

VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



HỆ ĐIỀU HÀNH (TN) (CO2018)

Report

Lab 1

Người hướng dẫn: Thầy Trần Trương Tuấn Phát
Sinh viên: Phan Đức Đạt - 2113152

HO CHI MINH CITY, 9/10/2023

Contents

1	Trả lời câu hỏi	2
1.1	Question: List some other popular Linux shells and describe their highlighted features.	2
1.2	Exercise: Try more options with the ls command and analyze its output. You can also try to combine multiple options into one, ex: -la.	3
1.3	Question: Compare the Output Redirection (>/») with the Piping () technique.	6
1.4	Question: Compare the sudo and the su command.	6
1.5	Discuss about the 777 permission on critical services (web hostings, sensitive databases,...).	7
1.6	Question: 1. What are the advantages of Makefile? Give examples? 2. Compiling a program in the first time usually takes a longer time in comparison with the next re-compiling. What is the reason? 3. Is there any Makefile mechanism for other programming languages? If it has, give an example?	7
2	Kết quả chạy code	9
2.1	Kết quả chạy bài Exercise 3.6	9
2.2	Kết quả chạy bài Exercise 5.3	13

1 Trả lời câu hỏi

1.1 **Question: List some other popular Linux shells and describe their highlighted features.**

1. Bash (Bourne-Again Shell)

- Được sử dụng rộng rãi, là shell mặc định cho hầu hết các phiên bản của Linux.
- Có khả năng viết script mạnh mẽ, thường được dùng để viết shell script.
- Bash cung cấp các câu lệnh sử dụng, cho người dùng tùy chỉnh và "tab completion" giúp người dùng tiết kiệm thời gian.

2. Zsh (Z Shell)

- Có khả năng tùy chỉnh mạnh mẽ, dễ dàng trong việc sử dụng giao diện dòng lệnh.
- Hỗ trợ nhiều plugin và theme.
- Cung cấp gợi ý tự động mạnh mẽ khi gõ lệnh.

3. Fish (Friendly Interactive Shell)

- Dễ dàng trong việc sử dụng.
- Có tính năng làm nổi bật cú pháp, giúp người dùng dễ phát hiện lỗi.
- "tab completion" giúp người dùng tiết kiệm thời gian khi gõ lệnh.

4. Dash

- Có dung lượng nhỏ và tối giản, thường dùng cho viết script.
- Tuân theo tiêu chuẩn POSIX.

5. Tcsh (TENEX C Shell)

- Có cú pháp giống với ngôn ngữ lập trình C.
- Cung cấp lịch sử lệnh, bí danh, và tính năng kiểm soát công việc.

6. Ksh (Korn Shell)

- Thích hợp cho việc sử dụng tương tác và viết script.
- Tuân theo tiêu chuẩn POSIX.

7. Elvish

- Có khả năng mở rộng, và có thể lập trình với ngôn ngữ riêng của nó.
- Cung cấp nhiều tính năng tương tác và gợi ý tự động.

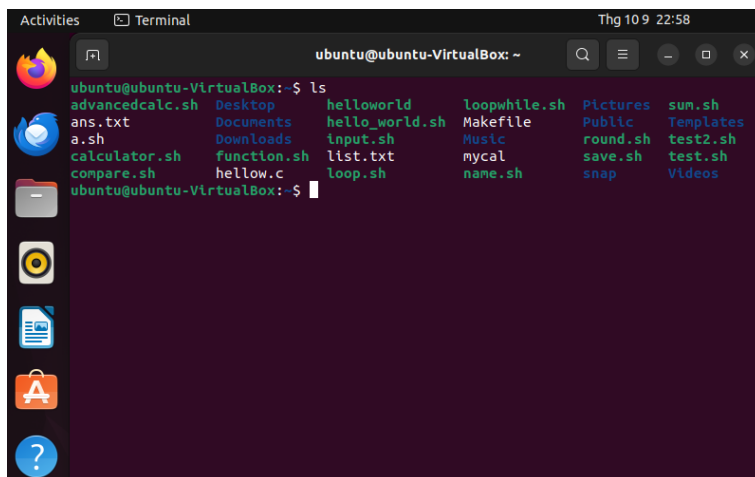
8. Oil (Oil Shell)

- Cải tiến trong việc viết script.
- Có tính tương thích cao.

1.2 Exercise: Try more options with the ls command and analyze its output. You can also try to combine multiple options into one, ex: -la.

1. ls

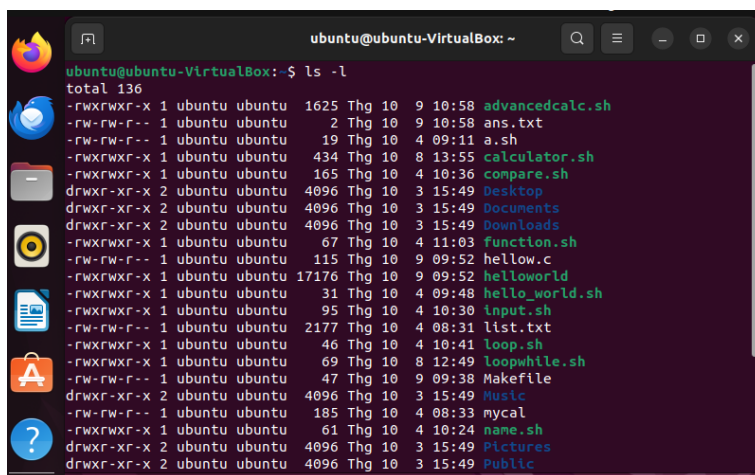
Khi chạy lệnh `ls` mà không có tùy chọn bổ sung, nó sẽ hiển thị danh sách các tập tin và thư mục trong thư mục hiện tại theo thứ tự mặc định, không bao gồm các tập tin và thư mục ẩn.



```
ubuntu@ubuntu-VirtualBox: ~  
$ ls  
advancedcalc.sh  Desktop      helloworld    loopwhile.sh  Pictures    sum.sh  
ans.txt          Documents    hello_world.sh Makefile       Public      Templates  
a.sh            Downloads    input.sh      Music          round.sh    test2.sh  
calculator.sh     function.sh  list.txt      mycal         save.sh     test.sh  
compare.sh       hellow.c    loop.sh       name.sh       snap        Videos  
ubuntu@ubuntu-VirtualBox: ~
```

