

**CÂU LẠC BỘ AN TOÀN THÔNG TIN PTIT**  
**ISP CLUB**

\*\*\*\*\*



**BÁO CÁO BÀI TẬP**  
**WEEK 1: Shell script, Build code C/C++**

<b>Giảng viên hướng dẫn:</b>	<b>Mùa hè vui vẻ</b>
<b>Tên sinh viên:</b>	<b>Nguyễn Anh Vũ</b>
<b>Mã sinh viên:</b>	<b>B21DCAT224</b>

**HÀ NỘI – 2022**

# MỤC LỤC

<b>MỤC LỤC .....</b>	<b>2</b>
<b>LỜI NÓI ĐẦU.....</b>	<b>3</b>
<b>CÀI ĐẶT UBUNTU.....</b>	<b>4</b>
<b>CÀI ĐẶT PHẦN MỀM.....</b>	<b>7</b>
I. Chạy file .deb .....	6
II. Cài qua apt-get.....	6
III. Build từ source code .....	6
IV. Sử dụng Ubuntu Software .....	7
<b>TÌM HIỂU LỆNH.....</b>	<b>7</b>
I. Một số lệnh hay dùng .....	7
1. Xem thông tin OS .....	7
2. Xem cấu hình máy .....	8
3. Xem thông tin về mạng .....	10
II. Quản lý tiến trình .....	12
III. Tìm kiếm file .....	14
IV. Đặt lịch chạy định kì .....	16
V. Phân quyền .....	17
VI. Định hướng lại vào ra cơ chế pipe.....	19
<b>FILE CODE .....</b>	<b>24</b>
<b>BASH CƠ BẢN.....</b>	<b>25</b>
<b>BASH CƠ BẢN 2.....</b>	<b>36</b>
<b>CODE C.....</b>	<b>42</b>

## LỜI NÓI ĐẦU

Em không biết nói gì hết trơn nhưng đây là lần đầu tiên em sử dụng report của chị P để làm chiếc deadline đầu tiên nên có những sai sót gì mong chị P thông cảm cho em :<

Dù có phải ăn những cú đấm thì em vẫn sẽ đón nhận để trưởng thành nhưng đấm em nhẹ tay thôi các anh chị nhớ :<

Một đũa vốn mù tịt kiến thức từ lúc giao deadline đến lúc nộp deadline vẫn mù (nhưng vẫn được khai sáng) nên chắc chắn sẽ có nhiều điều cần học hỏi và em cảm ơn anh chị nhiều vì đã cho em cơ hội này ạ hehe :>



## CÀI ĐẶT UBUNTU

### I. Chức năng, ý nghĩa của các thư mục dưới thư mục root (/):

```
vwx@VuxPC:/$ ls
bin    dev    lib    libx32  mnt    root    snap    sys    var
boot   etc    lib32  lost+found  opt    run    srv     tmp
cdrom  home   lib64  media    proc   sbin    swapfile  usr
vwx@VuxPC:/$
```

1. /bin (user binaries): tập tin thực thi của **người dùng**
  - Chứa tập tin thực thi
  - Chứa các lệnh thường dùng của **người dùng đơn** (ps, ls, etc.)
2. /sbin (system binaries): tập tin thực thi của **hệ thống**
  - Chứa tập tin thực thi
  - Chứa các lệnh dùng bởi root user để quản trị hệ thống (reboot, ifconfig, etc.)
3. /etc (configuration files): tập tin **cấu hình**
  - Chứa các tập tin **cấu hình cần thiết** cho các chương trình
  - Chứa các đoạn mã khởi động và tắt các chương trình đơn lẻ
4. /dev (device files): tập tin **thiết bị**
  - Chứa các tập tin thiết bị (vì Linux coi các thiết bị là files, nhưng không phải actual files)
  - Chứa pseudo-devices (thiết bị ảo)
5. /proc (process information): thông tin **tiến trình**
  - Chứa thông tin về tiến trình hệ thống
6. /var (variable files): tập tin **biến đổi**
7. /tmp (temporary files): tập tin **tạm thời**
  - Chứa các tin tạm (có thể bị xóa sau khi restart hoặc bất kỳ khi nào bởi các tiện ích khác)
8. /usr (user programs): **chương trình** của **người dùng**
9. /home (home directories): **thư mục người dùng**
  - Chứa home folder của mỗi người dùng

- 10./boot (boot loader files): tập tin của **chương trình khởi động** máy
- 11./lib (system libraries): tập tin **thư viện hệ thống**
  - Chứa các thư viện cần thiết cho các binaries ở /bin và /sbin
- 12./opt (optional add-on Apps): ứng dụng **tùy chọn**
- 13./mnt (mount directory): thư mục **Mount**
- 14./media (removable devices): thiết bị **tháo lắp**
- 15./srv (service data): dữ liệu **dịch vụ**
- 16./root: **home** directory của **root user**

