



LAB 1

CÀI ĐẶT HỆ ĐIỀU HÀNH VÀ PHẦN MỀM - SỬ DỤNG LỆNH CƠ BẢN

Họ tên và MSSV: Huỳnh Trung Tín B2012045

Nhóm học phần: Nhóm 7

- Các sinh viên bị phát hiện sao chép bài của nhau sẽ nhận 0đ cho tất cả bài thực hành của môn này.
- Bài nộp phải ở dạng PDF, hình minh họa phải rõ ràng chi tiết. Hình minh họa chỉ cần chụp ở nội dung thực hiện, không chụp toàn màn hình.

1. Sử dụng Linux

Tìm hiểu và thực hiện các yêu cầu sau:

1.1. Linux distribution (gọi tắt là distro) là gì? Giữa các distro giống và khác nhau thế nào?

- Linux chính là một hệ điều hành được tạo bởi Linus Torvalds và sau đó phát triển thành giải pháp hệ điều hành mã nguồn mở, nhằm giảm sự lệ thuộc vào những hệ điều hành đóng, ví dụ như Microsoft Windows. Một bản phân phối Linux là một hệ điều hành được tạo dựng từ tập hợp nhiều phần mềm dựa trên hạt nhân Linux và thường có một hệ thống quản lý gói tin.
- Điểm giống và khác nhau của distro:
- Điểm giống nhau là về cơ bản, chúng đều dựa trên 3 nhánh chính, đó là Debian, Red Hat, Slackware. Đồng thời, tất cả các bản distrolinux đều có Kernel và Linux.
- Khác nhau: chủ yếu dựa vào 2 yếu tố chính: thị trường distro Linux là gì và triết lý phần mềm của chúng.
- Xét về thị trường:

Các distro Linux phổ biến và phát triển hiện nay được chia thành 4 nhóm:

- Nhóm 1: Arch, Gentoo và Slackware
 - + Các bản distrolinux này nhắm vào người dùng am hiểu Linux. Do đó, phần lớn các phương thức xây dựng, cũng như cấu hình của hệ thống được thực hiện qua dòng lệnh.
- Nhóm 2: Debian, Fedora
 - + Đối tượng người dùng của nhóm 2 là người am hiểu về hệ thống nhưng chưa thực sự hiểu về Linux. Vì vậy, distro sẽ cung cấp cho họ nhiều công cụ hơn. Nhóm này phù hợp với người dùng mới bắt đầu sử dụng Linux.
 - + Tuy nhiên, các distro của nhóm 2 lại có quy trình phát triển và kiểm tra chất lượng phần mềm khắt khe hơn các nhóm còn lại. Do đó, để trở thành lập trình viên chính thức của nhóm này, bạn buộc phải có thời gian đóng góp dài. Đồng thời, được chứng nhận chất lượng bởi những lập

trình viên khác. Vì thế, giới công nghệ luôn đánh giá cao môi trường của nhóm Debian, Fedora.

- Nhóm 3: Centos, RHEL, SUSE EL
 - + Các bản distrolinux nhắm vào thị trường máy chủ, doanh nghiệp, cơ quan... Vì chúng có sự ổn định cao, thời gian ra phiên bản mới lâu, khoảng 3 – 5 năm tùy distrolinux. Ngoài ra, còn có dịch vụ hỗ trợ thương mại cho công ty, hướng dẫn sử dụng sản phẩm.
- Nhóm 4: Ubuntu, Open SUSE, Linux Mint
 - + Đối tượng khách hàng của nhóm 4 là người mới bắt đầu dùng Linux và người dùng cuối. Đặc tính của chúng là phát triển trong thời gian ngắn, ứng dụng các công nghệ mới liên tục, nhiều công cụ đồ họa để thiết kế và cấu hình hệ thống theo nhu cầu sử dụng. Nhóm này cũng rất thân thiện với người dùng mới làm quen Linux.
- Xét về triết lý phần mềm(Distro Philosophy):
 - + Triết lý phần mềm là những nguyên tắc, hay định hướng, mục tiêu của người phát triển chúng đặt ra. Vì thế, khi xét về yếu tố này thì distro cũng được phân thành 4 nhóm.
 - Nhóm 1: nhóm này có cấu trúc gọn, linh hoạt để các lập trình viên có thể xây dựng theo nhu cầu của mình.
 - Nhóm 2: nhóm này nhắm đến sự chuẩn hóa quá trình phát triển phần mềm, nhằm tạo ra hệ thống hoạt động nhịp nhàng và hạn chế tối đa lỗ hổng bảo mật.
 - Nhóm 3: phát triển theo hướng bền vững, chuyên nghiệp, phù hợp cho việc cung cấp dịch vụ/sản phẩm dài hạn, có vòng đời lên tới 7 năm.
 - Nhóm 4: đi theo hướng công nghệ. Nhóm này có nhiều công cụ hiệu ứng đồ họa và không cần cấu hình nhiều.

1.2. Kể tên ít nhất 3 Linux distro và một vài thông tin mô tả về các bản distro này.

1) Red Hat Linux:

- Red Hat Linux là một trong những distro thương mại đầu tiên của GNU/Linux và được xem là chuẩn nhất trong số các distro khác. Phiên bản 1.0 đầu tiên ra mắt vào năm 1995 nhưng chỉ vài tháng sau đó, phiên bản 2.0 bổ sung công nghệ RPM (RPM Package Manager – Trình quản lý gói tin của Red Hat) tiếp tục được xuất bản. Sự ra đời của RPM giúp việc cài đặt, cập nhật, xóa, bổ sung các gói phần mềm trở nên dễ dàng hơn trước rất nhiều.
- Các ngôn ngữ lập trình được sử dụng trong Red Hat: C chiếm vị trí quan trọng với hơn 60% số lượng các dòng mã lệnh (hơn 30 triệu dòng mã lệnh), kế tiếp là C++ với khoảng 10 triệu dòng mã lệnh và theo sau là Shell (khoảng 3 triệu dòng mã lệnh).

2) Fedora:

- Fedora là phiên bản miễn phí cho người dùng, đồng thời cũng là phiên bản thử nghiệm đưa ra cộng đồng sử dụng và phản hồi, từ đó những

đặc tính nổi bật của Fedora sẽ đúc kết vào phiên bản RHEL. Fedora là distro dựa trên RPM. Fedora có thể dùng cho máy tính để bàn và máy trạm, thậm chí máy chủ. Từ đĩa CD/DVD, người dùng có thể cài đặt tất cả các ứng dụng, dịch vụ, thư viện cần thiết của Fedora và sau đó chỉ việc sử dụng chúng. Tuy nhiên, cách quản lý gói tin của Fedora gây không ít khó khăn cho người dùng.

