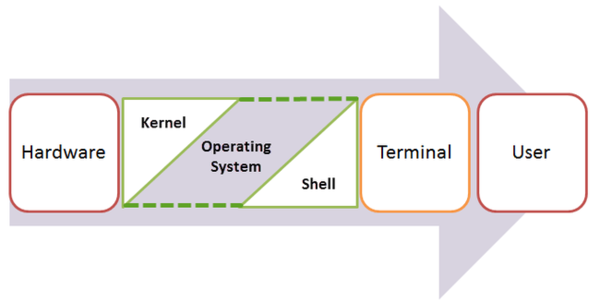
1. Hệ điều hành mở cho phép người dùng sử dụng miễn phí, tự do phát triển và định hướng hay tùy chỉnh theo nhu cầu thực tế của mình (được gọi là distro Linux hay distribution Linux).
2. **Understand how to use shell**
   * + Shell trình biên dịch người dùng đến Kernel



* File.sh: là các file shell script chứa các chuỗi lệnh cho hệ thống
* Các comment của shell đứng trước bằng dấu #
* #!bin/bash : chương trình được diễn giải và chạy bởi bash
* Chmod đặt quyền thực thi cho các người dùng khác nhau: Read (r)4,write(w)2,excute(x)1
* Ctrl+R: tìm kiếm lệnh đã thực thi
* Ctrl+P: dịch lên trên 1 lệnh
* Ctrl+N: dịch xuống dưới 1 lệnh
* Ctrl+A to move the cursor to the beginning of the line
* Sử dụng các câu lệnh trên LINUX:
* ls: lấy danh sách tất cả các file và thư mục trong thư mục hiện tại.
* pwd: xuất đường dẫn của thư mục làm việc.
* cd: thay đổi thư mục làm việc đến một thư mục mới.
* mkdir: tạo thư mục mới.
* rmdir: xoá thư mục rỗng.
* cp: copy một hay nhiều tập tin đến thư mục mới.
* mv: đổi tên hay di chuyển tập tin, thư mục. mv old\_file new\_file
* rm: xóa tập tin.
* wc: đếm số dòng, số kí tự... trong tập tin.
* touch: tạo một tập tin.
* cat: xem nội dung tập tin.
* vi: khởi động trình soạn thảo văn bản vi.
* df: kiểm tra dung lượng đĩa.
* du: xem dung lượng đĩa đã dùng cho một số tập tin nhất định
* nano: Khởi dộng trình soạn thảo văn bản nano
* less: Xem nội dung tập tin theo dòng
* tail: Xem nội dung tập tin (mặc định xem 10 dòng cuối, muốn xem 100 dòng cuối thì dùng lệnh sau: tail 100 tenfile)
* more: Xem nội dung tập tin theo trang
* head: Xem nội dung tập tin (mặc định xem 10 dòng đầu, muốn xem 100 dòng đầu thì dùng lệnh sau: head 100 tenfile)
* find <tiêu chuẩn cần tìm>: tìm tên file
* grep <Tiêu chuẩn cần tìm>: tìm nội dung trong file
* Các lệnh liên quan đến tiến trình:
* ps: kiểm tra hệ thống tiến trình đang chạy.
* kill: dừng tiến trình khi tiến trình bị treo.
* pstree: hiển thị tất cả các tiến trình dưới dạng cây.
* sleep: cho hệ thống ngừng hoạt động trong một khoảng thời gian.
* wget: tải các ứng dụng từ một website về
* Lệnh nén và giải nén:
* tar  -cvf: nén file/thư mục sang định dạng .tar
* tar  -xvf: giải nén file .tar

1. **Understand shell environment variables**
   * Cách chạy 1 file shell ( đuôi .sh):

* bash hello.sh
* sh hello.sh
* ./hello.sh

Biến trong shell:

* Không thể sử dụng các ký tự khác như !, \*, hoặc - , bởi vì các ký tự này có ý nghĩa đặc biệt đối với shell.
* Biến hệ thống: là chữ in hoa
* Biến do người dùng đặt: chữ thường và phải bắt đầu bằng kí tự, không được bằng số
* Không có giá trị khởi tạo thì bằng NULL
* Variable\_name=variable value

VD: VAR1=”khanh”

VAR2=100

Để truy cập các giá trị được giữ trong một biến, đặt trước tên biến đó với ký hiệu ( $)

VD: NAME = “Khanh”

echo $NAME

* Không có dấu cách 2 bên toán tử = khi gán giá trị cho biến
* Shell cung cấp một cách để đánh dấu các biến như read-only bằng cách sử dụng lệnh **readonly**. Sau khi một biến được đánh dấu read-only, giá trị của nó không thể thay đổi.
* Xóa 1 biến trong linux: Xóa một biến nói cho shell để gỡ bỏ biến đó từ danh sách các biến mà nó theo dõi.

