config        config-begin-ab test.txt
~/Desktop/example-linux >
```

b. Chuyển hướng đầu ra câu lệnh vào tập tin

“**Chuyển hướng**” là một tính năng trong Linux cho phép chúng ta thay đổi đầu vào tiêu chuẩn(stdin) và đầu ra tiêu chuẩn(stdout) khi thực hiện một câu lệnh. Sử dụng *>* hoặc *>>* để có thể lưu kết quả in ra của một câu lệnh vào một tập tin. Ví dụ:



TRƯỜNG ĐẠI HỌC BÁCH KHOA – ĐẠI HỌC QUỐC GIA TP.HCM
TRUNG TÂM KỸ THUẬT ĐIỆN TOÁN

268 Lý Thường Kiệt, Phường 14, Quận 10, TP.Hồ Chí Minh
Điện thoại: 84-8-3864 7256 ext 5371 – Fax: 84-8-3865 8687
Website: www.cce.hcmut.edu.vn
E-mail: dientoan@hcmut.edu.vn

```
student01@lab-computer:~$ ls /etc > list.txt
student01@lab-computer:~$ cat list.txt
ImageMagick-6
NetworkManager
UPower
X11
...
```

Lưu ý các tùy chọn ở đây:

- > : Ghi đè toàn bộ nội dung tập tin
- >> : Chèn nối tiếp vào nội dung đang có trên tập tin
- 2> : Chỉ lưu đầu ra vào tập tin nếu câu lệnh lỗi

c. Chuyển hướng đầu vào câu lệnh

Dùng dấu < để chuyển hướng cho đầu vào(stdin)

Ví dụ dưới đây chuyển đầu vào cho câu lệnh chuyển đổi chữ thường sang chữ hoa với kết quả của lệnh **cal** trong file mycal. Tức là chúng ta sẽ muốn in ra màn hình lịch ở định dạng in hoa:

```
student01@lab-computer:~$ cal > mycal
student01@lab-computer:~$ cat mycal
September 2022
Su Mo Tu We Th Fr Sa
          1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 2