2. ls -l

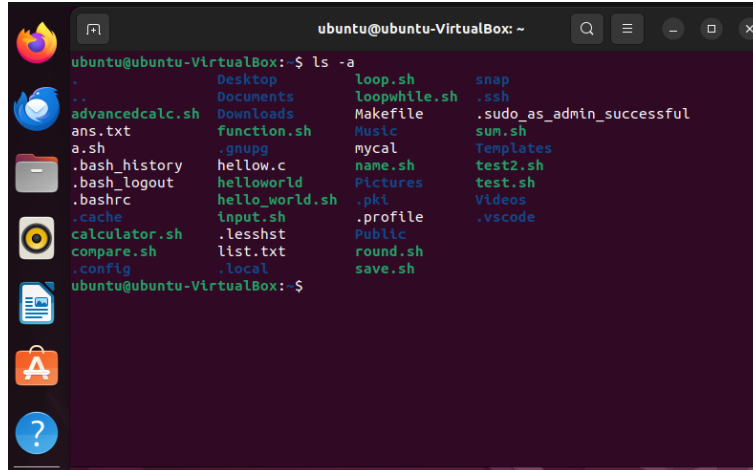
Khi chạy lệnh `ls` với tùy chọn `l`, nó sẽ hiển thị danh sách các tập tin và thư mục trong thư mục hiện tại dưới dạng danh sách chi tiết (long listing).



```
ubuntu@ubuntu-VirtualBox: ~  
$ ls -l  
total 136  
-rwxrwxr-x 1 ubuntu ubuntu 1625 Thg 10 9 10:58 advancedcalc.sh  
-rw-rw-r-- 1 ubuntu ubuntu 2 Thg 10 9 10:58 ans.txt  
-rw-rw-r-- 1 ubuntu ubuntu 19 Thg 10 4 09:11 a.sh  
-rwxrwxr-x 1 ubuntu ubuntu 434 Thg 10 8 13:55 calculator.sh  
-rwxrwxr-x 1 ubuntu ubuntu 165 Thg 10 4 10:36 compare.sh  
drwxr-xr-x 2 ubuntu ubuntu 4096 Thg 10 3 15:49 Desktop  
drwxr-xr-x 2 ubuntu ubuntu 4096 Thg 10 3 15:49 Documents  
drwxr-xr-x 2 ubuntu ubuntu 4096 Thg 10 3 15:49 Downloads  
-rwxrwxr-x 1 ubuntu ubuntu 67 Thg 10 4 11:03 function.sh  
-rw-rw-r-- 1 ubuntu ubuntu 115 Thg 10 9 09:52 hellow.c  
-rwxrwxr-x 1 ubuntu ubuntu 17176 Thg 10 9 09:52 helloworld  
-rwxrwxr-x 1 ubuntu ubuntu 31 Thg 10 4 09:48 hello_world.sh  
-rwxrwxr-x 1 ubuntu ubuntu 95 Thg 10 4 10:30 input.sh  
-rw-rw-r-- 1 ubuntu ubuntu 2177 Thg 10 4 08:31 list.txt  
-rwxrwxr-x 1 ubuntu ubuntu 46 Thg 10 4 10:41 loop.sh  
-rwxrwxr-x 1 ubuntu ubuntu 69 Thg 10 8 12:49 loopwhile.sh  
-rw-rw-r-- 1 ubuntu ubuntu 47 Thg 10 9 09:38 Makefile  
drwxr-xr-x 2 ubuntu ubuntu 4096 Thg 10 3 15:49 Music  
-rw-rw-r-- 1 ubuntu ubuntu 185 Thg 10 4 08:33 mycal  
-rwxrwxr-x 1 ubuntu ubuntu 61 Thg 10 4 10:24 name.sh  
drwxr-xr-x 2 ubuntu ubuntu 4096 Thg 10 3 15:49 Pictures  
drwxr-xr-x 2 ubuntu ubuntu 4096 Thg 10 3 15:49 Public
```