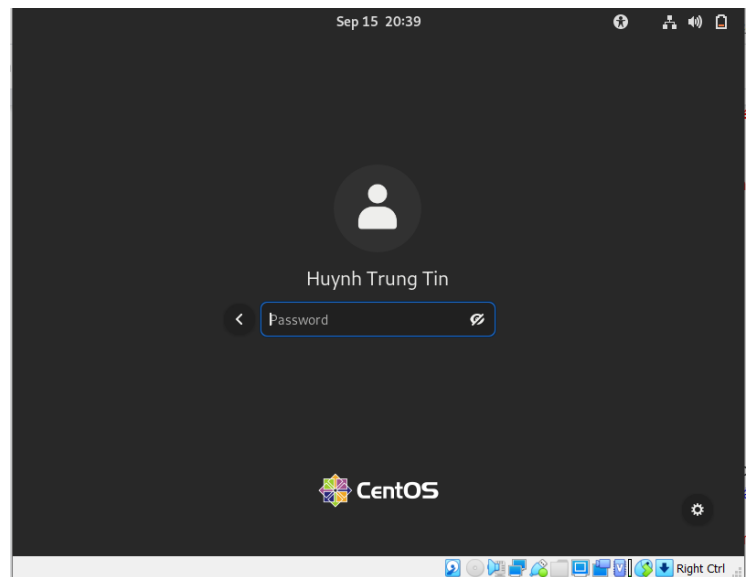
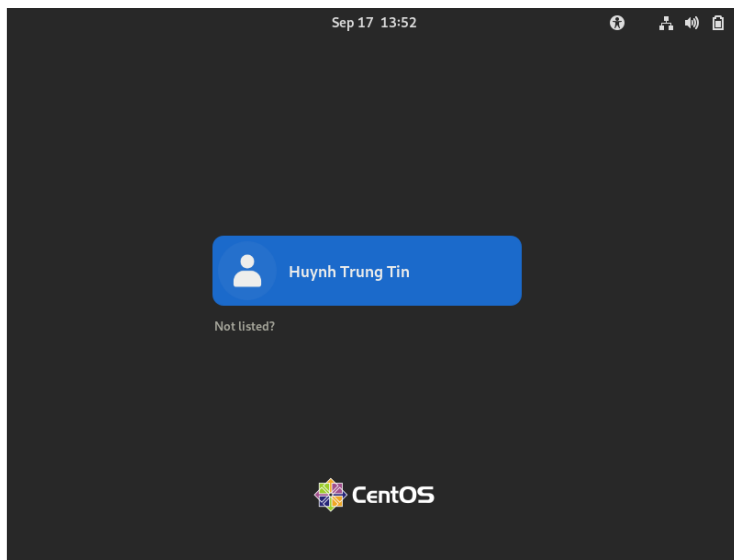
3) Debian

- Debian GNU/Linux là hệ điều hành (HĐH) PMTD mà sử dụng nhân Linux để phát triển distro riêng của nó. Vào năm 2007, Debian hỗ trợ cho nhiều kiến trúc khác nhau như Intel x86, ARM, Motorola, PowerPC, Alpha, SPARC.
- Quay lại những năm đầu, Debian GNU/Linux 3.0 (gọi là Woody) có phiên bản không ổn định là Sid, phiên bản thử nghiệm là Sarge. Phiên bản Debian GNU/Linux 4.0 được tung ra vào tháng 4/ 2007 (tên gọi là Etch) cấu tạo từ hơn 10.000 gói nguồn với hơn 288 triệu dòng mã lệnh. Ngôn ngữ sử dụng phổ biến nhất trong Debian GNU/Linux 4.0 là C với hơn 155 triệu dòng mã lệnh, nhưng thực tế C không còn chiếm ưu thế (trước đây C chiếm đến 80%).
- Debian cũng nổi tiếng với hệ thống quản lý gói, cụ thể APT (Advanced Packaging Tool - công cụ quản lý gói cao cấp). Debian có chính sách nghiêm ngặt đối với chất lượng các gói và bản phát hành cũng như tiến trình phát triển và kiểm tra mở. Cách này giúp cho việc nâng cấp các bản phát hành cũng như việc đặt hay gỡ bỏ dễ dàng hơn.

2. Cài đặt CentOS

Thực hiện cài đặt CentOS Stream 9 vào máy tính cá nhân (hoặc máy ảo) của bạn. Trong quá trình cài đặt:

- Đặt mật khẩu cho tài khoản root.
- Tạo một tài khoản có tên đăng nhập (User name) là mã số sinh viên, tên đầy đủ (Full name) là tên của sinh viên. [Cấp quyền quản trị cho tài khoản \(Make this user administrator\)](#).
- Sau khi hoàn thành cài đặt, **chụp màn hình đăng nhập** có chứa login name để chứng tỏ hoàn thành việc cài đặt.



3. Shell và lệnh Linux cơ bản

Tìm hiểu và thực hiện các yêu cầu sau:

3.1. Shell là gì? Kể tên một số shell trong Linux. Làm sao để biết shell nào đang được sử dụng bởi Linux?

- Shell là một giao diện để bạn tương tác với hệ điều hành của mình. Khi bạn nhập các lệnh vào shell, nó có trách nhiệm diễn giải các lệnh đó. Các hoạt động như sao chép tệp, piping, tệp danh sách đều nằm trong quyền hạn của shell.
- Một số shell có sẵn trong Linux:
- C shell: có cú pháp biểu thức toán học giống ngôn ngữ lập trình C.
- BASH (Bourne-Again Shell): cung cấp một số cải tiến cho người dùng: Hoàn thành lệnh, lịch sử lệnh
- KSH (Korn Shell): Là phiên bản kế thừa của Bourne shell – hỗ trợ mọi thứ mà Bourne shell có. Nó có các tính năng tương tác lập trình tiện lợi như mảng số học, thao tác chuỗi và các hàm giống ngôn ngữ C.
- Có 3 cách để biết shell nào đang được sử dụng bởi Linux:
 - + `echo $0:` sẽ in tên chương trình mà trong trường hợp shell là shell thực.
 - + `ps -ef | grep $$ | grep -v grep:` tìm kiếm ID quy trình hiện tại trong danh sách các quy trình đang chạy. Vì tiến trình hiện tại là shell nên nó sẽ được đưa vào.
 - + `echo $SHELL:` Đường dẫn đến shell hiện tại được lưu trữ dưới dạng biến SHELL cho bất kỳ shell nào.

3.2. Để tìm kiếm thông tin hướng dẫn về một lệnh hoặc tiện ích nào đó trong Linux, ta có thể sử dụng những câu lệnh nào? Cho ví dụ (chụp hình minh họa).

- Tìm kiếm thông tin hướng dẫn về một lệnh hoặc tiện ích nào đó trong Linux, ta có thể dùng các lệnh:

1) Lệnh ls - -help

```
B2012045@localhost:~  
[B2012045@localhost ~]$ ls --help
```

