



CÁC DÒNG LỆNH CƠ BẢN

NỘI DUNG

- Kiến trúc hệ thống Linux
- Giới thiệu về shell
- Hệ thống tập tin
- Các lệnh cơ bản
- Quản lý tập tin và thư mục
- Một số lệnh hữu ích



2.1. KIẾN TRÚC HỆ THỐNG LINUX

- Kernel: thành phần quan trọng nhất, là trung tâm điều khiển hệ điều hành Linux.
- Shell: cung cấp tập lệnh cho người dùng, tương tác với hệ điều hành và cung cấp khả năng lập trình đơn giản.
- Ứng dụng và tiện ích



ĐĂNG NHẬP HỆ THỐNG

localhost login:

Password:

```
[root@localhost~]#
```

```
[root@localhost~]$
```

- Kết thúc phiên làm việc: exit, Ctrl + D hoặc logout
- Tắt cả các tiến trình đang chạy đều kết thúc.
- Linux phân biệt chữ hoa – thường.
- Shell mặc định trên Linux là bash (GNU Bourne-Again SHell)



ĐỊNH DANH (identifier)

- Mỗi người dùng có một định danh duy nhất trong hệ thống. Đối với người dùng, định danh chính là username duy nhất trong hệ thống.
- Đối với hệ thống, một định danh người dùng bao gồm:
 - UID (user identifier)
 - GID (group identifier)
- Lệnh hiển thị thông tin về UID và GID: id
- Lệnh cho biết định danh: whoami hoặc who am i