3. `ls -a`

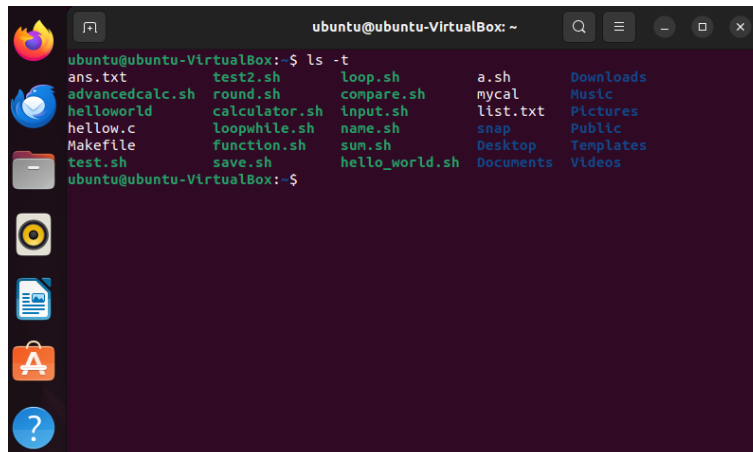
Khi chạy lệnh `ls` với tùy chọn `a`, nó sẽ hiển thị danh sách các tập tin và thư mục trong thư mục hiện tại, bao gồm cả các tập tin và thư mục ẩn (bắt đầu bằng dấu chấm `.`).



```
ubuntu@ubuntu-VirtualBox: ~  
ubuntu@ubuntu-VirtualBox:~$ ls -a  
.  
..  
advancedcalc.sh  
ans.txt  
a.sh  
.bash_history  
.bash_logout  
.bashrc  
.cache  
calculator.sh  
compare.sh  
.config  
.  
Desktop  
Documents  
Downloads  
function.sh  
.gnupg  
hellow.c  
helloworld  
hello_world.sh  
input.sh  
.lessht  
list.txt  
.local  
loop.sh  
loopwhile.sh  
Makefile  
Music  
mycal  
name.sh  
Pictures  
.pkd  
.profile  
Public  
round.sh  
save.sh  
snap  
.ssh  
.sudo_as_admin_successful  
sun.sh  
Templates  
test2.sh  
test.sh  
Videos  
.vscode  
ubuntu@ubuntu-VirtualBox:~$
```

4. `ls -t`

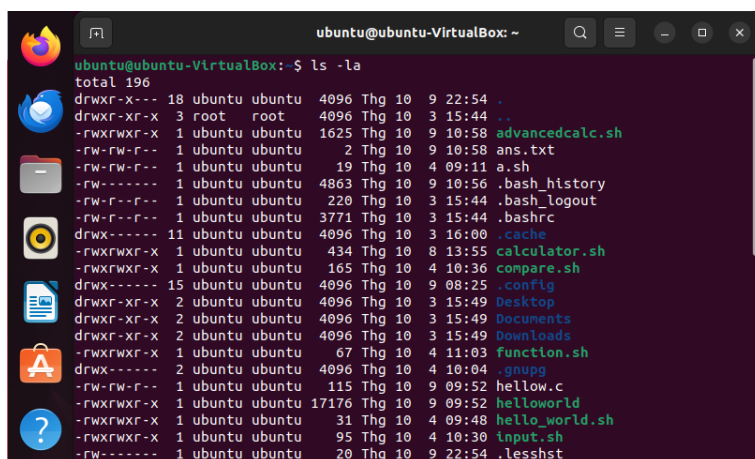
Khi chạy lệnh `ls` với tùy chọn `t`, nó sẽ hiển thị danh sách các tập tin và thư mục trong thư mục hiện tại được sắp xếp theo thời gian sửa đổi, với các tập tin hoặc thư mục mới nhất được đưa lên đầu danh sách.



```
ubuntu@ubuntu-VirtualBox: ~  
ubuntu@ubuntu-VirtualBox:~$ ls -t  
ans.txt  
test2.sh  
advancedcalc.sh  
round.sh  
helloworld  
calculator.sh  
hellow.c  
loopwhile.sh  
Makefile  
function.sh  
test.sh  
save.sh  
loop.sh  
compare.sh  
input.sh  
name.sh  
sun.sh  
hello_world.sh  
a.sh  
mycal  
list.txt  
snap  
Desktop  
Documents  
Downloads  
Music  
Pictures  
Public  
Templates  
Videos  
ubuntu@ubuntu-VirtualBox:~$
```