*Kết quả sau khi thực hiện lệnh:

```
B2012045@localhost:~  
[B2012045@localhost ~]$ 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 .  
-A, --almost-all        do not list implied . and ..  
                        with -l, print the author of each file  
-b, --escape            print C-style escapes for nongraphic characters  
                        --block-size=SIZE  with -l, scale sizes by SIZE when printing them;  
                        e.g., '--block-size=M'; see SIZE format below  
-B, --ignore-backups    do not list implied entries ending with ~  
-c                      with -lt: sort by, and show, ctime (time of last  
                        modification of file status information);  
                        with -l: show ctime and sort by name;  
                        otherwise: sort by ctime, newest first  
-C                      list entries by columns  
                        --color[=WHEN]    colorize the output; WHEN can be 'always' (default  
                        if omitted), 'auto', or 'never'; more info below  
-d, --directory        list directories themselves, not their contents  
-D, --dired            generate output designed for Emacs' dired mode  
-f                      do not sort, enable -aU, disable -ls --color  
-F, --classify         append indicator (one of */=>@|) to entries
```

```
B2012045@localhost:~  
-F, --classify         append indicator (one of */=>@|) to entries  
                        --file-type      likewise, except do not append '*'  
                        --format=WORD    across -x, commas -m, horizontal -x, long -l,  
                        single-column -l, verbose -l, vertical -C  
                        --full-time      like -l --time-style=full-iso  
-g                      like -l, but do not list owner  
                        --group-directories-first  
                        group directories before files;  
                        can be augmented with a --sort option, but any  
                        use of --sort=none (-U) disables grouping  
-G, --no-group         in a long listing, don't print group names  
-h, --human-readable   with -l and -s, print sizes like 1K 234M 2G etc.  
                        --si            likewise, but use powers of 1000 not 1024  
-H, --dereference-command-line  
                        follow symbolic links listed on the command line  
                        --dereference-command-line-symlink-to-dir  
                        follow each command line symbolic link  
                        that points to a directory  
--hide=PATTERN         do not list implied entries matching shell PATTERN  
                        (overridden by -a or -A)  
--hyperlink[=WHEN]    hyperlink file names; WHEN can be 'always'  
                        (default if omitted), 'auto', or 'never'  
--indicator-style=WORD append indicator with style WORD to entry names:  
                        none (default), slash (-p),
```

```
B2012045@localhost:~  
  
--indicator-style=WORD  append indicator with style WORD to entry names:  
                        none (default), slash (-p),  
                        file-type (--file-type), classify (-F)  
-i, --inode              print the index number of each file  
-I, --ignore=PATTERN    do not list implied entries matching shell PATTERN  
-k, --kibibytes          default to 1024-byte blocks for disk usage;  
                        used only with -s and per directory totals  
-l                      use a long listing format  
-L, --dereference        when showing file information for a symbolic  
                        link, show information for the file the link  
                        references rather than for the link itself  
-m                      fill width with a comma separated list of entries  
-n, --numeric-uid-gid    like -l, but list numeric user and group IDs  
-N, --literal            print entry names without quoting  
-o                      like -l, but do not list group information  
-p, --indicator-style=slash append / indicator to directories  
-q, --hide-control-chars print ? instead of nongraphic characters  
--show-control-chars    show nongraphic characters as-is (the default,  
                        unless program is 'ls' and output is a terminal)  
-Q, --quote-name         enclose entry names in double quotes  
--quoting-style=WORD    use quoting style WORD for entry names:  
                        literal, locale, shell, shell-always,  
                        shell-escape, shell-escape-always, c, escape
```

```
B2012045@localhost:~  
--quote-name            enclose entry names in double quotes  
--quoting-style=WORD    use quoting style WORD for entry names:  
                        literal, locale, shell, shell-always,  
                        shell-escape, shell-escape-always, c, escape  
                        (overrides QUOTING_STYLE environment variable)  
-r, --reverse           reverse order while sorting  
-R, --recursive         list subdirectories recursively  
-s, --size              print the allocated size of each file, in blocks  
-S                      sort by file size, largest first  
--sort=WORD             sort by WORD instead of name: none (-U), size (-S),  
                        time (-t), version (-v), extension (-X)  
--time=WORD            change the default of using modification times;  
                        access time (-u): atime, access, use;  
                        change time (-c): ctime, status;  
                        birth time: birth, creation;  
                        with -l, WORD determines which time to show;  
                        with --sort=time, sort by WORD (newest first)  
--time-style=TIME_STYLE time/date format with -l; see TIME_STYLE below  
-t                      sort by time, newest first; see --time  
-T, --tabsize=COLS      assume tab stops at each COLS instead of 8  
-u                      with -lt: sort by, and show, access time;  
                        with -l: show access time and sort by name;  
                        otherwise: sort by access time, newest first  
-U                      do not sort: list entries in directory order
```

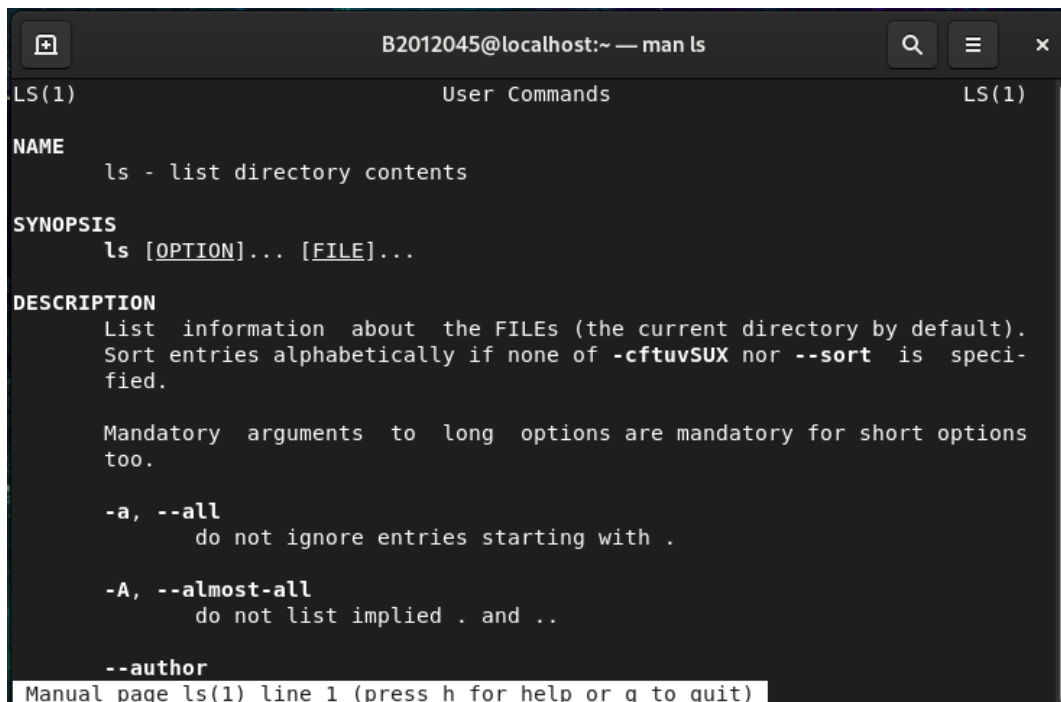
```
B2012045@localhost:~  
-U          do not sort; list entries in directory order  
-v          natural sort of (version) numbers within text  
-w, --width=COLS  set output width to COLS. 0 means no limit  
-x          list entries by lines instead of by columns  
-X          sort alphabetically by entry extension  
-Z, --context  print any security context of each file  
-l          list one file per line. Avoid '\n' with -q or -b  
--help      display this help and exit  
--version   output version information and exit  
  
The SIZE argument is an integer and optional unit (example: 10K is 10*1024).  
Units are K,M,G,T,P,E,Z,Y (powers of 1024) or KB,MB,... (powers of 1000).  
Binary prefixes can be used, too: KiB=K, MiB=M, and so on.  
  
The TIME_STYLE argument can be full-iso, long-iso, iso, locale, or +FORMAT.  
FORMAT is interpreted like in date(1). If FORMAT is FORMAT1<newline>FORMAT2,  
then FORMAT1 applies to non-recent files and FORMAT2 to recent files.  
TIME_STYLE prefixed with 'posix-' takes effect only outside the POSIX locale.  
Also the TIME_STYLE environment variable sets the default style to use.  
  
Using color to distinguish file types is disabled both by default and  
with --color=never. With --color=auto, ls emits color codes only when  
standard output is connected to a terminal. The LS_COLORS environment  
variable can change the settings. Use the dircolors command to set it.
```

```
Exit status:  
0  if OK,  
1  if minor problems (e.g., cannot access subdirectory),  
2  if serious trouble (e.g., cannot access command-line argument).  
  
GNU coreutils online help: <https://www.gnu.org/software/coreutils/>  
Full documentation <https://www.gnu.org/software/coreutils/ls>  
or available locally via: info '(coreutils) ls invocation'  
[B2012045@localhost ~]$
```

2) Lệnh man ls

```
B2012045@localhost:~  
[B2012045@localhost ~]$ man ls
```

*Kết quả sau khi thực hiện lệnh:



```
B2012045@localhost:~ — man ls
LS(1)                                User Commands                                LS(1)

NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILES (the current directory by default).
    Sort entries alphabetically if none of -cftuvSUX nor --sort is speci-
    fied.

    Mandatory arguments to long options are mandatory for short options
    too.

    -a, --all
        do not ignore entries starting with .

    -A, --almost-all
        do not list implied . and ..

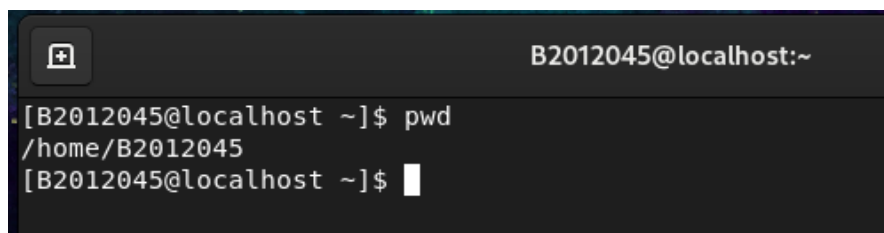
    --author

Manual page ls(1) line 1 (press h for help or q to quit)
```

3.3. Cho biết công dụng của lệnh **pwd** và **cd**. Cho ví dụ (chụp hình minh họa).

*Lệnh **pwd** được dùng để tìm đường dẫn của thư mục hiện tại (folder) mà bạn đang ở trong đó. Command này sẽ trả về đường dẫn hoàn chỉnh (đầy đủ), bắt đầu bằng dấu gạch chéo (/). Ví dụ một đường dẫn hoàn chỉnh là /home/username.

