

Báo Cáo Tuần 1

1. Các commandline trong Ubuntu, các trình soạn thảo trên terminal.

1.1. Các commandline.

a. Lệnh quản lý hệ thống và tập tin.

Commands		Description
<i>ls [options] [folder]</i>		Liệt kê dữ liệu có trong một thư mục hoặc thư mục hiện thời
<i>cd path</i>		Di chuyển đến thư mục theo đường dẫn path
<i>pwd</i>		Trả về thư mục hiện thời.
<i>touch [options] [file name]</i>		Tạo một tập tin mới theo dạng clear text
<i>lệnh vi nano</i>	<i>i</i>	Chuyển từ chế độ đọc sang ghi (Insert)
	<i>"/info"/</i>	Cú pháp để tìm ký tự info
	<i>“:q” “:q!” “:wq”</i>	<ul style="list-style-type: none"> - “:q” dùng để lưu file - “:q!” dùng để lưu file nhưng ko muốn lưu thay đổi - “:wq” dùng để lưu các thay đổi vài file. *Đối với file ở chế độ “Read only” chỉ lệnh “:wq!” được dùng để thực hiện thay đổi cho file
<i>mkdir [options] [folder name]</i>		Tạo một thư mục ngay tại thư mục hiện thời.
<i>chmod [options] [xxxx] [file or folder]</i>	<i>CHMOD = 644</i> <i>Mean:</i> <i>User: 6 = 4+2+0 ->rw</i> <i>Group: 4 = 4+0+0 ->r</i> <i>Other: 4 = 4+0+0 ->r</i>	chmod là thao tác thay đổi các quyền sau: Read (r) - 4 Write (w) - 2 Execute (x) - 1
<i>chmod [tham chiếu] [toán tử] [chế độ] [tệp tin hoặc thư mục] (*)</i>		<ul style="list-style-type: none"> - Tham chiếu dùng để xá định người dùng nào có quyền truy cập - Toán tử được dùng để quy định cách thay đổi của tệp tin. - Chế độ quy định các thao tác trên tệp tin.
<i>rm [options] [file or folder]</i>		Dùng để xóa dữ liệu
<i>cp [options] [link] [link]</i>		Lệnh copy dữ liệu
<i>mv [options] [link] [link]</i>		Lệnh di chuyển thư mục tập tin
<i>echo [value]</i> <i>echo [value] > [file]</i> <i>echo [value] >> [file]</i>		<ul style="list-style-type: none"> - echo [value] trả về một giá trị ra màn hình. - echo [value] > [file] : trả giá trị vào một tệp tin. - echo [value] >> [file]: trả giá trị vào một tệp tin ở một đường dẫn path nhất định
<i>cat [options] [file]</i> <i>cat [options] [file] > [file]</i> <i>cat [options] [file] >> [file]</i>		<ul style="list-style-type: none"> - cat [options] [file]: in ra màn hình dữ liệu từ một tệp tin - cat [options] [file] > [file]: in dữ liệu từ một tệp tin này đến một tệp tin khác - cat [options] [file] >> [file] : In dữ liệu

	từ tệp tin gốc đến tệp đích theo đường dẫn path.
--	--

(*) Chi tiết các mục tham chiếu, toán tử, chế độ.

(*).1. Tham chiếu dùng để xác định những người dùng nào được phép điều chỉnh quyền truy cập.

Tham chiếu	Nhóm	Mô tả
<i>u</i>	Người dùng	Chủ sở hữu tệp tin
<i>g</i>	Nhóm	Các người dùng thuộc nhóm của tệp tin
<i>o</i>	Khác	Các người dùng không thuộc nhóm cũng không phải chủ sở hữu.
<i>a</i>	Tất cả	Tất cả 3 lớp trên, tương đương với ugo (user-group-other)

(*).2. Toán tử quy định cách thay đổi chế độ của tệp tin.

Toán tử	Mô tả
+	Thêm chế độ sau đó vào tệp tin
-	Xóa chế độ theo sau trong tệp tin (nếu có)
=	Quy định chính xác chế độ cho tệp tin.

(*).3. Chế độ: Mỗi ký tự biểu diễn mỗi chế độ trên tệp tin.