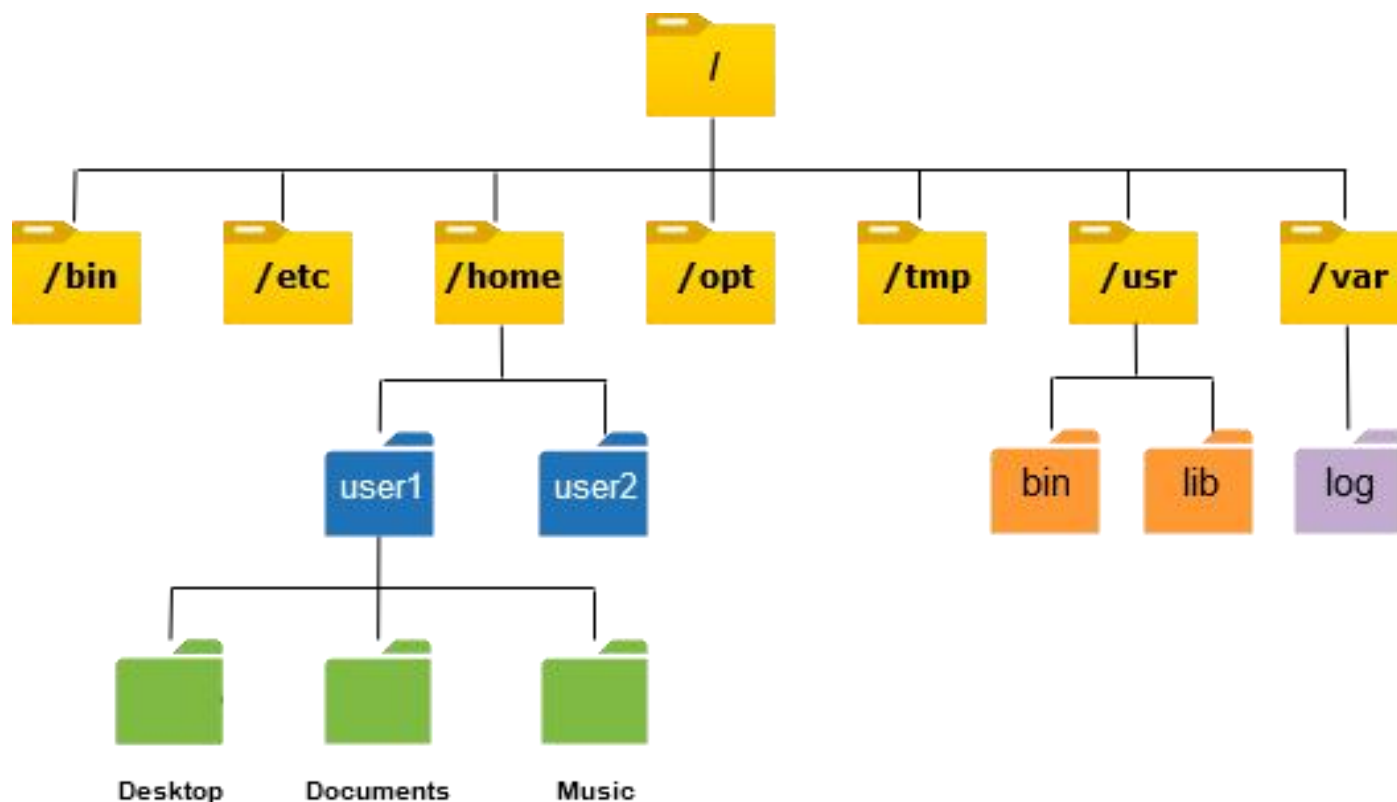
THƯ MỤC LÀM VIỆC

- Mỗi người dùng trong hệ thống đều có một thư mục home và người đó có toàn quyền trong thư mục này.
- Thư mục home:
\$ echo \$HOME
- Thư mục làm việc (hay thư mục hiện hành)
\$ pwd
\$ who



CẤU TRÚC HỆ THỐNG TẬP TIN

- Cấu trúc cây Nút: thư mục (directory, folder)
- Lá: tập tin (file)
- Điểm bắt đầu: gốc (root), kí hiệu /



CẤU TRÚC HỆ THỐNG TẬP TIN

- **/bin, /sbin**: chứa các lệnh cần thiết cho hệ thống
- **/dev**: tập tin thiết bị hoặc các file đặc biệt
- **/etc**: chứa các file cấu hình của Linux
- **/lib**: kernel modules, thư viện chia sẻ cho các lệnh nằm trong **/bin, /sbin**
- **/mnt, /media**: (mount point) dùng để ánh xạ các phân vùng đĩa



CẤU TRÚC HỆ THỐNG TẬP TIN

- **/proc**: những thông số của kernel
- **/boot**: Linux kernel, system map cho bước 2 của bootloader
- **/home**: thư mục người dùng
- **/root**: thư mục của root (admin, người quản trị)
- **/tmp**: thư mục tạm
- **/usr**: tài nguyên (tĩnh, chia sẻ) cho người dùng



CÁC KIỂU TẬP TIN

4 kiểu file cơ bản

- File thông thường (program, text, library...)
- Thư mục (là file đặc biệt, chứa thông tin của các file khác)
- File đặc biệt (device, socket, pipe...)
- File liên kết symbolic
 - Lưu ý: có thể dùng lệnh *file path_name* để biết *path_name* là loại file gì.



QUY ƯỚC ĐẶT TÊN FILE

- Tối đa 255 ký tự
- Có thể sử dụng bất kỳ kí tự nào kể cả các ký tự đặc biệt
Ví dụ: “*Very ? long – file + name . test*”
- Tập tin/thư mục ẩn bắt đầu bằng một dấu chấm “.”
Ví dụ: “.bash_history”



QUY ƯỚC ĐẶT TÊN FILE

Đường dẫn (path): Dùng để chỉ định một **phần tử** (nút) trên cây thư mục: thư mục hoặc tập tin

Đường dẫn tuyệt đối: bắt đầu từ gốc (/) đi qua các nút trung gian và sau cùng đến phần tử quan tâm (sử dụng dấu / để phân cách các nút trên đường dẫn)

Ví dụ: **/home/haui/linux/vidu.text**

Đường dẫn tương đối: bắt đầu từ thư mục hiện hành đi đến phần tử quan tâm (có thể đi ngược lên thư mục cha)