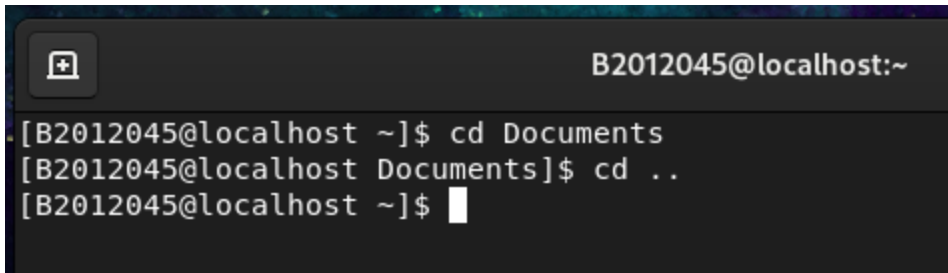
Hình minh họa:



```
B2012045@localhost:~
[B2012045@localhost ~]$ pwd
/home/B2012045
[B2012045@localhost ~]$
```

*Lệnh **cd** dùng để chuyển hướng trong hệ thống tập tin Linux, nó sẽ cần đường dẫn đầy đủ hoặc tên thư mục mà mình muốn chuyển tới.

- Có nhiều cách di chuyển nhanh bằng **cd** như sau:
 - + **cd ..** (với 2 chấm) để chuyển lên 1 cấp thư mục trên
 - + **cd** để tới thẳng thư mục home
 - + **cd-** (với dấu gạch ngang) để chuyển tới thư mục bạn đã ở trước đó



```
B2012045@localhost:~  
[B2012045@localhost ~]$ cd Documents  
[B2012045@localhost Documents]$ cd ..  
[B2012045@localhost ~]$
```

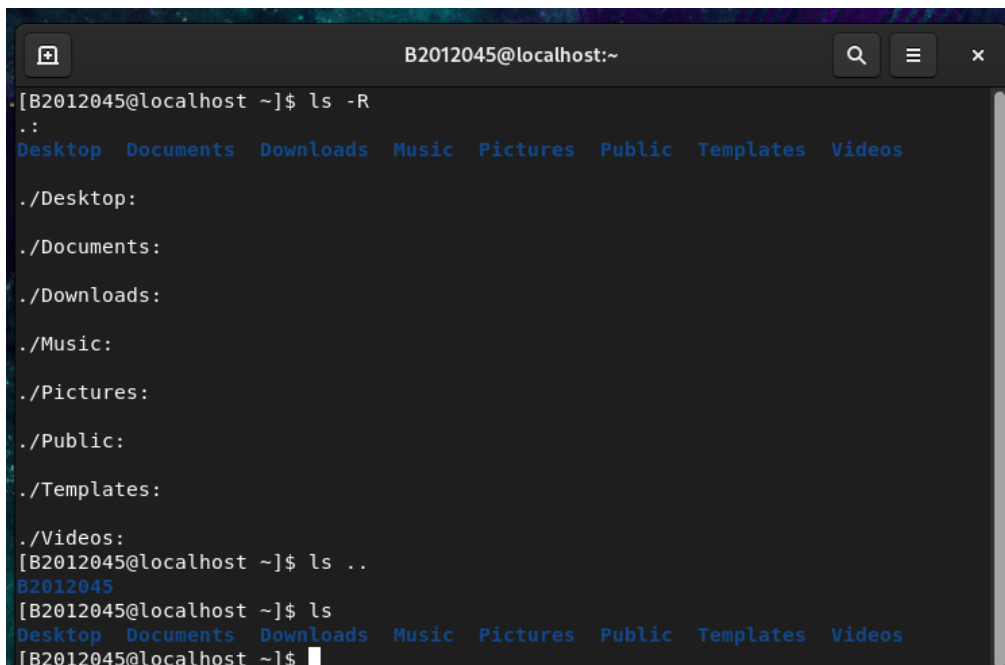
3.4. Cho biết công dụng của lệnh `ls` và vài tùy chọn của nó. Cho ví dụ (chụp hình minh họa).

Lệnh `ls` được dùng để xem nội dung thư mục. Mặc định là lệnh này sẽ hiển thị danh sách file trong thư mục hiện tại. Nếu bạn muốn xem nội dung thư mục khác, hãy nhập `ls` và sau đó là đường dẫn thư mục.

Có nhiều phiên bản để dùng với lệnh `ls` như sau:

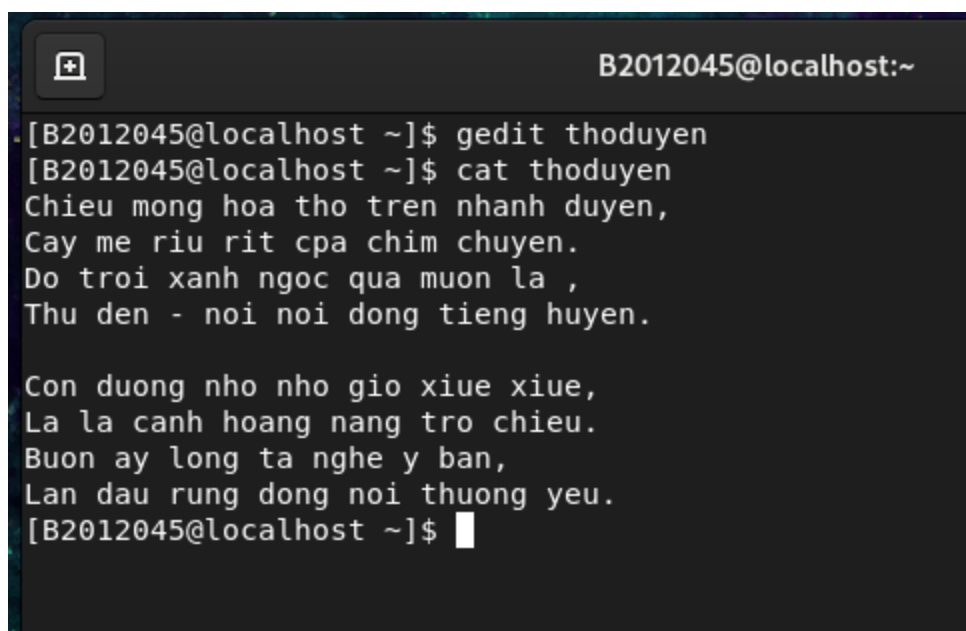
- `ls -R` liệt kê các file bao gồm cả các thư mục phụ bên trong
- `ls -a` liệt kê những file ẩn
- `ls -al` liệt kê tất cả file và thư mục với thông tin chi tiết như phân quyền, kích thước, chủ sở hữu,
- `ls ..` liệt kê các thư mục, tập tin mẹ