5. ls -la

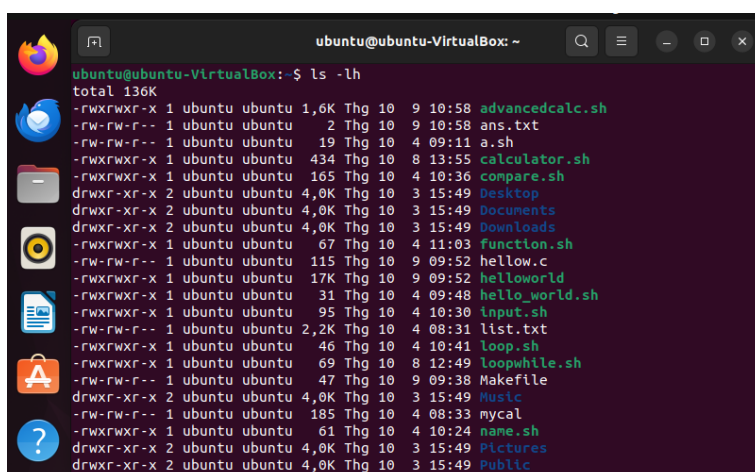
Khi chạy lệnh **ls** với tùy chọn **la**, nó sẽ hiển thị danh sách chi tiết về tất cả các tập tin và thư mục trong thư mục hiện tại, bao gồm cả các tập tin và thư mục ẩn (bắt đầu bằng dấu chấm **.**).



```
ubuntu@ubuntu-VirtualBox: ~  
ubuntu@ubuntu-VirtualBox:~$ ls -la  
total 196  
drwxr-x--- 18 ubuntu ubuntu 4096 Thg 10 9 22:54 .  
drwxr-xr-x  3 root  root  4096 Thg 10 3 15:44 ..  
-rwxrwxr-x  1 ubuntu ubuntu 1625 Thg 10 9 10:58 advancedcalc.sh  
-rw-rw-r--  1 ubuntu ubuntu  2 Thg 10 9 10:58 ans.txt  
-rw-rw-r--  1 ubuntu ubuntu  19 Thg 10 4 09:11 a.sh  
-rw-----  1 ubuntu ubuntu 4863 Thg 10 9 10:56 .bash_history  
-rw-r--r--  1 ubuntu ubuntu 220 Thg 10 3 15:44 .bash_logout  
-rw-r--r--  1 ubuntu ubuntu 3771 Thg 10 3 15:44 .bashrc  
drwx----- 11 ubuntu ubuntu 4096 Thg 10 3 16:00 .cache  
-rwxrwxr-x  1 ubuntu ubuntu  434 Thg 10 8 13:55 calculator.sh  
-rwxrwxr-x  1 ubuntu ubuntu  165 Thg 10 4 10:36 compare.sh  
drwx----- 15 ubuntu ubuntu 4096 Thg 10 9 08:25 .config  
drwxr-xr-x  2 ubuntu ubuntu 4096 Thg 10 3 15:49 Desktop  
drwxr-xr-x  2 ubuntu ubuntu 4096 Thg 10 3 15:49 Documents  
drwxr-xr-x  2 ubuntu ubuntu 4096 Thg 10 3 15:49 Downloads  
-rwxrwxr-x  1 ubuntu ubuntu  67 Thg 10 4 11:03 function.sh  
drwx-----  2 ubuntu ubuntu 4096 Thg 10 4 10:04 .gnupg  
-rw-rw-r--  1 ubuntu ubuntu  115 Thg 10 9 09:52 hellow.c  
-rwxrwxr-x  1 ubuntu ubuntu 17176 Thg 10 9 09:52 helloworld  
-rwxrwxr-x  1 ubuntu ubuntu  31 Thg 10 4 09:48 hello_world.sh  
-rwxrwxr-x  1 ubuntu ubuntu  95 Thg 10 4 10:30 input.sh  
-rw-----  1 ubuntu ubuntu  20 Thg 10 9 22:54 .lessht
```

6. ls -lh

Khi chạy lệnh **ls** với tùy chọn **lh**, nó sẽ hiển thị danh sách các tập tin và thư mục trong thư mục hiện tại dưới dạng danh sách chi tiết (long listing) và kích thước của các tập tin sẽ được hiển thị ở định dạng dễ đọc (kilobytes, megabytes, gigabytes, vv.), thay vì hiển thị kích thước tập tin gốc.



```
ubuntu@ubuntu-VirtualBox: ~  
ubuntu@ubuntu-VirtualBox:~$ ls -lh  
total 136K  
-rwxrwxr-x  1 ubuntu ubuntu 1,6K Thg 10 9 10:58 advancedcalc.sh  
-rw-rw-r--  1 ubuntu ubuntu  2 Thg 10 9 10:58 ans.txt  
-rw-rw-r--  1 ubuntu ubuntu  19 Thg 10 4 09:11 a.sh  
-rwxrwxr-x  1 ubuntu ubuntu  434 Thg 10 8 13:55 calculator.sh  
-rwxrwxr-x  1 ubuntu ubuntu  165 Thg 10 4 10:36 compare.sh  
drwxr-xr-x  2 ubuntu ubuntu 4,0K Thg 10 3 15:49 Desktop  
drwxr-xr-x  2 ubuntu ubuntu 4,0K Thg 10 3 15:49 Documents  
drwxr-xr-x  2 ubuntu ubuntu 4,0K Thg 10 3 15:49 Downloads  
-rwxrwxr-x  1 ubuntu ubuntu  67 Thg 10 4 11:03 function.sh  
-rw-rw-r--  1 ubuntu ubuntu  115 Thg 10 9 09:52 hellow.c  
-rwxrwxr-x  1 ubuntu ubuntu 17K Thg 10 9 09:52 helloworld  
-rwxrwxr-x  1 ubuntu ubuntu  31 Thg 10 4 09:48 hello_world.sh  
-rwxrwxr-x  1 ubuntu ubuntu  95 Thg 10 4 10:30 input.sh  
-rw-rw-r--  1 ubuntu ubuntu 2,2K Thg 10 4 08:31 list.txt  
-rwxrwxr-x  1 ubuntu ubuntu  46 Thg 10 4 10:41 loop.sh  
-rwxrwxr-x  1 ubuntu ubuntu  69 Thg 10 8 12:49 loopwhile.sh  
-rw-rw-r--  1 ubuntu ubuntu  47 Thg 10 9 09:38 Makefile  
drwxr-xr-x  2 ubuntu ubuntu 4,0K Thg 10 3 15:49 Music  
-rw-rw-r--  1 ubuntu ubuntu 185 Thg 10 4 08:33 mycal  
-rwxrwxr-x  1 ubuntu ubuntu  61 Thg 10 4 10:24 name.sh  
drwxr-xr-x  2 ubuntu ubuntu 4,0K Thg 10 3 15:49 Pictures  
drwxr-xr-x  2 ubuntu ubuntu 4,0K Thg 10 3 15:49 Public
```

1.3 Question: Compare the Output Redirection (>/») with the Piping (|) technique.

Kĩ thuật **Redirection**(>, ») và **Piping** là hai kĩ thuật khác nhau để điều khiển luồng dữ liệu giữa các lệnh, và nó được sử dụng cho các mục đích khác nhau.