Chế độ	Tên	Mô tả
<i>r</i>	read	Đọc tệp tin/ liệt kê các tệp tin có trong một thư mục.
<i>w</i>	write	Ghi tệp tin/tạo tệp tin mới trong một thư mục.
<i>x</i>	execute	Thực thi tệp tin/ xem một cây thư mục
<i>X</i>	special execute	<ul style="list-style-type: none"> - Có thể sử dụng thay cho x. - X đặt quyền thực thi cho thư mục bắt chấp các quyền hiện có và đặt quyền thực thi có tệp tin đã có ít nhất 1 quyền thực thi đã thiết lập (chủ sở hữu, nhóm, khác). Lệnh chỉ hữu dụng khi sử dụng kèm với toán tử '+' và thường sử dụng cùng với tùy chọn -R để đặt quyền truy cập cho nhóm hay "người dùng khác" trên một thư mục lớn mà không cần thiết lập quyền thực thi trên các tệp tin thông thường
<i>s</i>	Setuid ((u+s) - 4) /gid ((g+s) - 2)	setuid (Set User ID) và setgid (Set Group ID) được áp dụng cho các tệp tin thực thi và thư mục. Chúng cho phép người dùng thực thi tệp tin hoặc truy cập thư mục với quyền của chủ sở hữu tệp tin hoặc nhóm sở hữu, thay vì quyền của người dùng thực thi lệnh.

<i>t</i>	Sticky bit ((+t) - 1)	Chế độ t (sticky bit) được áp dụng cho các thư mục. Khi chế độ này được thiết lập, nó giới hạn khả năng xóa hoặc đổi tên các tệp tin trong thư mục chỉ cho chủ sở hữu của tệp tin, chủ sở hữu của thư mục, hoặc người dùng có quyền root
----------	-----------------------	--

(*).4. Các ví dụ thực tế lệnh *chmod* và giải thích.

Command	Description
<i>chmod a+r file</i>	Thêm chế độ read cho tất cả người dùng
<i>chmod +rwx file</i>	Cú pháp này sẽ chuyển tệp về chế độ 755(rwx - rx - rx)
<i>chmod u=rw, go=file</i>	Chuyển chính xác quyền read và write cho người sở hữu (u). Tất cả người dùng khác bị vô hiệu hóa quyền thiết lập.
<i>chmod -R u+w, go-w directory</i>	Thêm chế độ write cho user cho tất cả tệp tin và thư mục trong thư mục. Đồng thời xóa quyền write với những người còn lại.
<i>chmod 777 file</i>	Cấp tất cả quyền cho tất cả người dùng
<i>chmod 664 file</i>	Cấp quyền cho người dùng theo thứ tự (rw-r-r)
<i>chmod 0755 file</i>	Tương tự như 755. Số 0 được coi như không có chế độ đặc biệt.
<i>chmod 2755</i>	Đặt chế độ đặc biệt cho nhóm người dùng là 2 - set gid và quyền 755 trên tệp tin
<i>chmod -R u+rwX, g-rwx, o-rwx directory</i>	Thêm quyền rwx cho user, xóa tất cả quyền của nhóm và người dùng khác.

b. Các command về quản lý user:

- User có username và password. Có 2 loại user là *super user* và *regular user*.
- Mỗi User có một định danh riêng gọi là UID.

Command	Description
<i>\$useradd [option] <username></i>	Tạo user
<i>\$usermod [option] <username></i>	Thay đổi thông tin cá nhân
<i>\$userdel [option] <username></i>	Xóa người dùng.

c. Các lệnh về Network.

Command	Description
<i>\$ifconfig -a</i>	Dùng để xác định địa chỉ IP và các network interface.
<i>\$cat/proc/ interrupts</i>	Dùng để xem các thiết bị được kết nối vào computer (từ IRQ 1 - IRQ 15)
<i>\$ifconfig eth0 192.168.1.5 netmask 255.255.255.0 up</i>	Dùng để thiết lập địa chỉ IP cho một card mạng.

d. Lệnh man

- Là lệnh để hỗ trợ tìm thông tin trợ giúp về một câu lệnh, hàm chức năng, xem giải nghĩa của một file cấu hình trong Linux.
- Lệnh này lấy thông tin từ Manual Page để hiển thị các thông tin cần thiết cho người dùng.
- \$man man.