Ví dụ: giả sử thư mục hiện hành **/home/haui**, ta có thể sử dụng đường dẫn tương đối: **linux/ vidu.text**



- Đường dẫn đặt biệt
 - .. thư mục cha
 - . thư mục đang làm việc
 - ~ thư mục home của người dùng hiện hành



THỰC THI LỆNH

- Lệnh là một chương trình, script thông dịch, hoặc lệnh built-in của shell
- Thực thi lệnh:
\$ command
\$ /full/path/to/command
\$ relative/path/to/command
- Thi hành đồng thời nhiều lệnh
\$ command1; command2 ; command3 ...
- Pipeline (đường ống)
\$ cat vd.txt | more

Ví dụ:

```
$ ls  
$ /bin/ls  
$ local/bin/xvncb
```



CẤU TRÚC DÒNG LỆNH

command [option] [argument]

- Một số lưu ý:
 - Các thành phần phân tách bằng khoảng trắng.
 - Tùy chọn thường bắt đầu bằng “-” hoặc “--”

Ví dụ: **ls -a -l -F**

ls --color

- Nhiều tùy chọn có thể kết hợp bằng dấu “-”
ví dụ: ls -al tương đương ls -a -l



CẤU TRÚC DÒNG LỆNH

- ❑ Một số lệnh đặc biệt, không cần ký tự “-”
- ❑ Muốn xem trợ giúp dùng tham số `--help` hoặc `man`.

Ví dụ: **`ls --help`** hoặc **`man ls`**

- ❑ Ghi chú: Có khá nhiều loại shell trên Linux, kiểm tra đang dùng loại shell nào:
`echo $SHELL`



KÍ TỰ THAY THẾ

- Tên tập tin hoặc thư mục làm tham số dòng lệnh có thể ở dạng không tường minh.
- Dùng ký tự thay thế cho một phần hoặc toàn bộ tên
 - * : mọi chuỗi kể cả rỗng
 - ? : một ký tự bất kỳ
 - [...] : tương ứng với một trong các ký tự
 - [!/^] : không tương ứng
 - \ : loại bỏ ý nghĩa đặc biệt của các ký tự *,?,.)



THAY ĐỔI MẬT KHẨU

- `passwd` – thay đổi mật khẩu

`$ passwd`

New password:

Confirm new password:

- `pwd` – hiển thị thư mục hiện hành
- `Man` – xem trợ giúp cho một lệnh hay một hàm nào đó.

`$ man passwd`

- Kết thúc: `q`
- Về trang trước: `b`
- Về trang sau: `f`



CHUYỂN ĐỔI THƯ MỤC

- Lệnh: `cd <pathname>`
 - `cd ~` chuyển đến thư mục home
 - `cd ..` chuyển đến thư mục cha
 - `cd ~user` chuyển đến thư mục home của user
 - `cd path` chuyển đến thư mục path
 - Ví dụ:
 - `cd /usr`
 - `cd ../../etc`
- Lưu ý:
- `cd -P`: cho phép chuyển đến thư mục theo cấu trúc vật lý thay vì các symbolic link
 - `cd -L`: cho phép chuyển đến thư mục theo symbolic link



LIỆT KÊ NỘI DUNG THƯ MỤC

- Lệnh ls (listing directory)
 - `ls [option] path_name`
- Các tham số:
 - -a/A: liệt kê các file ẩn
 - -d: chỉ liệt kê tên của thư mục, không liệt kê nội dung.
 - -i: cho biết chỉ số của inode của file
 - -l: liệt kê đầy đủ (long listing)
 - -R: liệt kê các thư mục con đệ quy
 - -t: sắp xếp theo thời gian cập nhật
 - -F: liệt kê các file và cho biết kiểu của file



LIỆT KÊ NỘI DUNG THƯ MỤC

- - : file thông thường
- d : thư mục
- b : block device
- c : character device
- p : pipe
- s : socket
- l : symlink