student01@lab-computer:~$ tr 'a-z' 'A-Z' < mycal
SEPTEMBER 2022
SU MO TU WE TH FR SA
          1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30
```

d. Chuyển hướng đầu ra của câu lệnh này thành đầu vào của câu lệnh khác

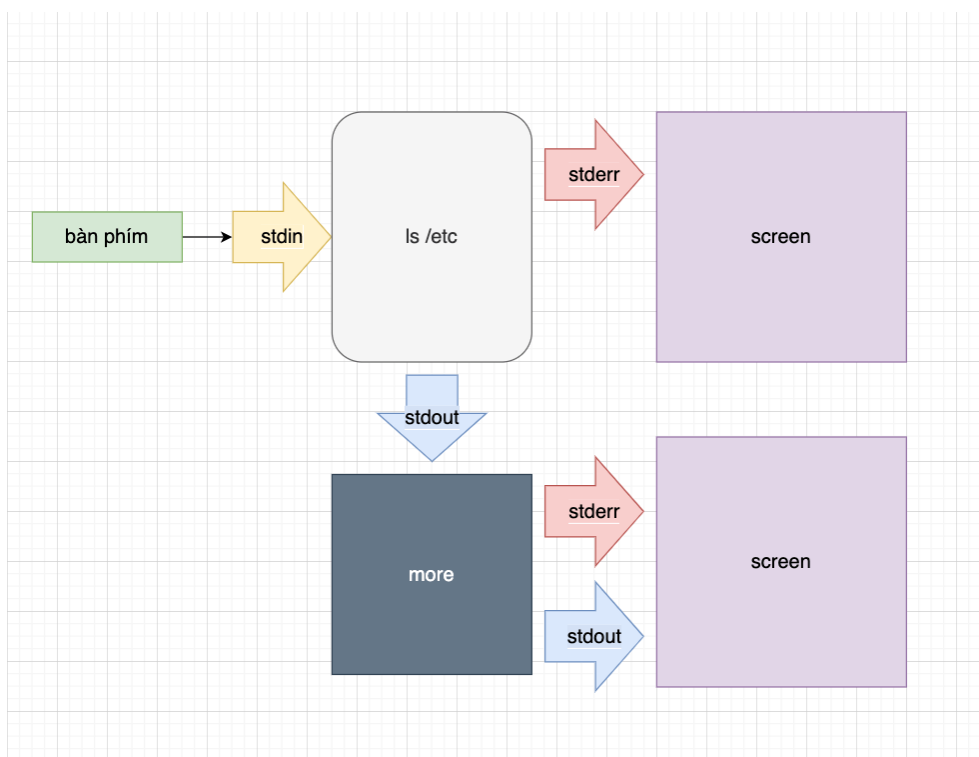
Chúng ta dùng dấu | để chuyển hướng (Piping) cho đầu ra vào câu lệnh khác. Chúng ta cũng có thể chuyển hướng đầu ra của câu lệnh này vào câu lệnh khác thay vì chuyển vào file như trên. Việc này rất hữu ích khi mà có nhiều câu lệnh cần nối chuỗi với nhau hay có kết quả quá nhiều.

Ví dụ: `ls /etc | more`



TRƯỜNG ĐẠI HỌC BÁCH KHOA – ĐẠI HỌC QUỐC GIA TP.HCM
TRUNG TÂM KỸ THUẬT ĐIỆN TOÁN

268 Lý Thường Kiệt, Phường 14, Quận 10, TP.Hồ Chí Minh
Điện thoại: 84-8-3864 7256 ext 5371 – Fax: 84-8-3865 8687
Website: www.cce.hcmut.edu.vn
E-mail: dientoan@hcmut.edu.vn



Khi chúng ta gõ “*ls /etc | more*” từ bàn phím, kết quả lệnh *ls* nếu có lỗi sẽ được in ra màn hình, nếu không có lỗi sẽ chuyển đầu ra thành đầu vào của lệnh *more* và xuất kết quả ra màn hình.

e. Lọc kết quả đầu ra bằng lệnh *grep*

Grep là từ viết tắt của Global Regular Expression Print. Lệnh *grep* trong Linux được sử dụng để tìm kiếm một chuỗi ký tự trong một file được chỉ định, rất tiện lợi khi tìm kiếm các file log lớn. Ví dụ lọc các tập tin trong thư mục */etc* có chứa ký tự “*sys*” :

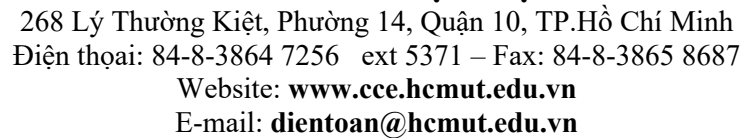
```
student01@lab-computer:~$ ls -a /etc | grep sys
rsyslog.conf
rsyslog.d
sysctl.conf
...
```

6. Sudo

Sudo là từ viết tắt của “ substitute user do “, hay “ super user do “. Đây là một chương trình của Linux. Nó được cho phép user chạy chương trình với những đặc quyền bảo mật thông tin của user khác trong Linux (thường là root user).

Đối với một số chương trình cần đặc quyền hay thư mục/tập tin được bảo vệ, chúng ta phải sử dụng *sudo* để có thể thực thi chương trình hoặc các hành động trên thư mục/tập tin đó. Ví dụ:

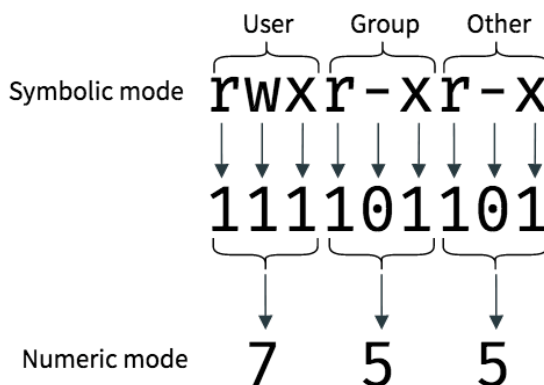
```
student01@lab-computer:~$ mkdir /opt/sample
mkdir: cannot create directory '/opt/sample': Permission denied
student01@lab-computer:~$ sudo mkdir /opt/sample
```





TRƯỜNG ĐẠI HỌC BÁCH KHOA – ĐẠI HỌC QUỐC GIA TP.HCM
TRUNG TÂM KỸ THUẬT ĐIỆN TOÁN

268 Lý Thường Kiệt, Phường 14, Quận 10, TP.Hồ Chí Minh
Điện thoại: 84-8-3864 7256 ext 5371 – Fax: 84-8-3865 8687
Website: www.cce.hcmut.edu.vn
E-mail: dientoan@hcmut.edu.vn



Để thay đổi phân quyền cho file và thư mục ta sẽ sử dụng lệnh *chmod*:
chmod <tùy chọn> <chỉ số phân quyền> <tên tập tin/thư mục>

Ví dụ dưới đây cấp quyền thực thi cho bash script *a.sh*:

```
student01@lab-computer:~$ echo 'echo "Hello World"' > a.sh
student01@lab-computer:~$ ls
a.sh
student01@lab-computer:~$ ./a.sh
-bash: ./a.sh: Permission denied
student01@lab-computer:~$ chmod 744 a.sh
student01@lab-computer:~$ ./a.sh
Hello World
```

Ngoài ra chúng ta còn có thể thay đổi các quyền sở hữu của một file. Để thay đổi được cần phải có quyền sudo:

sudo chown <username> <filename>

V. LẬP TRÌNH BASH SCRIPT

1. Giới thiệu về Bash Script

Bash Script là một chuỗi các lệnh được viết trong một tệp. Chúng được đọc và thực thi bởi chương trình bash. Chương trình thực hiện từng dòng một. Ví dụ: chúng ta có thể điều hướng đến một đường dẫn nhất định, tạo một thư mục và tạo ra một quy trình bên trong nó bằng dòng lệnh.

Chúng ta có thể thực hiện trình tự các bước tương tự bằng cách lưu các lệnh trong một tệp lệnh bash và chạy bất kỳ lúc nào chúng ta muốn, thay vì phải gõ lại các lệnh bằng tay.

Theo quy ước đặt tên, các tệp lệnh bash kết thúc bằng *.sh*. Tuy nhiên, các tệp lệnh bash có thể chạy hoàn toàn tốt mà không cần phần mở rộng sh.

Một Bash Script cũng được xác định với một *shebang*. Shebang là dòng đầu tiên của mỗi script. Shebang ra lệnh cho shell thực thi nó thông qua bash shell hoặc các shell khác mà chúng ta mong muốn như zsh. Shebang chỉ đơn giản là một đường dẫn tuyệt đối đến trình thông dịch bash, như ví dụ dưới đây:



TRƯỜNG ĐẠI HỌC BÁCH KHOA – ĐẠI HỌC QUỐC GIA TP.HCM
TRUNG TÂM KỸ THUẬT ĐIỆN TOÁN

268 Lý Thường Kiệt, Phường 14, Quận 10, TP.Hồ Chí Minh
Điện thoại: 84-8-3864 7256 ext 5371 – Fax: 84-8-3865 8687
Website: www.cce.hcmut.edu.vn
E-mail: dientoan@hcmut.edu.vn

```
#!/bin/bash
```

Lưu ý là người dùng cần có quyền thực thi (execute) để có thể chạy một bash script.

2. Cách tạo một Bash Script

Giờ chúng ta sẽ tạo một script đơn giản trong bash để xuất ra dòng chữ Hello World. Đầu tiên, chúng ta tạo một tệp tin có tên hello_world.sh:

```
student01@lab-computer:~$ touch hello_world.sh
```

Sau đó đường dẫn đến trình thực thi bash của hệ thống để đưa vào shebang:

```
student01@lab-computer:~$ which bash  
/bin/bash
```

Sử dụng nano để sửa tệp tin hello_world.sh mà chúng ta vừa tạo phía trên với nội dung như sau:

```
#!/bin/bash  
echo "Hello World"
```

Sau đó thêm quyền thực thi của script:

```
student01@lab-computer:~$ chmod +x hello_world.sh
```

Cuối cùng chúng ta có thể chạy script trên bằng 1 trong 2 cách:

```
student01@lab-computer:~$ ./hello_world.sh  
Hello World  
student01@lab-computer:~$ bash hello_world.sh  
Hello World
```

3. Cú pháp Bash Script

a. Khai báo biến

Chúng ta có thể xác định một biến bằng cách sử dụng cú pháp *variable_name=value* (Lưu ý không có dấu cách). Để nhận giá trị của biến, hãy thêm \$ vào trước biến.

```
#!/bin/bash  
# A simple variable example - hello.sh  
greeting=Hello  
name=Tux  
echo $greeting $name  
student01@lab-computer:~$ ./hello.sh  
Hello Tux
```

b. Phép toán số học

Bash cũng hỗ trợ các phép tính toán học:

var=\$((expression))



TRƯỜNG ĐẠI HỌC BÁCH KHOA – ĐẠI HỌC QUỐC GIA TP.HCM
TRUNG TÂM KỸ THUẬT ĐIỆN TOÁN

268 Lý Thường Kiệt, Phường 14, Quận 10, TP.Hồ Chí Minh
Điện thoại: 84-8-3864 7256 ext 5371 – Fax: 84-8-3865 8687
Website: www.cce.hcmut.edu.vn
E-mail: dientoan@hcmut.edu.vn

Ví dụ:

```
#!/bin/bash
# file sum.sh

var=$((1+2))

echo $var

student01@lab-computer:~$ ./sum.sh
3
```

c. Truyền đối số vào Bash Script

Khi khởi chạy một Bash Script, chúng ta có thể truyền vào các đối số và sử dụng trong script. Ví dụ: `./sum.sh 1 2`

Danh sách biến tương ứng các đối số và các biến hệ thống khác:

- \$0- Tên của file Bash script.
- \$1 - \$9 - lần lượt là các đối số truyền vào cho file Bash script.
- \$# - Số lượng các arguments chúng ta truyền vào cho file the Bash script.
- @\$ - Tất cả các đối số cung cấp cho file Bash script.
- \$? - Trạng thái của câu lệnh thực hiện gần nhất (0 -> true , 1 -> false)
- \$\$ - ID của script hiện tại .

```
#!/bin/bash
# file name.sh

echo $0
echo Your name is $1

student01@lab-computer:~$ ./name.sh student
./name.sh
Your name is student
```

d. Đọc vào từ bàn phím:

Đôi khi chúng ta sẽ cần thu thập thông tin đầu vào của người dùng và thực hiện các thao tác có liên quan. Trong bash, chúng ta có thể lấy thông tin đầu vào của người dùng bằng lệnh `read`.

read -p "Thông báo tới người dùng" tên_biến_lưu