1. Redirection(>, »)

- Mục đích: dùng để lưu trữ đầu ra của một lệnh vào trong một tập tin.
- Cách sử dụng:
Toán tử > dùng để ghi đè toàn bộ nội dung vào file output.
Toán tử » dùng để ghi nối tiếp vào file output thay vì ghi đè lên toàn bộ nội dung của file.

2. Piping (|)

- Mục đích: được sử dụng để chuyển đầu ra của một lệnh làm đầu vào cho một lệnh khác, cho phép bạn kết nối nhiều lệnh lại với nhau.
- Cách sử dụng: Ký hiệu | được sử dụng để kết nối nhiều lệnh trong một dãy lệnh.

=> Tóm lại, **Redirection** được sử dụng để chuyển hướng đầu ra của một lệnh sang một tập tin, trong khi **Piping** được sử dụng để gửi đầu ra của một lệnh như là đầu vào của một lệnh khác.

1.4 Question: Compare the sudo and the su command.

Lệnh **sudo** và **su** đều được sử dụng để thực hiện các nhiệm vụ quản trị hoặc thực thi các lệnh dưới tên một người dùng khác. Tuy nhiên, chúng có mục đích và cách sử dụng khác nhau.

1. sudo (Superuser Do)

- Mục đích: **sudo** được sử dụng để thực thi một lệnh cụ thể với quyền hạn của superuser (root) hoặc để chạy lệnh dưới tên một người dùng khác, thường với quyền hạn được nâng cao.
- Cách sử dụng: Tiền tố lệnh bạn muốn chạy bằng sudo, và nó sẽ yêu cầu bạn nhập mật khẩu của bạn (hoặc mật khẩu root, tùy thuộc vào cấu hình của hệ thống).
- Ví dụ: sudo apt-get update

2. su (Switch User)

- Mục đích: **su** được sử dụng để chuyển đổi sang môi trường của người dùng khác, thường là tài khoản root hoặc bất kỳ người dùng nào khác nếu được chỉ định.
- Cách sử dụng: Bạn sử dụng su để chuyển đổi sang môi trường người dùng khác. Theo mặc định, nó chuyển đổi sang tài khoản root và bạn phải cung cấp mật khẩu root.
- Ví dụ: su - username

=> Tóm lại, lệnh **su** được sử dụng để chuyển sang một tài khoản người dùng khác và chạy các lệnh với đặc quyền cao hơn, trong khi lệnh **sudo** được sử dụng để chạy một lệnh duy nhất với đặc quyền cao hơn mà không cần chuyển sang một tài khoản người dùng khác.

1.5 Discuss about the 777 permission on critical services (web hostings, sensitive databases,...).

Thiết lập quyền 777 cho các dịch vụ quan trọng như thư mục lưu trữ web hoặc cơ sở dữ liệu nhạy cảm nói chung là nguy hiểm. Bởi thiết lập quyền 777 là thiết lập quyền đọc, ghi, và thực thi toàn bộ cho mọi người bao gồm chủ sở hữu, nhóm, và tất cả các người dùng khác. Điều này có nghĩa là bất kỳ ai cũng có thể sửa đổi, xóa bỏ, hoặc thực thi các tập tin trong thư mục hoặc truy cập vào các tập tin cơ sở dữ liệu nhạy cảm. Dẫn đến việc rò rỉ dữ liệu, mất mát dữ liệu và không thể kiểm soát được dữ liệu.

=> Vậy nên, việc sử dụng quyền 777 trên các dịch vụ quan trọng là một rủi ro bảo mật và nên tránh bằng mọi giá. Thay vào đó, tuân thủ các phương pháp bảo mật tốt nhất, bao gồm nguyên tắc đặc quyền tối thiểu và quyền mặc định an toàn, để bảo vệ dữ liệu nhạy cảm và duy trì tính toàn vẹn cũng như tính khả dụng của các dịch vụ quan trọng.

1.6 Question:

1. What are the advantages of Makefile? Give examples?
2. Compiling a program in the first time usually takes a longer time in comparison with the next re-compiling. What is the reason?
3. Is there any Makefile mechanism for other programming languages? If it has, give an example?

1. Advantages of makefile.

- Makefile cho phép biên dịch các phần của chương trình mà đã có sự thay đổi, thay vì biên dịch lại toàn bộ chương trình => tiết kiệm thời gian.
- Makefile giúp quản lý các project phức tạp với nhiều thành phần con.
- Makefile giúp code rõ ràng và dễ dàng hơn khi đọc và debug.

• Makefile - example

```
1  # Default target: compile all
2  all: calc
3
4  # Compile calc executable from object files
5  calc: calc.o calculator.o
6      gcc -o calc calc.o calculator.o -lm
7
8  # Compile calc.c into calc.o object file
9  calc.o: calc.c calculator.h
10     gcc -c calc.c -o calc.o
11
12 # Compile calculator.c into calculator.o object file
13 calculator.o: calculator.c calculator.h
14     gcc -c calculator.c -o calculator.o
15
16 # Clean target: remove object files and executable
17 clean:
18     rm -f calc calc.o calculator.o
```


2. Compiling a program in the first time usually takes a longer time in comparison with the next re-compiling.

Biên dịch lần đầu thường lâu hơn biên dịch các lần biên dịch sau vì trình biên dịch cần xử lý tất cả các source file và header file, sau đó là chuyển sang object file. Project càng lớn thì thời gian biên dịch càng dài. Tuy nhiên, ở các lần biên dịch sau thì trình biên dịch sẽ cần ít thời gian hơn vì chỉ biên dịch các phần có sự thay đổi, còn các phần không bị thay đổi sẽ không cần biên dịch lại. (Một trong những lợi thế của Makefile).