1.2. Các trình soạn thảo trên Terminal.

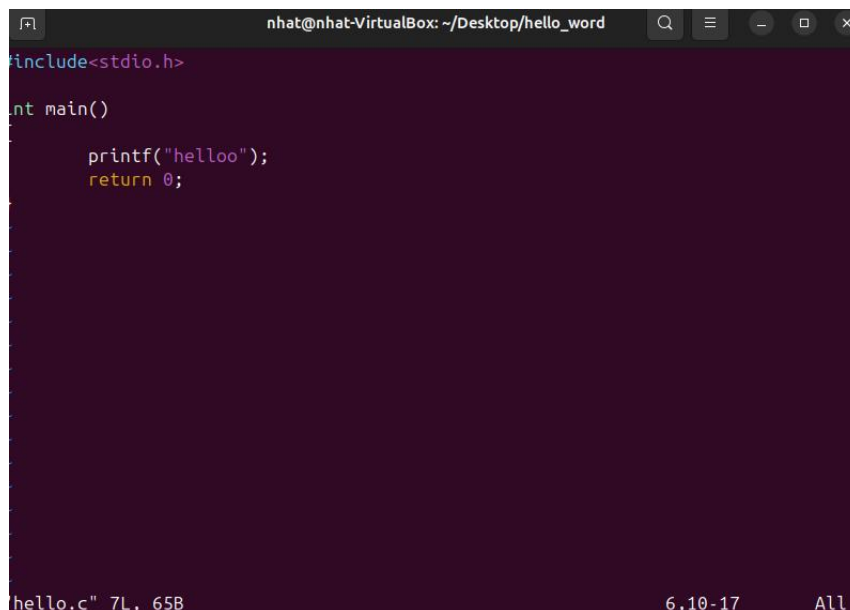
1.2.1. Vim.

a. Vim (Visual Improved).

- Cài đặt vim theo lệnh : *apt-get install vim*
(Có thể dùng sudo nếu ko có quyền root)

```
nhat@nhat-VirtualBox:/$ sudo apt-get install vim
[sudo] password for nhat:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libsodium23 vim-common vim-runtime vim-tiny xxd
Suggested packages:
  ctags vim-doc vim-scripts indent
The following NEW packages will be installed:
  libsodium23 vim vim-runtime
The following packages will be upgraded:
  vim-common vim-tiny xxd
3 upgraded, 3 newly installed, 0 to remove and 59 not upgraded.
Need to get 10.6 MB of archives.
After this operation, 42.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://vn.archive.ubuntu.com/ubuntu noble-updates/main amd64 vim-tiny amd64 2:9.1.0016-1ubuntu7.1 [803 kB]
Get:2 http://vn.archive.ubuntu.com/ubuntu noble-updates/main amd64 vim-common all 2:9.1.0016-1ubuntu7.1 [385 kB]
Get:3 http://vn.archive.ubuntu.com/ubuntu noble-updates/main amd64 xxd amd64 2:9
```

(*) Ví dụ về chương trình được viết trong vim:



```

#include<stdio.h>

int main()
{
    printf("hello");
    return 0;
}

hello.c" 7L, 65B 6,10-17 All
```

- Cài đặt trình biên dịch gcc: *apt-get install build-essential*

```

nhat@nhat-VirtualBox:~/Desktop/hello_word$ sudo apt-get install build-essential
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu bzip2 dpkg-dev fakeroot
  g++ g++-13 g++-13-x86-64-linux-gnu g++-x86-64-linux-gnu gcc gcc-13
  gcc-13-x86-64-linux-gnu gcc-x86-64-linux-gnu libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan8 libbinutils
  libbcc1-0 libctf-nobfd0 libctf0 libdpkg-perl libfakeroot
  libfile-fcntllock-perl libgcc-13-dev libgprofng0 libhwasan0 libitm1 liblsan0
  libquadmath0 libsframe1 libstdc++-13-dev libtsan2 libubsan1
  lto-disabled-list make
Suggested packages:
  binutils-doc gprofng-gui bzip2-doc debian-keyring g++-multilib
  g++-13-multilib gcc-13-doc gcc-multilib autoconf automake libtool flex bison
  gcc-doc gcc-13-multilib gcc-13-locales gdb-x86-64-linux-gnu git bzip2
  libstdc++-13-doc make-doc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential bzip2
  dpkg-dev fakeroot g++ g++-13 g++-13-x86-64-linux-gnu g++-x86-64-linux-gnu