TẠO THƯ MỤC

```
mkdir [option] directory ...
```

```
$ mkdir dir1
```

```
$ mkdir dir1 dir2 ...
```

Tham số:

-p: tạo thư mục cha nếu chưa tồn tại.

```
$ mkdir -p /home/tui/abc
```



XÓA THƯ MỤC

```
rmmdir [option]  directory ...
```

```
$ rmmdir dir1
```

```
$ rmmdir dir1 dir2 ...
```

Tham số:

-p: xóa thư mục cha nếu tồn tại.

```
$ rmmdir -p home/abc
```

Tương đương:

```
$ rmmdir home/abc home
```



SAO CHÉP

- Copy files:

```
$ cp [option] from to
```

- Copy directory:

```
$ cp -r from to
```

- Tham số:

-f: ghi đè không cần hỏi

-i: hỏi trước khi ghi đè

-r: copy toàn bộ thư mục kể cả thư mục con



DI CHUYỂN

- Di chuyển:

```
$ mv [option] from to
```

- Tham số:

-f: ghi đè không cần hỏi

-i: hỏi trước khi ghi đè

- Ví dụ:

```
$ mv /root/a.txt /home/b.txt
```



LÀM VIỆC TRÊN FILE

- Tạo file
- Hiển thị nội dung file
- Tìm kiếm file
- Tìm kiếm trong nội dung của file



TẠO FILE

- Tạo file và nhập vào nội dung

```
$ cat > name_of_file
```

- Sau khi nhập xong, nhấn Enter để xuống dòng.
- Ấn Ctrl + d để lưu nội dung soạn thảo vào file và kết thúc thao tác.

- Ví dụ:

```
$ cat > a.txt <enter>
```

```
day la vi du dau tien <enter>
```

```
ctrl + d
```

```
$
```

- Tạo file rỗng bằng lệnh touch

```
touch new_file
```



HIỂN THỊ NỘI DUNG FILE

- Với file có nội dung dài, dùng lệnh more
 - more filename
 - Dấu nhắc –more—(nn%) xuất hiện bên dưới màn hình.
 - Có thể dùng các phím điều khiển trong lúc đang xem nội dung file.
 - Space bar – hiển thị trang kế tiếp
 - <return> - hiển thị dòng tiếp theo
 - q – thoát khỏi màn hình more
 - b – về trang trước
 - h – xem trợ giúp
 - less: tương tự như more nhưng cho phép sử dụng các phím mũi tên di chuyển lên xuống



HIỂN THỊ NỘI DUNG FILE

- Dùng lệnh cat:

```
$ cat [option] name_of_file
```

- Tham số:

- -s: xóa bỏ các dòng trắng, chỉ để lại một dòng
- -n: đánh số thứ tự tất cả các dòng
- -b: đánh số thứ tự tất cả các dòng, ngoại trừ các dòng trắng.

- Sử dụng cat để nối tập tin:

```
$ cat a.txt b.txt > c.txt
```



HIỂN THỊ NỘI DUNG FILE

- Hiển thị n dòng đầu tiên của một tập tin văn bản, dùng lệnh head
\$ head -n filename
(mặc định hiển thị 10 dòng đầu tiên của file)
- Hiển thị n dòng sau cùng của một tập tin văn bản, dùng lệnh tail:

\$ tail -n filename

(mặc định hiển thị 10 dòng cuối cùng của file)



TÌM KIẾM FILE

- Tìm kiếm một file trong hệ thống tập tin, dùng lệnh find

```
$ find pathname -name filename -  
print
```

(Có thể dùng wildcard đặt trong dấu nháy kép)

- Ví dụ:

```
$ find / -name "*.cpp" -print
```

- Cũng có thể định vị một file bằng các lệnh which, whereis, locate.