NAME = “Khanh”

unset NAME

echo $NAME

note: Không thể dùng unset xóa biến readonly

* Xem toàn bộ biến môi trường: env
* Các kí tự toán học

expr toántử 1 toánhạng toántử 2

Các dấu ngoặc:

* + Tất cả các kí tự trong dấu ngoặc kép đều không có giá trị tính toán
  + Dấu nháy ngược: yêu cầu thực hiện lệnh

Cấu trúc điều khiển trong shell script:

1. GETTING HELP: Sử dụng lệnh man để tra cứu

Các option của 1 số lệnh cơ bản:

* + Lệnh ls:

ls -l: Hiển thị thư mục và các file có trong thư mục hiện tại với nhiều thông tin hơn như các quyền read, write, excute

ls -a: Hiển thị các file có thư mục ẩn

ls \*.txt: hiển thị các file có đuôi là txt

* + Lệnh cd:

cd / : di chuyển đến thư mục gốc của user hiện tại

cd ..: lùi về thu mục trước đó

* + Lệnh cat:

cat filename.txt: hiển thị trong file

cat >filename.txt: xóa trong file và thêm dữ liệu mới vào file

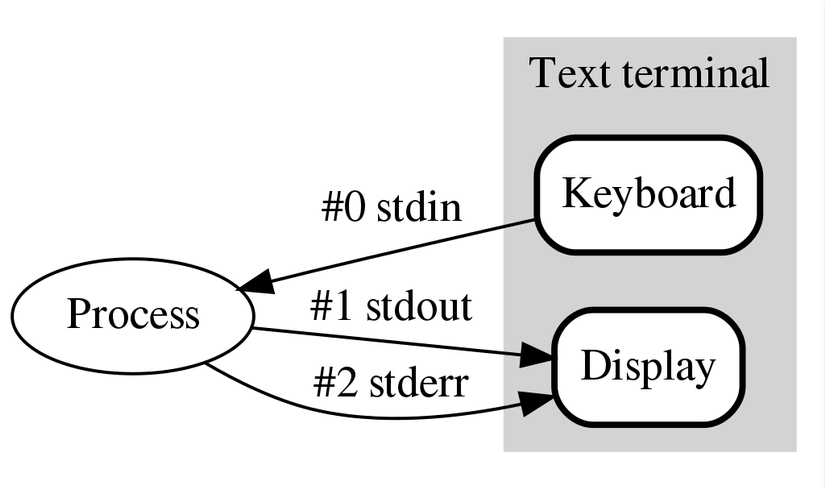
* + Lệnh rm:

rm : xóa thư mục trống

rm -rf: xóa thư mục là các file, thư mục con của nó.

1. Using stream, Redirection, and Pipes

* Stream



Standard Input (stdin): có file desciptor là 0, trong hầu hết các trường hợp data đến từ keyboard của người dùng.

Standard Output (stdout): có file descriptor là 1, thể hiện trên màn hình dạng text-mode hoặc GUI.

Standard Error (stderr): có file descriptor là 2, thể hiện các thông tin quan trọng như lỗi ra màn hình như stdout, ta có thể tách biệt stderr ra 1 file riêng so với stdout để theo dõi.

* Redirecting:

VD: echo $LOGNAME >test.txt

nano test.txt hiển thị ra khanhnt

| Redirection Operator | Chức năng |
| --- | --- |
| > | Tạo file mới chứa stdout, nếu file đã tồn tại sẽ ghi đè lên. Không cần file descriptor kèm theo |
| >> | Chèn thêm stdout vào file có sẵn, nếu file không tồn tại sẽ tạo file mới. Không cần file descriptor kèm theo |
| 2> | Tạo file mới chứa stderr, nếu file đã tồn tại sẽ ghi đè lên. Cần file descriptor kèm theo |
| 2>> | Chèn thêm stderr vào fiel có sẵn, nếu file chưa tồn tại sẽ tạo mới. Cần file descriptor kèm theo |
| &> | Tạo file mới chứa stdout và stderr, nếu file đã tồn tại sẽ ghi đè lên. Không cần file descriptor kèm theo |
| < | Gửi nội dung file ra màn hình input. Không cần file descriptor kèm theo |
| << | Chỉ gửi những dòng khai báo ra màn hình input. Không cần file descriptor kèm theo |
| > | Tạo file mới chứa stdout, nếu file đã tồn tại sẽ ghi đè lên. Không cần file descriptor kèm theo |
| <> | Nội dung file được dùng cho cả input và output. Không cần file descriptor kèm theo |

Tips:

Khi muốn chương trình nhận input từ file cho đến dòng tồn tại ký tự EOF (End of File), ta gõ lệnh:

file\_name<<EOF

* Piping Date between programs

Pipe data được sử dụng trong trường hợp ta muốn lấy output của program1 đem làm input của program2. Cách sử dụng như sau:

$program 1 | program 2 | program 3