```

(*) Biên dịch chương trình trên:

```

nhat@nhat-VirtualBox:~/Desktop/hello_word$ gcc hello.c -o hello
nhat@nhat-VirtualBox:~/Desktop/hello_word$ ./hello
helloo
nhat@nhat-VirtualBox:~/Desktop/hello_word$

```

b. Makefile.

- make đã được cài đặt tự động chung với gói build-essential khi cài đặt gcc.
- Nếu thường xuyên chèn tệp vào project hay cần thay đổi các tham số biên dịch, make được sử dụng để tự động hóa các thao tác có tính lặp đi lặp lại.
- Mặc định, Makefile sẽ được thực thi trong thư mục hiện hành khi gọi make.

(*) Make file và thực thi project

Sử dụng một chương trình khác để minh họa về Makefile và thực thi project.

+ Cấu trúc project:

```

nhat@nhat-VirtualBox:~/Desktop/hello_word$ ls
main.c  makefile  sum  sum.c  sum.h
nhat@nhat-VirtualBox:~/Desktop/hello_word$

```

+ thư viện sum

```

#ifndef __SUM_H
#define __SUM_H

int sum(int a, int b);
void hello();

#endif

```

```
#include <stdio.h>
#include "sum.h"

int sum (int a, int b) {return a+b;}
void hello()
{
    printf("hellooo \n");
}
```

+ Hàm main():

```
#include "sum.h"
#include <stdio.h>

int main()
{
    int a = 10; int b = 20;
    printf("a + b = %d \n", sum(a,b));
    hello();
    return 0;
}
```

+ Kết quả thực thi chương trình:

```
nhat@nhat-VirtualBox: ~/Desktop/hello_word$ make
gcc main.c sum.c -o sum
./sum
a + b = 30
hellooo
nhat@nhat-VirtualBox: ~/Desktop/hello_word$
```

c. Debug

- Sử dụng gdb để tiến hành gỡ lỗi. \$gdb file.
- Tạo một breakpoint trong chương trình tại dòng cần dừng hoặc 1 hàm cú pháp:

break <file> <line> || break <func_name>.

VD:

```
nhat@nhat-VirtualBox: ~/Desktop/hello_word$ gdb main.c
GNU gdb (Ubuntu 15.0.50.20240403-0ubuntu1) 15.0.50.20240403-git
Copyright (C) 2024 Free Software Foundation, Inc.
```

- Sau khi đặt breakpoint, trong lúc thực thi chương trình gdb sẽ dừng lại ở breakpoint để sửa lỗi. Để tiếp tục chương trình, chạy lệnh *run*
- Có 4 loại thao tác phổ biến trong gdb có thể sử dụng khi dừng tại breakpoint:

- + c hoặc continue: gdb sẽ tiếp tục thực thi cho tới breakpoint tiếp theo.
- + n hoặc next: gdb sẽ thực thi lệnh tiếp theo như một lệnh duy nhất.

- + s hoặc step: tương tự như next, nhưng thay vì thực thi dòng tiếp theo như một dòng duy nhất thì gdb sẽ xem như vào mã nguồn của một function và thực hiện từng dòng.
- + l hoặc layout: gdb sẽ hiển thị mã nguồn xung quanh breakpoint

- Nếu không chắc chắn về bất kỳ thao tác nào, có thể sử dụng lệnh *help <command>* để chắc chắn hơn. VD: *help next*.

1.2.2. emacs.

2. BashScript trong Linux.

2.1. Tổng quan về Bash shell:

- Khi shell được sử dụng để tương tác, sẽ luôn hiện dấu "\$" khi đang chờ lệnh từ người dùng => Khi shell chạy dưới quyền root sẽ được chuyển thành "#"
- + Tương tác dưới quyền root: *sudo -i*
- + Thoát quyền root: *exit*.

```
nhat@nhat-VirtualBox:~$ sudo -i
root@nhat-VirtualBox:~# exit
logout
nhat@nhat-VirtualBox:~$
```

- Các lệnh về thao tác dữ liệu, tệp tương tự như phần commandline.