Hình ảnh minh họa cho một số lệnh:



```
B2012045@localhost:~  
[B2012045@localhost ~]$ ls -R  
.:  
Desktop Documents Downloads Music Pictures Public Templates Videos  
./Desktop:  
./Documents:  
./Downloads:  
./Music:  
./Pictures:  
./Public:  
./Templates:  
./Videos:  
[B2012045@localhost ~]$ ls ..  
B2012045  
[B2012045@localhost ~]$ ls  
Desktop Documents Downloads Music Pictures Public Templates Videos  
[B2012045@localhost ~]$
```

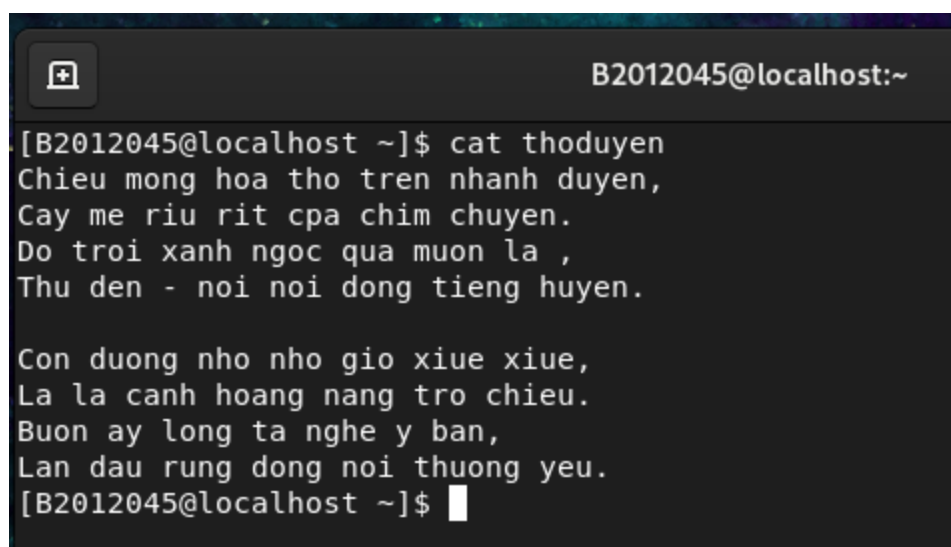
3.5. Dùng công cụ `gedit` để tạo một tập tin có tên *thoduyen* với nội dung là 8 câu đầu bài thơ Thơ Duyên của Xuân Diệu (chụp hình minh họa).



```
B2012045@localhost:~  
[B2012045@localhost ~]$ gedit thoduyen  
[B2012045@localhost ~]$ cat thoduyen  
Chieu mong hoa tho tren nhanh duyen,  
Cay me riu rit cpa chim chuyen.  
Do troi xanh ngọc qua muon la ,  
Thu đen - noi noi dong tieng huyen.  
  
Con duong nho nho gio xiu xiu,  
La la canh hoang nang tro chieu.  
Buon ay long ta nghe y ban,  
Lan dau rung dong noi thuong yeu.  
[B2012045@localhost ~]$
```

3.6. Cho biết công dụng của lệnh **cat**, **more**, **less**, **head** và **tail**. Cho ví dụ (chụp hình minh họa).

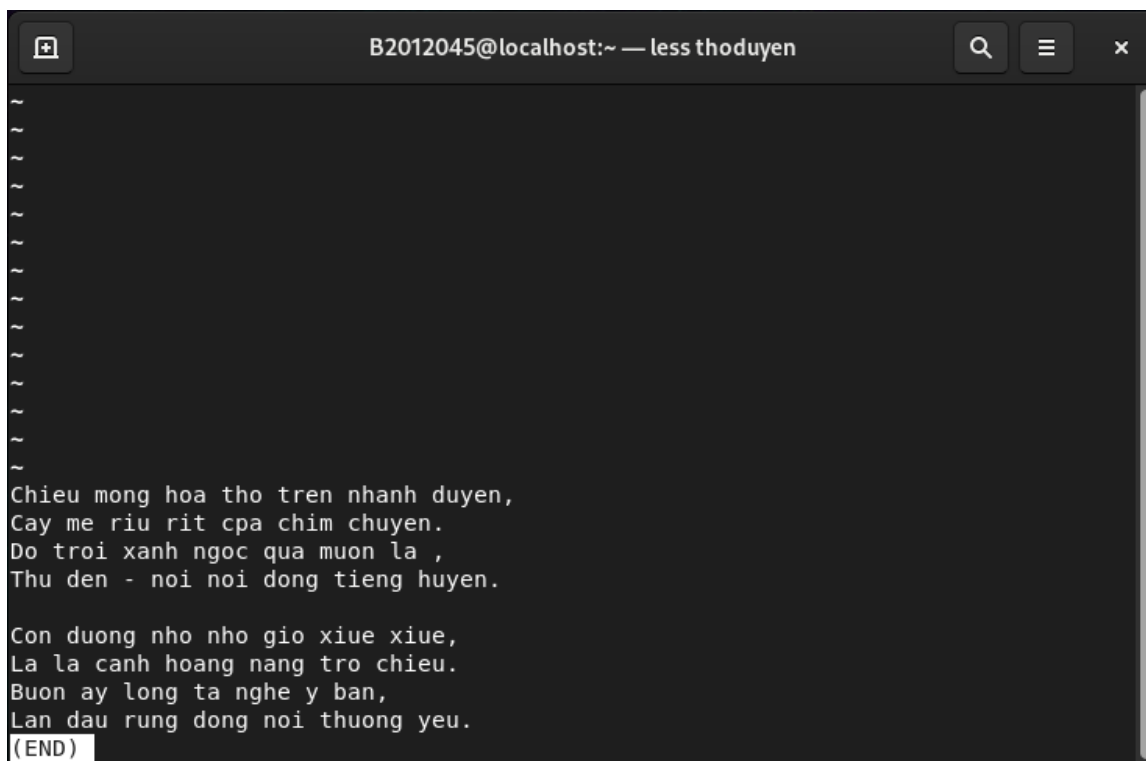
Lệnh **cat**: **cat** là một trong các lệnh cơ bản trong Linux được sử dụng thường xuyên nhất trong Linux. Nó được dùng để xem nội dung file trên output tiêu chuẩn (sdout). Để chạy lệnh này, gõ **cat** theo sau là tên file và phần mở rộng. Hình ảnh minh họa:



```
B2012045@localhost:~  
[B2012045@localhost ~]$ cat thoduyen  
Chieu mong hoa tho tren nhanh duyen,  
Cay me riu rit cpa chim chuyen.  
Do troi xanh ngọc qua muon la ,  
Thu đen - noi noi dong tieng huyen.  
  
Con duong nho nho gio xiu xiu,  
La la canh hoang nang tro chieu.  
Buon ay long ta nghe y ban,  
Lan dau rung dong noi thuong yeu.  
[B2012045@localhost ~]$
```

Lệnh **less** dùng mở một tệp để đọc tương tác, cho phép di chuyển lên xuống và tìm kiếm. Để chạy lệnh này, gõ **less** sau đó là tên file

Hình ảnh minh họa:

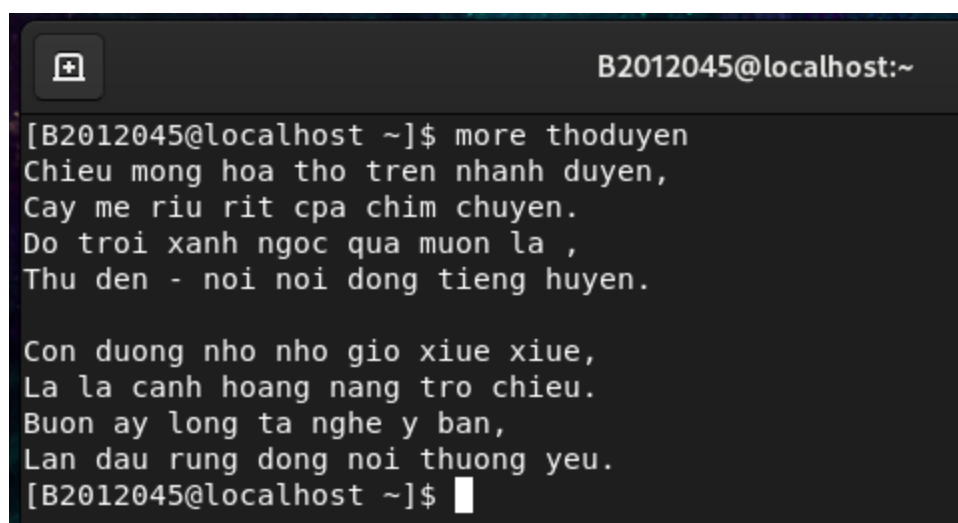


```
B2012045@localhost:~ — less thoduyen
~
~
~
~
~
~
~
~
~
~
~
~
Chieu mong hoa tho tren nhanh duyen,
Cay me riu rit cpa chim chuyen.
Do troi xanh ngọc qua muon la ,
Thu den - noi noi dong tieng huyen.

Con duong nho nho gio xiue xiue,
La la canh hoang nang tro chieu.
Bun ay long ta nghe y ban,
Lan dau rung dong noi thuong yeu.
(END)
```

Lệnh more dùng mở một tệp để đọc tương tác, cho phép di chuyển lên xuống và tìm kiếm. Để mở tệp tin, gõ more theo sau đó là tên file

Hình ảnh minh họa:



```
B2012045@localhost:~
[B2012045@localhost ~]$ more thoduyen
Chieu mong hoa tho tren nhanh duyen,
Cay me riu rit cpa chim chuyen.
Do troi xanh ngọc qua muon la ,
Thu den - noi noi dong tieng huyen.

Con duong nho nho gio xiue xiue,
La la canh hoang nang tro chieu.
Bun ay long ta nghe y ban,
Lan dau rung dong noi thuong yeu.
[B2012045@localhost ~]$
```

Lệnh head được sử dụng để xem dòng đầu tiên của bất kỳ file văn bản nào. Theo mặc định, nó sẽ hiển thị 10 dòng đầu tiên, nhưng bạn có thể thay đổi số này theo ý mình.

Hình ảnh minh họa:

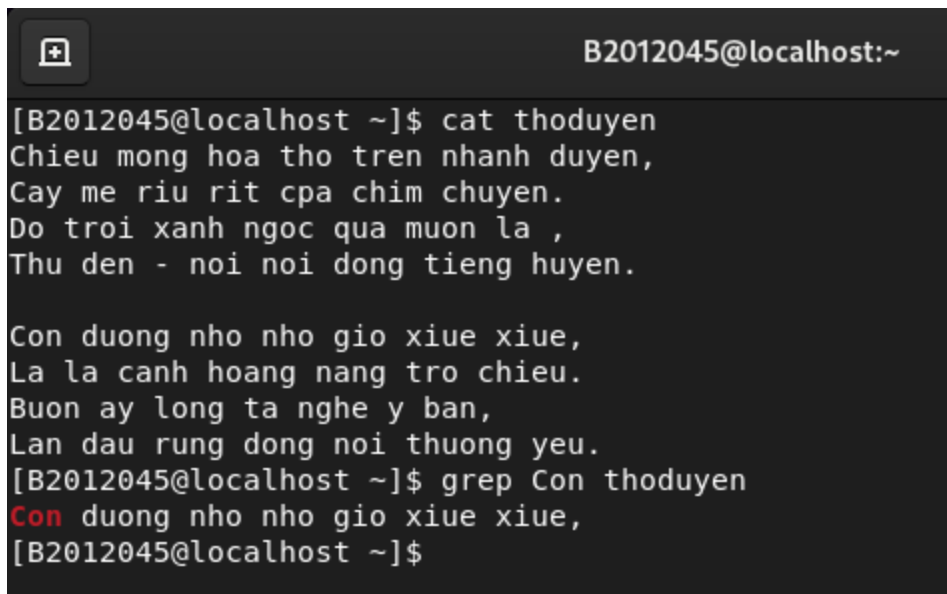
```
[B2012045@localhost ~]$ head -n 3 thoduyen
Chieu mong hoa tho tren nhanh duyen,
Cay me riu rit cpa chim chuyen.
Do troi xanh ngọc qua muon la ,
[B2012045@localhost ~]$
```

Command này có chức năng tương tự như command head, nhưng thay vì hiển thị dòng đầu tiên, command tail sẽ hiển thị 10 dòng cuối cùng của file văn bản.

```
[B2012045@localhost ~]$ tail -3 thoduyen
La la canh hoang nang tro chieu.
Bun ay long ta nghe y ban,
Lan dau rung dong noi thuong yeu.
```

3.7. Cho biết công dụng của lệnh **grep**. Cho ví dụ (chụp hình minh họa).

Command grep cho phép bạn tìm kiếm tất cả text thông qua tập tin nhất định.



```
B2012045@localhost:~
[B2012045@localhost ~]$ cat thoduyen
Chieu mong hoa tho tren nhanh duyen,
Cay me riu rit cpa chim chuyen.
Do troi xanh ngọc qua muon la ,
Thu den - noi noi dong tieng huyen.

Con duong nho nho gio xiue xiue,
La la canh hoang nang tro chieu.
Bun ay long ta nghe y ban,
Lan dau rung dong noi thuong yeu.
[B2012045@localhost ~]$ grep Con thoduyen
Con duong nho nho gio xiue xiue,
[B2012045@localhost ~]$
```

3.8. Cho biết công dụng của lệnh **cp** và **mv**. Cho ví dụ (chụp hình minh họa).

command cp để sao chép files từ thư mục hiện tại.

```
[B2012045@localhost ~]$ cp thoduyen /home/B2012045/Pictures
[B2012045@localhost ~]$ cd /home/B2012045/Pictures
[B2012045@localhost Pictures]$ ls
thoduyen
[B2012045@localhost Pictures]$
```

mv là di chuyển files, dù nó cũng có thể được dùng để đổi tên files.

```
[B2012045@localhost ~]$ mv thoduyen /home/B2012045/Music
[B2012045@localhost ~]$ cd Music
[B2012045@localhost Music]$ ls
thoduyen
[B2012045@localhost Music]$ mv thoduyen /home/B2012045/Music/newthoduyen
[B2012045@localhost Music]$ ls
newthoduyen
[B2012045@localhost Music]$
```

- 3.9. Cho biết công dụng của lệnh **mkdir** và **rm**. Cho ví dụ (chụp hình minh hoạ).
Command **mkdir** được dùng để tạo thư mục mới

```
[B2012045@localhost Desktop]$ mkdir tinB2012045
[B2012045@localhost Desktop]$ ls
tinB2012045
```

-Công dụng lệnh **rm**: xóa file và thư mục

```
[B2012045@localhost Desktop]$ rm -r tinB2012045
[B2012045@localhost Desktop]$ ls
[B2012045@localhost Desktop]$
```

4. RPM, cập nhật và cài đặt các package

- 4.1. Các tập tin có phần mở rộng **.rpm** và **.deb** có chức năng gì?

.rpm là gói cài đặt ban đầu được phát triển cho hệ điều hành Red Hat Linux, nhưng hiện cũng được sử dụng bởi nhiều bản phân phối Linux khác, thường được sử dụng để cài đặt các chương trình trên các hệ thống Linux

.deb là file Debian Software Package (gói phần mềm Debian). Chúng được sử dụng chủ yếu trong các hệ điều hành dựa trên Unix, bao gồm Ubuntu và iOS.