3. Is there any Makefile mechanism for other programming languages?

Có, Makefile thường được sử dụng trong phát triển phần mềm cho nhiều ngôn ngữ lập trình khác nhau, không chỉ dành riêng cho C và C++. Make chính là một công cụ tự động hóa quy trình xây dựng không phụ thuộc vào ngôn ngữ, vì vậy bạn có thể tạo Makefile cho nhiều ngôn ngữ lập trình khác nhau để tự động hóa quá trình xây dựng và biên dịch.

Example:

• **In python:**

```
1 # Variables
2 VENV_DIR = venv
3 TEST_DIR = tests
4
5 # Targets and Rules
6 .PHONY: install test clean
7
8 install:
9     pip install -r requirements.txt
10
11 test:
12     pytest $(TEST_DIR)
13
14 clean:
15     rm -rf $(VENV_DIR) __pycache__
```

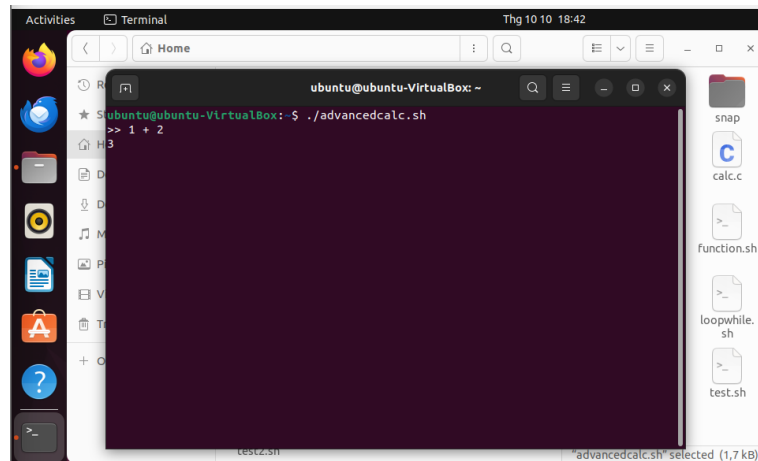
• **In Java:**

```
1 # Variables
2 JAVAC = javac
3 SOURCES = Main.java MyClass.java
4 CLASSES = $(SOURCES:.java=.class)
5
6 # Targets and Rules
7 all: $(CLASSES)
8
9 %.class: %.java
10     $(JAVAC) $<
11
12 clean:
13     rm -f $(CLASSES)
```

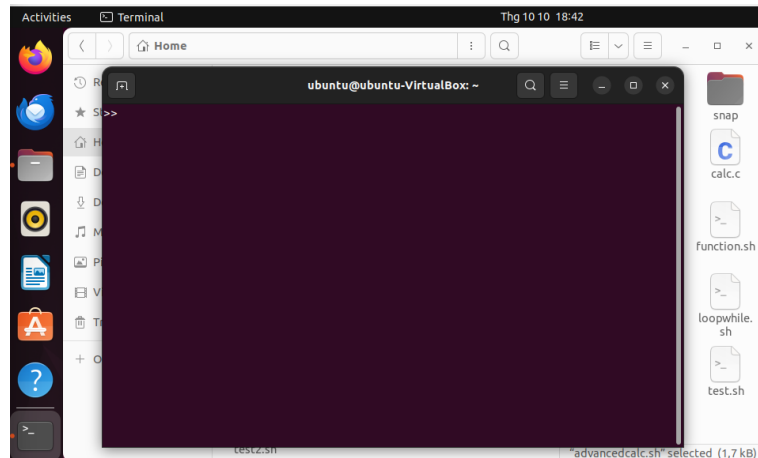
2 Kết quả chạy code

2.1 Kết quả chạy bài Exercise 3.6

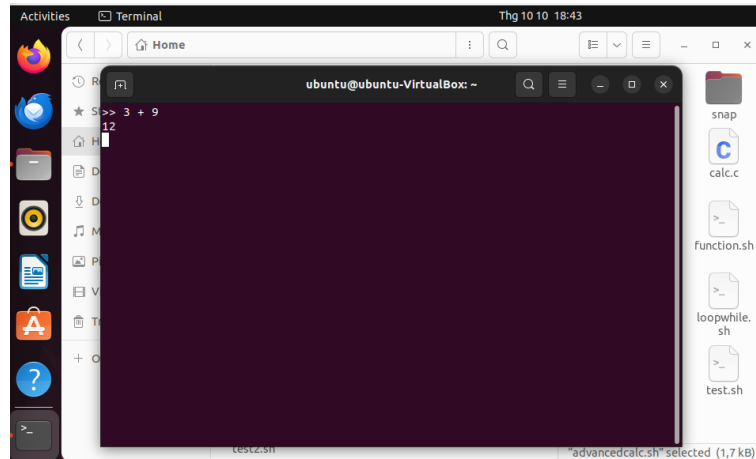
- Đầu tiên em chạy chương trình nhập vào $1 + 2$ kết quả sẽ hiện ra là 3.