```
#!/bin/bash
# file input.sh

read -p "Enter a number: " a
read -p "Enter a number: " b

var=$((a+b))

echo $var

student01@lab-computer:~$ ./input.sh
Enter a number: 1
```



TRƯỜNG ĐẠI HỌC BÁCH KHOA – ĐẠI HỌC QUỐC GIA TP.HCM
TRUNG TÂM KỸ THUẬT ĐIỆN TOÁN

268 Lý Thường Kiệt, Phường 14, Quận 10, TP.Hồ Chí Minh
Điện thoại: 84-8-3864 7256 ext 5371 – Fax: 84-8-3865 8687
Website: www.cce.hcmut.edu.vn
E-mail: dientoan@hcmut.edu.vn

Enter a number: 2
3

e. So sánh

So sánh được sử dụng để kiểm tra xem các câu lệnh đang được đánh giá là đúng hay sai. Chúng ta có thể sử dụng các toán tử được hiển thị bên dưới để so sánh hai câu lệnh:

Phép toán	Cú pháp
=	<code>\$a -eq \$b</code>
>=	<code>\$a -ge \$b</code>
>	<code>\$a -gt \$b</code>
<=	<code>\$a -le \$b</code>
<	<code>\$a -lt \$b</code>
!=	<code>\$b -ne \$b</code>

Cấu trúc lệnh so sánh (if):

```
if [ conditions ]  
then  
    commands  
fi
```

Ví dụ so sánh 2 số X, Y nhập vào từ bàn phím:

```
#!/bin/bash  
# file compare.sh  
  
read x  
read y  
  
if [ $x -gt $y ]  
then  
echo X is greater than Y  
elif [ $x -lt $y ]  
then  
echo X is less than Y  
elif [ $x -eq $y ]  
then  
echo X is equal to Y  
fi  
student01@lab-computer:~$ ./ compare.sh  
3  
5  
X is less than Y
```



TRƯỜNG ĐẠI HỌC BÁCH KHOA – ĐẠI HỌC QUỐC GIA TP.HCM
TRUNG TÂM KỸ THUẬT ĐIỆN TOÁN

268 Lý Thường Kiệt, Phường 14, Quận 10, TP.Hồ Chí Minh
Điện thoại: 84-8-3864 7256 ext 5371 – Fax: 84-8-3865 8687
Website: www.cce.hcmut.edu.vn
E-mail: dientoan@hcmut.edu.vn

Nếu cấu trúc so sánh phức tạp hơn thì chúng ta có thể sử dụng các dạng điều kiện sau:

- *if...then...else...fi*
- *if..elif..else..fi*
- *if..then..else..if..then..fi..fi..* (Điều kiện lồng nhau)

với các phép toán logic AND (&&) và OR (||) để gộp nhiều điều kiện lại với nhau

f. Vòng lặp

Vòng lặp for cho phép chúng ta thực thi các câu lệnh một số lần cụ thể.

```
#!/bin/bash

for i in {1..5}
do
    echo $i
done

student01@lab-computer:~$ ./script.sh
1
2
3
4
5
```

Hoặc sử dụng vòng lặp while với điều kiện dừng:

```
#!/bin/bash
i=1
while [[ $i -le 5 ]] ; do
    echo "$i"
    (( i += 1 ))
done

student01@lab-computer:~$ ./script.sh
1
2
3
4
5
```

g. Lưu kết quả từ một lệnh Bash

Trong trường hợp chúng ta cần lưu đầu ra của một lệnh phức tạp trong Bash Script, chúng ta có thể viết câu lệnh bên trong dấu `**<lệnh>**` hoặc **\$(lệnh)** :

```
#!/bin/bash

var=`df -h | grep tmpfs`
echo $var

student01@lab-computer:~$ ./script.sh
tmpfs 201M 22M 179M 11% /run tmpfs 1001M 192K 1000M 1% /dev/shm tmpfs 5.0M 4.0K 5.0M 1% /run/lock tmpfs 1001M 0 1001M 0% /sys/fs/cgroup tmpfs 201M 0 201M 0% /run/user/1001 tmpfs 201M 24K 201M 1% /run/user/112 tmpfs 201M 0 201M 0% /run/user/1000 tmpfs 201M 0 201M 0% /run/user/1003
```



TRƯỜNG ĐẠI HỌC BÁCH KHOA – ĐẠI HỌC QUỐC GIA TP.HCM
TRUNG TÂM KỸ THUẬT ĐIỆN TOÁN

268 Lý Thường Kiệt, Phường 14, Quận 10, TP.Hồ Chí Minh
Điện thoại: 84-8-3864 7256 ext 5371 – Fax: 84-8-3865 8687
Website: www.cce.hcmut.edu.vn
E-mail: dientoan@hcmut.edu.vn

h. Định nghĩa hàm

Một hàm Bash về cơ bản là một tập hợp các lệnh có thể được gọi nhiều lần. Mục đích của một hàm là giúp chúng ta làm cho các tập lệnh bash của mình dễ đọc hơn và tránh việc viết cùng một đoạn mã lặp đi lặp lại.

Cấu trúc hàm trên Bash:

```
functionName () {  
  
    first command  
  
    second command  
  
    ...  
  
}
```

Ví dụ:

```
#!/bin/bash  
  
hello_world () {  
    echo 'hello everyone'  
}  
  
hello_world  
  
student01@lab-computer:~$ ./script.sh  
hello everyone
```

4. Bài tập ví dụ

Tạo một ứng dụng máy tính đơn giản có thể thực hiện các phép toán số học cơ bản như cộng, trừ, nhân hoặc chia tùy thuộc vào số mà người dùng nhập vào Bash. Ví dụ:

```
Enter two numbers:  
5.6  
3.4  
Enter Choice:  
1. Addition  
2. Subtraction  
3. Multiplication  
4. Division  
3  
5.6 * 3.4 = 19.0
```

a. Phân tích yêu cầu:

Chúng ta sẽ thực hiện các bước sau:

- Đọc hai số a và b
- Lựa chọn phép toán (1-phép cộng, 2-phép trừ, 3-phép nhân, 4-phép chia):
 - o Nếu lựa chọn bằng 1: Tính $res = a + b$
 - o Nếu lựa chọn bằng 2: Tính $res = a - b$
 - o Nếu lựa chọn bằng 3: Tính $res = a * b$



TRƯỜNG ĐẠI HỌC BÁCH KHOA – ĐẠI HỌC QUỐC GIA TP.HCM
TRUNG TÂM KỸ THUẬT ĐIỆN TOÁN

268 Lý Thường Kiệt, Phường 14, Quận 10, TP.Hồ Chí Minh
Điện thoại: 84-8-3864 7256 ext 5371 – Fax: 84-8-3865 8687
Website: www.cce.hcmut.edu.vn
E-mail: dientoan@hcmut.edu.vn

- Nếu lựa chọn bằng 4: Tính $res = a / b$
- Xuất kết quả đầu ra: res

b. Chương trình hoàn chỉnh

```
#!/bin/bash

# Take user Input
echo "Enter Two numbers : "
read a
read b

# Input type of operation
echo "Enter Choice : "
echo "1. Addition"
echo "2. Subtraction"
echo "3. Multiplication"
echo "4. Division"
read ch

# Switch Case to perform
# calculator operations
case $ch in
    1)res=`echo $a + $b | bc`
    ;;
    2)res=`echo $a - $b | bc`
    ;;
    3)res=`echo $a \* $b | bc`
    ;;
    4)res=`echo "scale=2; $a / $b" | bc`
    ;;
    esac
echo "Result : $res"
```

VI. BÀI TẬP THỰC HÀNH

Từ ví dụ ứng dụng máy tính cơ bản bằng Shell Script mà chúng ta đã thực hành phía trên, mỗi học viên tiến hành xây dựng một ứng dụng máy tính nâng cao thỏa mãn các yêu cầu sau:

1. Yêu cầu nhập liệu

- Cho phép nhập trực tiếp phép tính vào với định dạng 2 ngôi, phân cách bởi khoảng trắng:
<number> <operator> <number>
Ví dụ: “2 + 5”
- Hỗ trợ 5 phép tính cơ bản: Cộng (+), Trừ (-), Nhân (*), Chia (/), Chia lấy nguyên (%)
- Lưu trữ kết quả tính toán gần nhất vào biến ANS và có thể truy cập lại biến này tại phép tính tiếp theo. Ví dụ:

```
~$ ./calc.sh
>> 2 + 5
7
```



TRƯỜNG ĐẠI HỌC BÁCH KHOA – ĐẠI HỌC QUỐC GIA TP.HCM
TRUNG TÂM KỸ THUẬT ĐIỆN TOÁN

268 Lý Thường Kiệt, Phường 14, Quận 10, TP.Hồ Chí Minh
Điện thoại: 84-8-3864 7256 ext 5371 – Fax: 84-8-3865 8687
Website: www.cce.hcmut.edu.vn
E-mail: dientoan@hcmut.edu.vn

```
~$ ./calc.sh  
>> ANS + 3  
10
```

- Biến ANS được khởi tạo với giá trị là 0 và không bị mất đi sau khi khởi động lại chương trình
- Lưu trữ lịch sử của 5 phép tính thành công gần nhất và sử dụng lệnh đầu vào *HIST* để xem lại:

```
~$ ./calc.sh  
>> HIST  
1 + 2 = 3  
ANS - 4 = -1  
3 * 5 = 15  
9 % 6 = 1  
25 / 6 = 4.17
```

- Khi chờ nhập phép toán, chỉ hiển thị duy nhất “>> ”(Lưu ý: Có 1 dấu cách ở sau):

```
~$ ./calc.sh  
>>
```

- Sau khi nhập phép tính hợp lệ vào (có dấu cách giữa các toán tử), ấn ENTER để ra kết quả (Lưu ý: In ra kết quả kèm kí tự xuống dòng):

```
~$ ./calc.sh  
>> 1 + 2  
3
```

2. Yêu cầu hiển thị:

- Đối với các kết quả không phải số nguyên (số thập phân), làm tròn tới 2 chữ số thập phân:

```
~$ ./calc.sh  
>> 5 / 3  
1.67
```

- Sau khi hoàn thành một phép tính, ấn phím bất kì để bắt đầu một phép tính toán mới với thông tin hiển thị được thiết lập lại về ban đầu:

```
>>
```

- Đối với các phép toán không hợp lệ (Chia cho 0), hiển thị *MATH ERROR*:

```
~$ ./calc.sh  
>> 1 / 0
```




TRƯỜNG ĐẠI HỌC BÁCH KHOA – ĐẠI HỌC QUỐC GIA TP.HCM
TRUNG TÂM KỸ THUẬT ĐIỆN TOÁN

268 Lý Thường Kiệt, Phường 14, Quận 10, TP.Hồ Chí Minh
Điện thoại: 84-8-3864 7256 ext 5371 – Fax: 84-8-3865 8687
Website: www.cce.hcmut.edu.vn
E-mail: dientoan@hcmut.edu.vn

MATH ERROR

- Đối với các đầu vào không đúng cú pháp, hiển thị *SYNTAX ERROR*:

```
~$ ./calc.sh  
>> 1 @ 2  
SYNTAX ERROR
```

- Nhập *EXIT* để thoát chương trình

```
~$ ./calc.sh  
>> EXIT  
~$
```

3. Cách thức nộp bài:

- Nộp duy nhất một tập tin có thể thực thi: *calc.sh* theo hướng dẫn của Giảng viên

VII. TÀI LIỆU THAM KHẢO

- CLI là gì? Và vì sao nên dùng CLI thay vì GUI: <https://www.hostinger.vn/huong-dan/cli-la-gi>
- ZSH là gì và tại sao bạn nên sử dụng nó thay vì Bash?: <https://vi.phhsnews.com/articles/howto/what-is-zsh-and-why-should-you-use-it-instead-of-bash.html>
- Bash scripting cheatsheet: <https://devhints.io/bash>
- Simple Calculator in Bash: <https://www.geeksforgeeks.org/simple-calculator-bash/>