2.2. Tạo và thực thi lệnh bash.

- Quy ước đặt tên tập lệnh bash: Tập lệnh bash kết thúc bằng đuôi ".sh", tập bash vẫn có thể chạy tốt mà không cần đuôi ".sh" mở rộng.
- Thêm "shebang": Thêm đường dẫn đến bash. "#!/usr/bin/bash"
- VD về tạo tập lệnh bash và thực thi:

```
#!/usr/bin/bash
echo "To day is "
date
echo -e "\nenter the path to directory"
read the_path

echo -e "\nyour path has the flowing files and foders: "
ls $the_path
~
~
~
~
```

- + Kết quả thực thi tập lệnh bash:


```

nhat@nhat-VirtualBox:~$ vim run_all.sh
nhat@nhat-VirtualBox:~$ ./run_all.sh
To day is
Fri Jul 12 12:00:41 PM +07 2024

enter the path to directory
/home/nhat

your path has the flowing files and foders:
Desktop    Downloads  Pictures  run_all.sh  Templates
Documents  Music      Public    snap         Videos
nhat@nhat-VirtualBox:~$ vim run_all.sh
nhat@nhat-VirtualBox:~$

```

2.3. Cơ bản về tập lệnh Bash:

2.3.1. Bình luận trong tập lệnh Bash: “#”

2.3.2. Biến và kiểu dữ liệu trong Bash:

a. Gán giá trị trực tiếp.

b. Gán giá trị bằng đầu ra thu được từ chương trình con hoặc lệnh:

- Để truy cập giá trị biến: thêm dấu \$ vào biến.

c. Quy ước đặt tên biến:

1. Tên biến phải bắt đầu bằng một chữ cái hoặc dấu gạch dưới (_)
2. Tên biến có thể chứa các chữ cái, số và dấu gạch dưới (_)
3. Tên biến phân biệt chữ hoa và chữ thường.
4. Tên biến không được chứa khoảng trắng và ký tự đặc biệt.
5. Sử dụng tên mô tả phản ánh mục đích của biến.
6. Tránh sử dụng tên hàm làm tên biến.

2.3.3. I/O trong các tập lệnh Bash.

a. Thu thập đầu vào.

a.1. Đọc dữ liệu đầu vào và lưu vào biến. (Sử dụng lệnh read)

- Tập lệnh Bash đọc giá trị và lưu vào biến:

```

nhat@nhat-VirtualBox: ~/IO_bash
#!/usr/bin/bash
echo "What's your name?"
read name
echo -e "Wellcome to bash " $name
~
~
~
~
~
~

```

- Kết quả thực thi tập lệnh Bash:

```

nhat@nhat-VirtualBox:~/IO_bash$ vim main.sh
nhat@nhat-VirtualBox:~/IO_bash$ chmod u+x main.sh
nhat@nhat-VirtualBox:~/IO_bash$ ./main.sh
What's your name?
Nhat depzai
Wellcome to bash  Nhat depzai

```


a.2. Đọc dữ liệu từ một tập tin:

- Tạo file input.txt:

```
nhat@nhat-VirtualBox: ~/IO_bash
Hello
Hiii
abcdxy
~
~
~
~
~
```

- Tạo file bash để đọc dữ liệu từ input.txt.

```
nhat@nhat-VirtualBox: ~/IO_bash
#!/usr/bin/bash
echo "What's your name?"
read name
echo -e "Wellcome to bash " $name

while read line
do
    echo $line
done < input.txt
~
~
~
```

- Kết quả thực thi tập lệnh bash.

```
nhat@nhat-VirtualBox:~/IO_bash$ vim main.sh
nhat@nhat-VirtualBox:~/IO_bash$ vim input.txt
nhat@nhat-VirtualBox:~/IO_bash$ vim main.sh
nhat@nhat-VirtualBox:~/IO_bash$ ./main.sh
What's your name?
Nhat
Wellcome to bash  Nhat
Hello
Hiii
abcdxyz
```

a.3. Command line arguments:

- Trong 1 tập lệnh hoặc một hàm bash, \$1 biểu thị đối số ban đầu được truyền, \$2, \$3, ...
- Kiểm thử:
 - + Tạo file bash:

```
#!/usr/bin/bash
echo "hello, $1"
```

+ Kết quả:

```
nhat@nhat-VirtualBox:~/IO_bash$ vim run.sh
nhat@nhat-VirtualBox:~/IO_bash$ ./run.sh Ubuntu
hello, Ubuntu
nhat@nhat-VirtualBox:~/IO_bash$
```

b. Đầu ra của tập lệnh Bash.

b.1. Hiển thị ra Terminal. (Sử dụng lệnh echo)

b.2. Ghi đè kết quả đầu ra vào tệp.

+ Lệnh Bash

```
echo "Wellcome to bash " $name > output.txt
```

+ Kết quả sau khi lệnh Bash được thực hiện

```
nhat@nhat-VirtualBox: ~/IO_bash
Wellcome to bash  Ubuntu
```

b.3. Ghi kết quả đầu ra vào cuối tệp:

+ Lệnh Bash.

```
echo "Wellcome to bash " $name > output.txt
echo "More text." >> output.txt
```

+ Kết quả sau khi lệnh Bash được thực thi.

```
nhat@nhat-VirtualBox: ~/IO_bash
Wellcome to bash  Nhat
More text.
```

b.4. Liệt kê các tệp có trong thư mục hiện hành và đưa vào tệp

+ Lệnh Bash.

```
nhat@nhat-VirtualBox:~/IO_bash$ ls > output2.txt
nhat@nhat-VirtualBox:~/IO_bash$
```

+ Kết quả thực thi lệnh

```
nhat@nhat-VirtualBox:~/IO_bash$ cat output2.txt
input.txt
main.sh
output2.txt
output.txt
run.sh
nhat@nhat-VirtualBox:~/IO_bash$
```

2.3.4. Các câu lệnh điều kiện.

- Cú pháp câu lệnh điều kiện trong bash:

```
if test condition; then
    statement
elif test condition; then
    statement
else
    default
fi
```

- Có thể sử dụng toán tử logic là AND “-a” và OR “-o” để thực hiện các phép so sánh.

- Script kiểm tra 1 số là số âm, dương hay 0:

```
nhat@nhat-VirtualBox: ~/c
#!/usr/bin/bash
echo "Enter a number: "
read num

if test $num -gt 0; then
    echo "$num is positive"
elif test $num -lt 0; then
    echo "$num is negative"
else
    echo "$num is zero"
fi
~
~
~
```

- Kết quả kiểm thử

```

nhat@nhat-VirtualBox:~/cnd_command$ ./main.sh
Enter a number:
12
12 is positive
nhat@nhat-VirtualBox:~/cnd_command$ ./main.sh
Enter a number:
-5
-5 is negative
nhat@nhat-VirtualBox:~/cnd_command$ ./main.sh
Enter a number:
0
0 is zero
nhat@nhat-VirtualBox:~/cnd_command$ vim main.sh
nhat@nhat-VirtualBox:~/cnd_command$

```

2.3.5. Lặp và phân nhánh trong Bash.

a. Vòng lặp while.

- Vòng while kiểm tra điều kiện và lặp cho đến khi điều kiện vẫn còn true.
- While Script:

```

nhat@nhat-VirtualBox:~/loop$ vim main.sh
#!/usr/bin/bash
i=1
while test $i -lt 10; do
    echo "$i"
    ((i+=1))
done
~
~

```

- Kết quả kiểm thử:

```

nhat@nhat-VirtualBox:~/loop$ vim main.sh
nhat@nhat-VirtualBox:~/loop$ ./main.sh
1
2
3
4
5
6
7
8
9

```

b. Vòng lặp for.

- For Script.

```
#!/usr/bin/bash
for i in {1..5}
do
    echo $i
done
~
~
```

- Kết quả kiểm thử.

```
nhathat@nhathat-VirtualBox:~/loop$ ./main2.sh
1
2
3
4
5
```

c. Lệnh rẽ nhánh case.