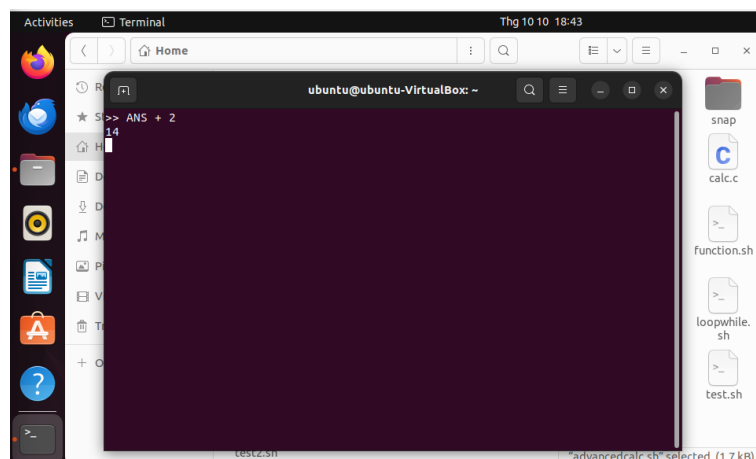
- Tiếp theo em nhấn nút bất kì trên bàn phím nó sẽ ra dấu » để nhập biểu thức mới.



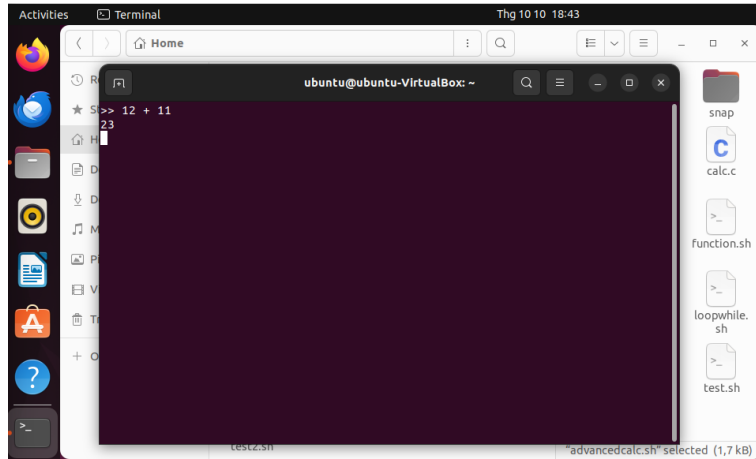
- Sau đó em nhập $3 + 9$ kết quả cho ra là 12.



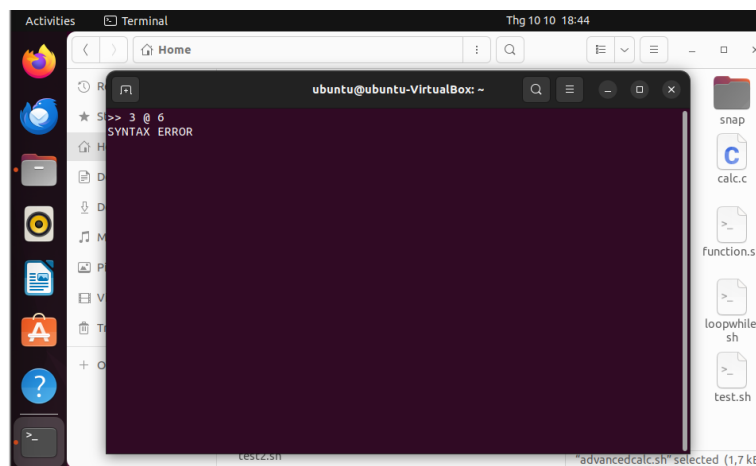
- Tiếp theo em nhập `ANS + 2` lúc này `ANS` là 12 + thêm 2 là 14.



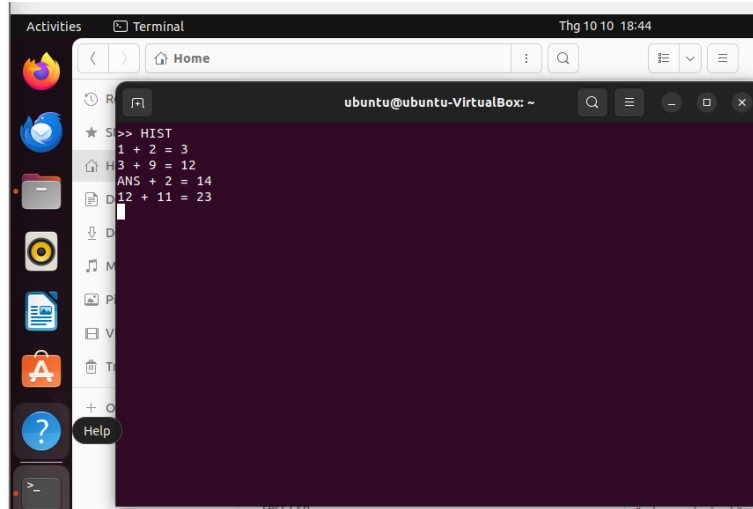
- Tiếp theo em nhập $12 + 11$ hiện kết quả là 23.



- Sau đó em nhập `3 @ 6` lúc này kiểm tra báo lỗi và in ra là SYNTAX ERROR.