- 4.2. Thay vì đòi hỏi người dùng phải biết trước đường dẫn download file **.rpm** khi cài đặt/cập nhật ứng dụng có một lệnh trong RHEL/CentOS cho phép truy xuất đến kho file **.rpm**, sau đó hệ thống sẽ download file thích hợp về để thực hiện cài đặt/cập nhật. Các lệnh đó là lệnh nào?

Để cài đặt gói **.rpm** trong CentOS Linux, hãy nhập lệnh sau:

```
sudo rpm -i sample_file.rpm
```

- 4.3. Cài đặt trình soạn thảo **nano** (chụp hình minh hoạ, nếu trình soạn thảo nano đã được cài sẵn thì chụp thông báo)

```
$sudo dnf install nano
```

- 4.4. Cài đặt trình duyệt web thuần văn bản (text-based web browser) **lynx**. Sau khi cài đặt xong, thực hiện truy cập vào website <https://www.ctu.edu.vn/> (chụp hình minh hoạ).

```
$sudo dnf install lynx
```

```
[B2012045@localhost Desktop]$ cd --
[B2012045@localhost ~]$ sudo dnf install nano

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for B2012045:
CentOS Stream 9 - BaseOS                7.4 kB/s | 4.7 kB    00:00
CentOS Stream 9 - AppStream             8.4 kB/s | 4.7 kB    00:00
CentOS Stream 9 - Extras packages       9.5 kB/s | 5.0 kB    00:00
Package nano-5.6.1-5.el9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[B2012045@localhost ~]$
```

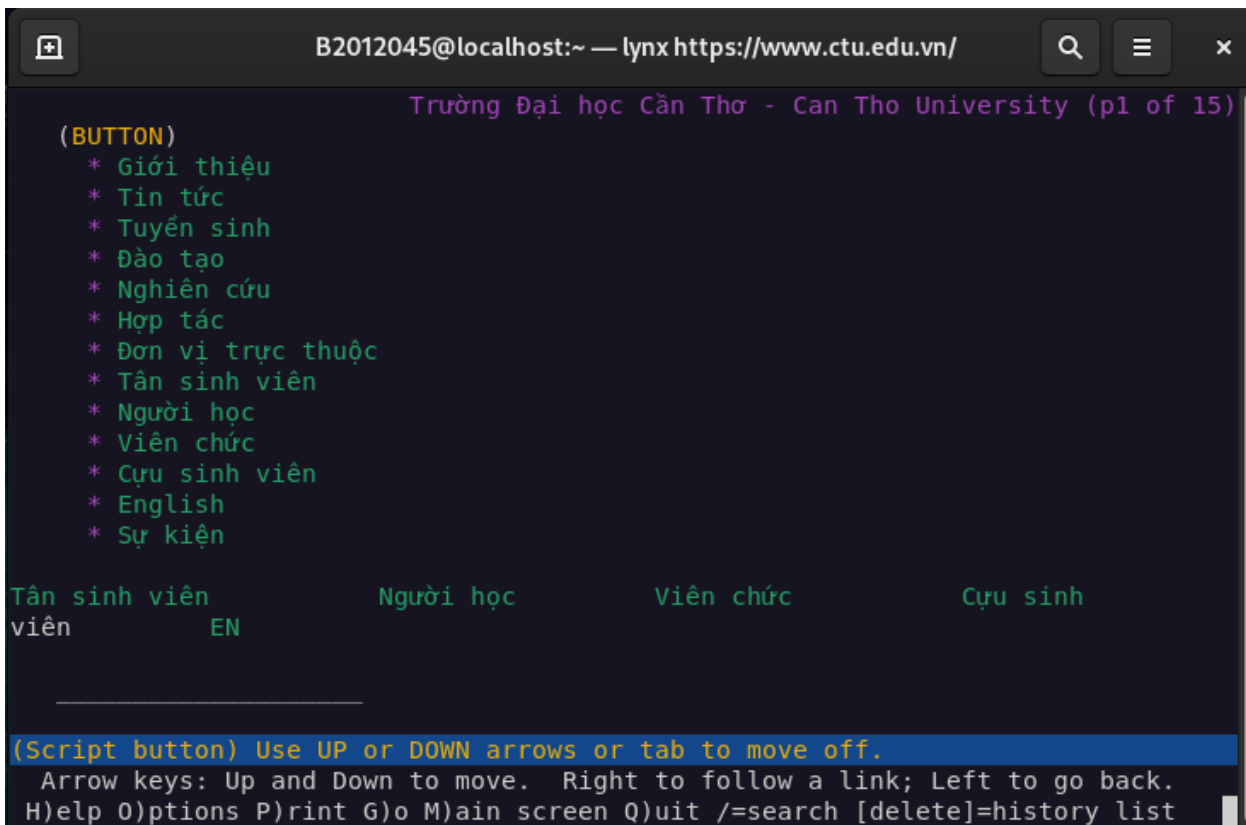
```
[B2012045@localhost ~]$ sudo dnf install lynx
Last metadata expiration check: 0:02:21 ago on Sat 17 Sep 2022 05:57:55 PM +07.
Dependencies resolved.
=====
Package                Architecture Version           Repository        Size
=====
Installing:
  lynx                  x86_64        2.8.9-19.el9     appstream         1.6 M
Transaction Summary
=====
Install 1 Package

Total download size: 1.6 M
Installed size: 6.1 M
Is this ok [y/N]: y
Downloading Packages:
lynx-2.8.9-19.el9.x86_64.rpm          1.1 MB/s | 1.6 MB    00:01
-----
Total                                800 kB/s | 1.6 MB    00:02
CentOS Stream 9 - AppStream          1.2 MB/s | 1.6 kB    00:00
Importing GPG key 0x8483C65D:
  Userid      : "CentOS (CentOS Official Signing Key) <security@centos.org>"
  Fingerprint: 99DB 70FA E1D7 CE22 7FB6 4882 05B5 55B3 8483 C65D
```

```
Total                               800 kB/s | 1.6 MB    00:02
CentOS Stream 9 - AppStream         1.2 MB/s | 1.6 kB    00:00
Importing GPG key 0x8483C65D:
  Userid      : "CentOS (CentOS Official Signing Key) <security@centos.org>"
  Fingerprint: 99DB 70FA E1D7 CE22 7FB6 4882 05B5 55B3 8483 C65D
  From        : /etc/pki/rpm-gpg/RPM-GPG-KEY-centosofficial
Is this ok [y/N]: y
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :                                1/1
  Installing     : lynx-2.8.9-19.el9.x86_64       1/1
  Running scriptlet: lynx-2.8.9-19.el9.x86_64     1/1
  Verifying      : lynx-2.8.9-19.el9.x86_64       1/1

Installed:
  lynx-2.8.9-19.el9.x86_64

Complete!
```



```
B2012045@localhost:~ — lynx https://www.ctu.edu.vn/
Trường Đại học Cần Thơ - Can Tho University (p1 of 15)

(BUTTON)
* Giới thiệu
* Tin tức
* Tuyển sinh
* Đào tạo
* Nghiên cứu
* Hợp tác
* Đơn vị trực thuộc
* Tân sinh viên
* Người học
* Viên chức
* Cựu sinh viên
* English
* Sự kiện

Tân sinh viên      Người học      Viên chức      Cựu sinh
viên              EN

(Script button) Use UP or DOWN arrows or tab to move off.
Arrow keys: Up and Down to move. Right to follow a link; Left to go back.
H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list
```

4.5. Gỡ bỏ chương trình lynx ra khỏi hệ thống (chụp hình minh họa).

```
[B2012045@localhost ~]$ sudo dnf remove lynx
[sudo] password for B2012045:
Dependencies resolved.
=====
Package           Architecture      Version           Repository        Size
=====
Removing:
  lynx             x86_64            2.8.9-19.el9      @appstream        6.1 M

Transaction Summary
=====
Remove 1 Package

Freed space: 6.1 M
Is this ok [y/N]: y
```

```
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :                                1/1
  Erasing        : lynx-2.8.9-19.el9.x86_64      1/1
  Running scriptlet: lynx-2.8.9-19.el9.x86_64      1/1
  Verifying      : lynx-2.8.9-19.el9.x86_64      1/1

Removed:
  lynx-2.8.9-19.el9.x86_64

Complete!
[B2012045@localhost ~]$
```