## II. **Các kiểu hệ thống file:**

# CÀI ĐẶT PHẦN MỀM

## I. CHẠY FILE .DEB:

- File .deb: là file Debian Software Package – gói phần mềm của Debian

```
anhvux@AnhVux-PC:~/Downloads$ sudo dpkg -i Nessus-10.2.0-ubuntu1110_amd64.deb
Selecting previously unselected package nessus.
(Reading database ... 178749 files and directories currently installed.)
Preparing to unpack Nessus-10.2.0-ubuntu1110_amd64.deb ...
Unpacking nessus (10.2.0) ...
Setting up nessus (10.2.0) ...
Unpacking Nessus Scanner Core Components...
Created symlink /etc/systemd/system/nessusd.service → /lib/systemd/system/nessusd.service.
Created symlink /etc/systemd/system/multi-user.target.wants/nessusd.service → /lib/systemd/system/nessusd.service.

- You can start Nessus Scanner by typing /bin/systemctl start nessusd.service
- Then go to https://AnhVux-PC:8834/ to configure your scanner
```

🔗 lệnh *dpkg* để cài hoặc gỡ các file .deb, tùy chọn *-i* để tải (install)

## II. CÀI QUA APT-GET:

```
anhvux@AnhVux-PC:~/Documents$ sudo apt-get install net-tools
[sudo] password for anhvux:
Reading package lists... Done
Building dependency tree
Reading state information... Done
net-tools is already the newest version (1.60+git20180626.aebd88e-1ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
```

🔗 Lệnh *apt-get* dùng để quản lý các gói, *install* để tải gói

## III. BUILD TỪ SOURCE CODE:

- Lấy source code ở một nguồn đáng tin cậy nào đó (GitHub chẳng hạn)
- Sử dụng `./configure` để **kiểm tra hệ thống** có đáp ứng đủ những yêu cầu để cài đặt gói không. Chạy xong nó sẽ **hiển thị các gói** cần thiết khác để cài đặt (nếu có). Sau cùng khi thỏa mãn các điều kiện trên, **MakeFile** được tạo ra (những file có hướng dẫn để build)
- Thực thi lệnh *make*: toàn bộ mã nguồn của gói được **biên dịch** sang dạng thực thi (bằng cách đọc các MakeFiles)
- Thực thi *make install*: chép cái file thực thi sang **đúng vị trí** trên hệ thống

# TÌM HIỂU LỆNH

## I. MỘT SỐ LỆNH HAY DÙNG:

### 1. Xem thông tin OS:

- Tên, phiên bản, kiến trúc (32bit hay 64bit):

```
anhvux@AnhVux-PC:~$ hostnamectl
Static hostname: AnhVux-PC
Icon name: computer-laptop
Chassis: laptop
Machine ID: 38265a3ec78948c788897b44ffbd138d
Boot ID: 14de10d58bce4c5d9f90916332874aea
Operating System: Ubuntu 20.04.4 LTS
Kernel: Linux 5.13.0-51-generic
Architecture: x86-64
```

🔗 *hostnamectl* dùng để **hiển thị hostname** của hệ thống và các thông tin liên quan

- Danh sách các gói phần mềm được cài đặt:

```
anhvux@AnhVux-PC:~$ apt list --installed
Listing... Done
accountsservice/focal-updates,focal-security,now 0.6.55-0ubuntu12~20.04.5 amd64 [installed,automatic]
acl/focal,now 2.2.53-6 amd64 [installed,automatic]
acpi-support/focal,now 0.143 amd64 [installed,automatic]
acpid/focal,now 1:2.0.32-1ubuntu1 amd64 [installed,automatic]
adduser/focal,focal,now 3.118ubuntu2 all [installed,automatic]
adwaita-icon-theme/focal-updates,focal-updates,now 3.36.1-2ubuntu0.20.04.2 all [installed,automatic]
alsalot/focal,now 1:3.22.9-1 amd64 [installed,automatic]
alsa-base/focal,focal,now 1.0.25+dfsg-0ubuntu5 all [installed,automatic]
alsa-topology-conf/focal,focal,now 1.2.2-1 all [installed,automatic]
alsa-ucm-conf/now 1.2.2-1ubuntu0.11 all [installed,upgradable to: 1.2.2-1ubuntu0.13]
alsa-utils/focal-updates,now 1.2.2-1ubuntu2.1 amd64 [installed,automatic]
amd64-microcode/focal,now 3.20191218.1ubuntu1 amd64 [installed,automatic]
anacron/focal,now 2.3-29 amd64 [installed,automatic]
app/focal,now 2.2.3.dfsg.1-5 amd64 [installed,automatic]
app-install-data-partner/focal,focal,now 19.04 all [installed,automatic]
apparmor/focal-updates,now 2.13.3-7ubuntu5.1 amd64 [installed,automatic]
appport-gtk/focal-updates,focal-updates,focal-security,focal-security,now 2.20.11-0ubuntu27.24 all [installed,automatic]
appport-symptoms/focal,focal,now 0.23 all [installed,automatic]
appport/focal-updates,focal-updates,focal-security,focal-security,now 2.20.11-0ubuntu27.24 all [installed,automatic]
appstream/focal,now 0.12.10-2 amd64 [installed,automatic]
apt-config-icons-hidpi/focal,focal,now 0.12.10-2 all [installed,automatic]
apt-config-icons/focal,focal,now 0.12.10-2 all [installed,automatic]
apt-utils/now 2.0.6 amd64 [installed,upgradable to: 2.0.9]
```