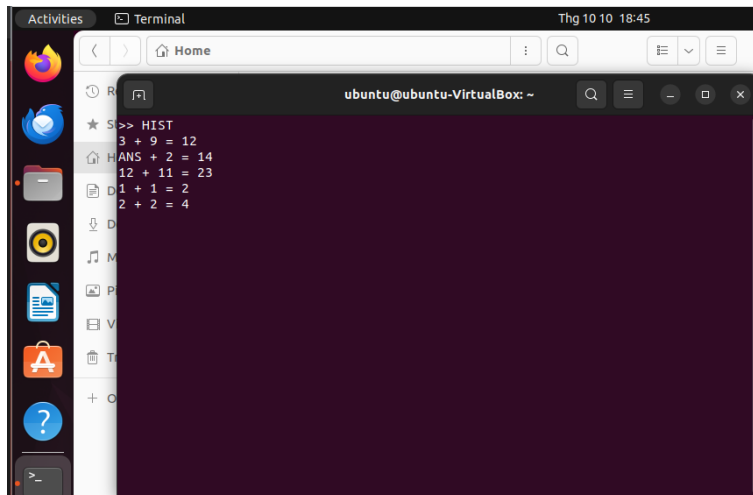


- Tiếp theo thực hiện lệnh HIST lần đầu tiên nó sẽ hiện ra 4 biểu thức đã được tính toán thành công.



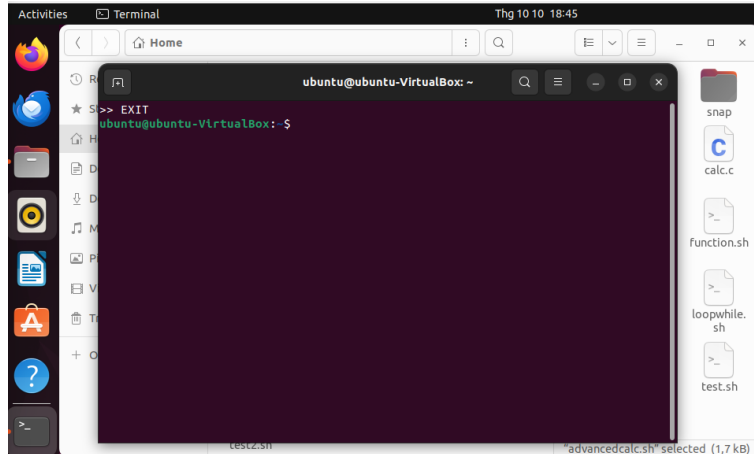
```
ubuntu@ubuntu-VirtualBox: ~  
★ S>>> HIST  
1 + 2 = 3  
3 + 9 = 12  
12 + 2 = 14  
12 + 11 = 23
```

- Tiếp đó em thực hiện 2 phép tính là $1 + 1$ và $2 + 2$ sau đó chạy lệnh HIST nó sẽ xóa đi phép tính thực hiện lúc đầu tiên là $1 + 2$ và chỉ hiện thị 5 phép tính thực thi thành công gần nhất.



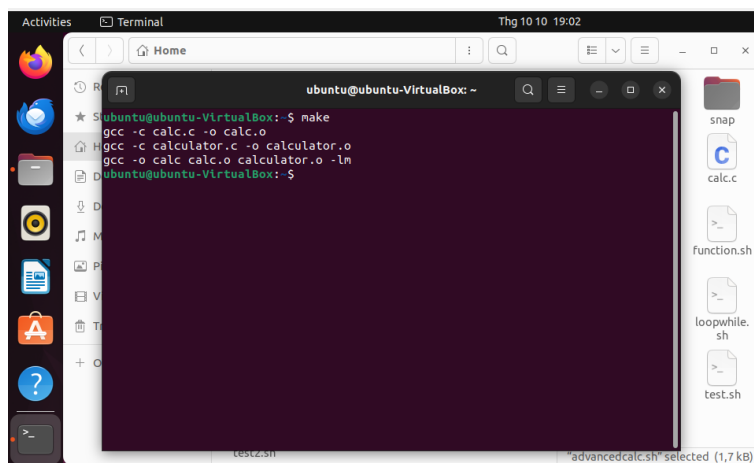
```
ubuntu@ubuntu-VirtualBox: ~  
★ S>>> HIST  
3 + 9 = 12  
12 + 2 = 14  
12 + 11 = 23  
1 + 1 = 2  
2 + 2 = 4
```

- Cuối cùng khi nhập EXIT ta thoát khỏi chương trình.

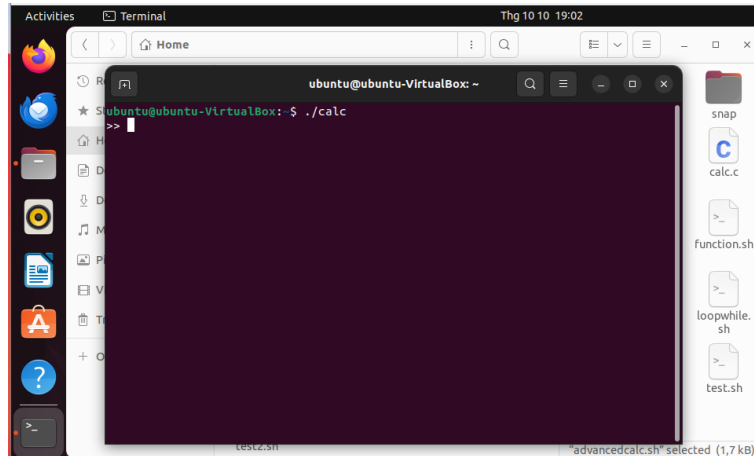


2.2 Kết quả chạy bài Exercise 5.3

- Chạy **make** để thực hiện biên dịch và liên kết dự án trên quy tắc định nghĩa trong Makefile.



- Chạy chương trình, sau đó thao tác như Exercise 3.6.



- Chạy **make clean** để xóa các tệp đối tượng và tệp thực thi đã tạo ra.