4.6. Thêm repository và cài đặt công cụ hỗ trợ lập trình VS Code (chụp hình minh họa).

- + Thêm Microsoft GPG key

\$ sudo rpm --import <https://packages.microsoft.com/keys/microsoft.asc>

Hình ảnh minh họa:

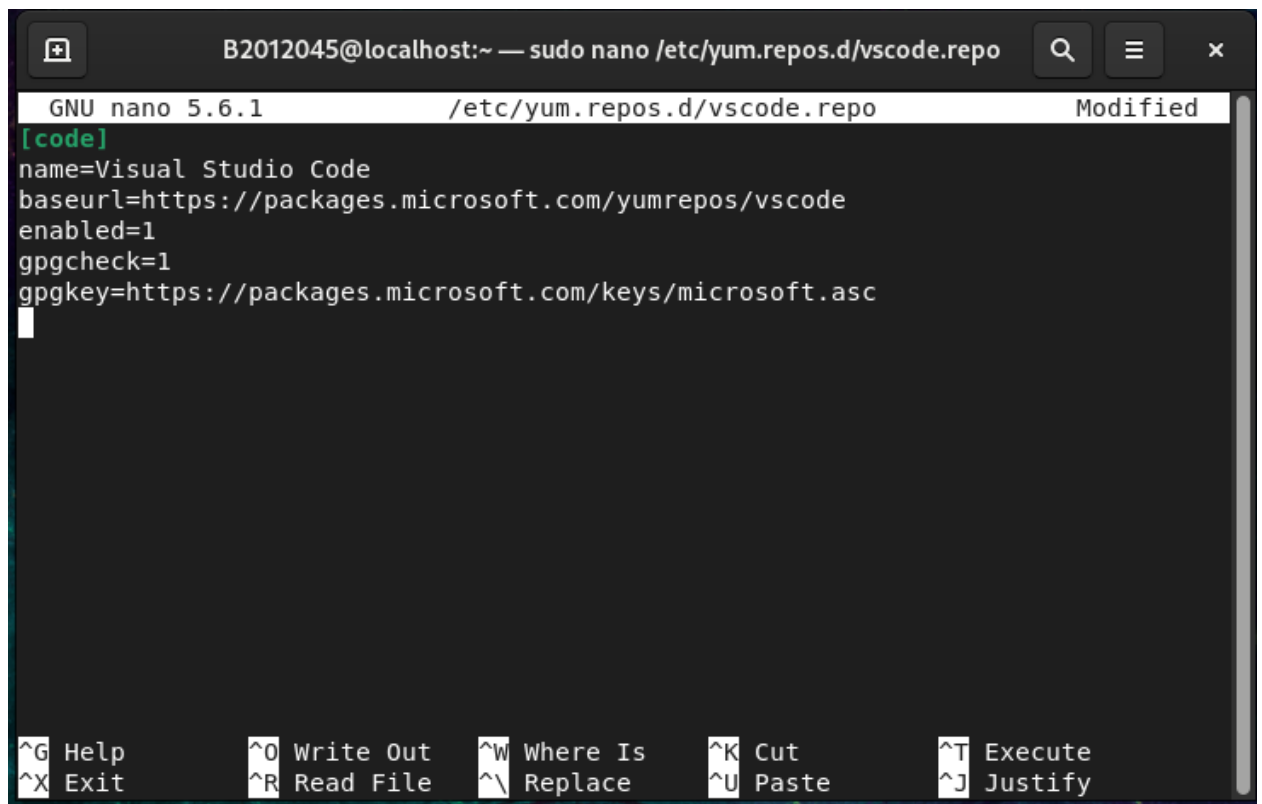
```
[B2012045@localhost ~]$ sudo --import https://packages.microsoft.com/keys/microsoft.asc
sudo: unrecognized option '--import'
usage: sudo -h | -K | -k | -V
usage: sudo -v [-AknS] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-AknS] [-g group] [-h host] [-p prompt] [-U user] [-u user]
       [command]
usage: sudo [-AbEHknPS] [-r role] [-t type] [-C num] [-D directory] [-g group]
       [-h host] [-p prompt] [-R directory] [-T timeout] [-u user]
       [VAR=value] [-i|-s] [<command>]
usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-D directory] [-g group]
       [-h host] [-p prompt] [-R directory] [-T timeout] [-u user] file
...
```


+ Tạo tập tin repository

```
$ sudo nano /etc/yum.repos.d/vscode.repo
```

Nội dung tập tin `vscode.repo`

```
[code]
name=Visual Studio Code
baseurl=https://packages.microsoft.com/yumrepos/vscode
enabled=1
gpgcheck=1
gpgkey=https://packages.microsoft.com/keys/microsoft.asc
```



```
B2012045@localhost:~ — sudo nano /etc/yum.repos.d/vscode.repo
GNU nano 5.6.1 /etc/yum.repos.d/vscode.repo Modified
[code]
name=Visual Studio Code
baseurl=https://packages.microsoft.com/yumrepos/vscode
enabled=1
gpgcheck=1
gpgkey=https://packages.microsoft.com/keys/microsoft.asc
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify
```

+ Cập nhật cache của `dnf` và cài đặt

```
$ dnf check-update
```

```
[B2012045@localhost ~]$ sudo rpm --import https://packages.microsoft.com/keys/microsoft.asc
[sudo] password for B2012045:
[B2012045@localhost ~]$ sudo nano /etc/yum.repos.d/vscode.repo
[B2012045@localhost ~]$ dnf check-update
CentOS Stream 9 - BaseOS                224 kB/s | 5.9 MB      00:26
CentOS Stream 9 - AppStream             615 kB/s | 15 MB      00:25
CentOS Stream 9 - Extras packages       1.3 kB/s | 8.5 kB      00:06
Visual Studio Code                     2.3 MB/s | 29 MB      00:12
[B2012045@localhost ~]$
```

```
$ sudo dnf install code
```

```
[B2012045@localhost ~]$ sudo dnf install code
[sudo] password for B2012045:
Visual Studio Code                2.4 MB/s | 29 MB      00:12
Last metadata expiration check: 0:00:17 ago on Sat 17 Sep 2022 08:01:21 PM +07.
Dependencies resolved.
=====
Package           Architecture Version                      Repository      Size
=====
Installing:
code              x86_64      1.71.2-1663191299.el7      code            115 M
Transaction Summary
=====
Install 1 Package

Total download size: 115 M
Installed size: 343 M
Is this ok [y/N]: y
Downloading Packages:
code-1.71.2-1663191299.el7.x86_64.rpm      2.1 MB/s | 115 MB      00:55
-----
Total                                       2.1 MB/s | 115 MB      00:55
Running transaction check
Transaction check succeeded.
```

```
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :                                1/1
  Installing     : code-1.71.2-1663191299.el7.x86_64 1/1
  Running scriptlet: code-1.71.2-1663191299.el7.x86_64 1/1
  Verifying      : code-1.71.2-1663191299.el7.x86_64 1/1

Installed:
  code-1.71.2-1663191299.el7.x86_64

Complete!
```

4.7. Lệnh nào thực hiện cập nhật tất cả các ứng dụng trong hệ thống (chụp hình minh họa)?

```
[B2012045@localhost ~]$ sudo dnf update
Last metadata expiration check: 0:03:42 ago on Sat 17 Sep 2022 08:01:21 PM +07.
Dependencies resolved.
Nothing to do.
Complete!
[B2012045@localhost ~]$
```

--- Hết ---