TÌM KIẾM FILE

- Các tham số:
 - -name *chuỗi*: tìm kiếm các tập tin có chứa *chuỗi*
 - -group *tên*: tìm kiếm các tập tin thuộc nhóm *tên*
 - -user *tên*: tìm kiếm các tập tin có chủ nhân *tên*
 - -size [+/-]n: tìm kiếm các tập tin có kích thước lớn/nhỏ hơn n block (512bytes/block)
 - -mmin [+/-]n: tìm kiếm các tập tin được sửa đổi cách thời điểm hiện hành nhiều/ít hơn n phút
 - -mtime [+/-]n: ... n ngày
 - -newer *file*: tìm kiếm các tập tin được sửa đổi sau ngày của tập tin *file*



TÌM KIẾM TRONG NỘI DUNG FILE

- Tìm một chuỗi ký tự trong một tập tin bằng lệnh grep

```
$ grep pattern filename
```

pattern: chuỗi ký tự cần tìm kiếm. Nếu chuỗi ký tự có ký tự đặc biệt thì phải đặt trong dấu nháy đơn.

- Ví dụ:

```
$ grep linux /usr/man/man*/*
```

```
$ grep -n '[dD]on\'t' notes
```

```
$ grep a01 /etc/passwd
```



TÌM KIẾM TRONG NỘI DUNG FILE

- Các tham số của grep:
 - -i: không phân biệt chữ hoa với thường
 - -r: tìm tất cả kể cả các tập tin trong thư mục con
 - -n: hiển thị thông tin về dòng có chứa chuỗi cần tìm
 - -L: chỉ hiển thị tên các tập tin không chứa chuỗi cần tìm
 - -l: chỉ hiển thị tên các tập tin chứa chuỗi cần tìm
 - -e pattern: cho phép sử dụng kí tự đại diện pattern trong chuỗi cần tìm kiếm.



SO SÁNH HAI TẬP TIN

- So sánh hai tập tin có kiểu bất kì và ghi kết quả ra thiết bị xuất chuẩn:

```
$ cmp [-l] file1 file2
```

-l: xuất ra danh sách tất cả các vị trí khác nhau (số byte, hệ 10).
- Ví dụ:

```
$ cmp a.txt b.txt
```

```
$ cmp -l /root/a.txt /home/sv/b.txt
```



SO SÁNH HAI TẬP TIN

- Tìm sự khác biệt giữa hai tập tin:
`$ diff [option] from-file to-file`
- Các tùy chọn:
 - i: không phân biệt chữ hoa với thường
 - s: hiển thị thông báo nếu 2 tập tin giống nhau
 - w: bỏ qua khoảng trắng giữa các từ
 - r: so sánh tất cả các tập tin trong thư mục con, khi so sánh 2 thư mục



- Tạo thư mục ViDu1, ViDu2. Trong ViDu1, tạo thư mục Bai1, tạo 1 file baitap.txt
- Sao chép file baitap.txt từ Bai1 sang ViDu2



TÍNH NĂNG BASH SHELL

- Mở rộng ký tự thay thế (wildcard)
- Tự động điền dòng lệnh
- Lịch sử dòng lệnh
- Định hướng lại nhập/xuất
- Ống lệnh (pipe)



MỞ RỘNG KÝ TỰ THAY THẾ

- Tên file hoặc thư mục làm tham số dòng lệnh có thể ở dạng không tường minh.
- Dùng ký tự thay thế cho toàn bộ hoặc một phần của tên
 - * : tương ứng mọi chuỗi, kể cả chuỗi rỗng
 - ? : tương ứng một ký tự đơn
 - [...]: tương ứng một trong các ký tự bên trong
 - [!/^]: không tương ứng với một trong các ký tự bên trong ngoặc
 - \ loại bỏ ý nghĩa đặc biệt của các ký tự *, ?,)



MỞ RỘNG KÝ TỰ THAY THẾ

- Nhấn <Tab> để tự động điền đầy đủ dòng lệnh .
- Username, hostname, command, filename
- Liệt kê tất cả khả năng có thể.
- History: in ra danh sách các lệnh đã thi hành.
- !n: thực thi lại dòng lệnh thứ n.
- !string: thực thi lại dòng lệnh ngay trước đó bắt đầu bằng “string”