- Case Script:

```
#!/usr/bin/bash
echo "Enter a Number: "
read num

case $num in
    1)
        echo "LOW";;
    2)
        echo "MEDIUM";;
    3)
        echo "HIGH";;
    *)
        echo "OFF";;
esac
~
~
```

- Kết quả kiểm thử:

```
nhathat@nhathat-VirtualBox:~/loop$ ./main3.sh
Enter a Number:
2
MEDIUM
nhathat@nhathat-VirtualBox:~/loop$ ./main3.sh
Enter a Number:
1
LOW
nhathat@nhathat-VirtualBox:~/loop$ ./main3.sh
Enter a Number:
4
OFF
```

2.4. Cron.

- Dùng để lên lịch thực hiện các công việc có sẵn trong Linux.

- Cú pháp lên lịch cho Cron:

“ ***** *sh/path/to/script.sh* ”

* lần lượt biểu thị phút - giờ - ngày - tháng - ngày trong tuần.

- “crontab-e”: dùng để lên lịch thực hiện.
- “crontab-l”: dùng để hiển thị lịch đã lên.

```
nhat@nhat-VirtualBox:~/loop$ crontab -e
crontab: installing new crontab
nhat@nhat-VirtualBox:~/loop$ crontab -e
No modification made
nhat@nhat-VirtualBox:~/loop$ crontab -l
#Chay file main.sh vào lu nua dem moi ngay
0 0 * * * /home/nhat/loop/main.sh
#Chay file main.sh sau moi 5 phut
*/5 * * * * /home/nhat/loop/main.sh
#Chay file main.sh luc 6h sang tu thu Hai den thu Sau
0 6 * * 1-5 /home/nhat/loop/main.sh
#Chay file main.sh vào 7 ngày đầu tiên của mỗi tháng
0 0 1-7 * * /home/nhat/loop/main.sh
#Chay file main.sh ngày đầu tiên của mỗi tháng vào buổi trưa
0 12 1 * * /home/nhat/loop/main.sh
nhat@nhat-VirtualBox:~/loop$
nhat@nhat-VirtualBox:~/loop$
```

- Sử dụng lệnh “set -e” : Để thoát script bất cứ khi nào xuất hiện lỗi.
- Xem nhật ký Cron bằng cú pháp: *grep CRON /var/log/syslog*.

```
nhat@nhat-VirtualBox:~/loop$
nhat@nhat-VirtualBox:~/loop$ cd ..
nhat@nhat-VirtualBox:~$ grep CRON /var/log/syslog
2024-07-09T08:14:26.607456+00:00 nhat-VirtualBox cron[847]: (CRON) INFO (pidfile fd = 3)
2024-07-09T08:14:26.607458+00:00 nhat-VirtualBox cron[847]: (CRON) INFO (Running @reboot jobs)
2024-07-09T08:15:31.821162+00:00 nhat-VirtualBox CRON[3291]: (root) CMD (command -v debian-sa1 > /dev/null && debian-sa1 1 1)
2024-07-09T08:17:01.852966+00:00 nhat-VirtualBox CRON[3509]: (root) CMD (cd / && run-parts --report /etc/cron.hourly)
2024-07-09T08:25:01.931811+00:00 nhat-VirtualBox CRON[3633]: (root) CMD (command -v debian-sa1 > /dev/null && debian-sa1 1 1)
2024-07-09T08:30:01.958207+00:00 nhat-VirtualBox CRON[3892]: (root) CMD ([ -x /etc/init.d/anacron ] && if [ ! -d /run/systemd/system ]; then /usr/sbin/invoke-rc.d anacron start >/dev/null; fi)
2024-07-09T08:35:01.048945+00:00 nhat-VirtualBox CRON[3928]: (root) CMD (command -v debian-sa1 > /dev/null && debian-sa1 1 1)
2024-07-09T08:45:01.091882+00:00 nhat-VirtualBox CRON[4477]: (root) CMD (command -v debian-sa1 > /dev/null && debian-sa1 1 1)
2024-07-09T08:55:01.166806+00:00 nhat-VirtualBox CRON[4597]: (root) CMD (command -v debian-sa1 > /dev/null && debian-sa1 1 1)
```