🔗 lệnh *apt list --installed* hiển thị **danh sách tất cả các gói** được cài bằng tệp **apt** và **.deb**

## 2. Xem cấu hình máy:

- CPU:

```
anhvux@AnhVux-PC:~$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
Address sizes:         39 bits physical, 48 bits virtual
CPU(s):                8
On-line CPU(s) list:   0-7
Thread(s) per core:    2
Core(s) per socket:    4
Socket(s):             1
NUMA node(s):         1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                126
Model name:            Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz
Stepping:              5
CPU MHz:               1200.000
CPU max MHz:           3600,0000
CPU min MHz:           400,0000
BogoMIPS:              2380.80
Virtualization:        VT-x
L1d cache:             192 KiB
L1i cache:             128 KiB
L2 cache:              2 MiB
L3 cache:              6 MiB
NUMA node0 CPU(s):    0-7
```

🔗 lệnh `lscpu` cung cấp thông tin về cấu trúc CPU

- RAM:

```
anhvux@AnhVux-PC:~$ free -m
              total        used         free       shared    buff/cache   available
Mem:          7719         3011         2499          591         2207         3856
Swap:         2047           0         2047
```

🔗 lệnh `free` dùng để hiện RAM vật lý (mem) và RAM ảo (swap), sử dụng thêm tùy chọn `-m` để hiện ra dung lượng dưới dạng MB



- Ổ cứng:

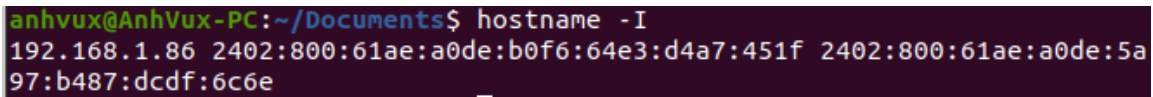
```
anhvux@AnhVux-PC:~$ df
Filesystem      1K-blocks      Used Available Use% Mounted on
udev            3918800          0    3918800   0% /dev
tmpfs           790496         1952     788544   1% /run
/dev/nvme0n1p4 104135476 12650812   86151780  13% /
tmpfs           3952472      200352     3752120   6% /dev/shm
tmpfs           5120           4         5116   1% /run/lock
tmpfs           3952472          0    3952472   0% /sys/fs/cgroup
/dev/loop0       128          128           0 100% /snap/bare/5
/dev/loop1      225792     225792           0 100% /snap/code/98
/dev/loop2       48128     48128           0 100% /snap/snapd/16010
/dev/loop3       63488     63488           0 100% /snap/core20/1518
/dev/loop4       63488     63488           0 100% /snap/core20/1328
/dev/loop5      260224     260224           0 100% /snap/gnome-3-38-2004/106
/dev/loop6       44672     44672           0 100% /snap/snapd/14978
/dev/loop7      225792     225792           0 100% /snap/code/99
/dev/loop8      116736     116736           0 100% /snap/core/13308
/dev/loop9      254848     254848           0 100% /snap/gnome-3-38-2004/99
/dev/loop10      66816     66816           0 100% /snap/gtk-common-themes/1519
/dev/loop11      83328     83328           0 100% /snap/gtk-common-themes/1534
/dev/loop12      55552     55552           0 100% /snap/snap-store/558
/dev/nvme0n1p1  251904     90532     161372   36% /boot/efi
tmpfs           790492         28     790464   1% /run/user/1000
```

⚡ lệnh *df* (disk filesystem) trả về **dung lượng của các đĩa** (đã dùng, còn bao nhiêu)



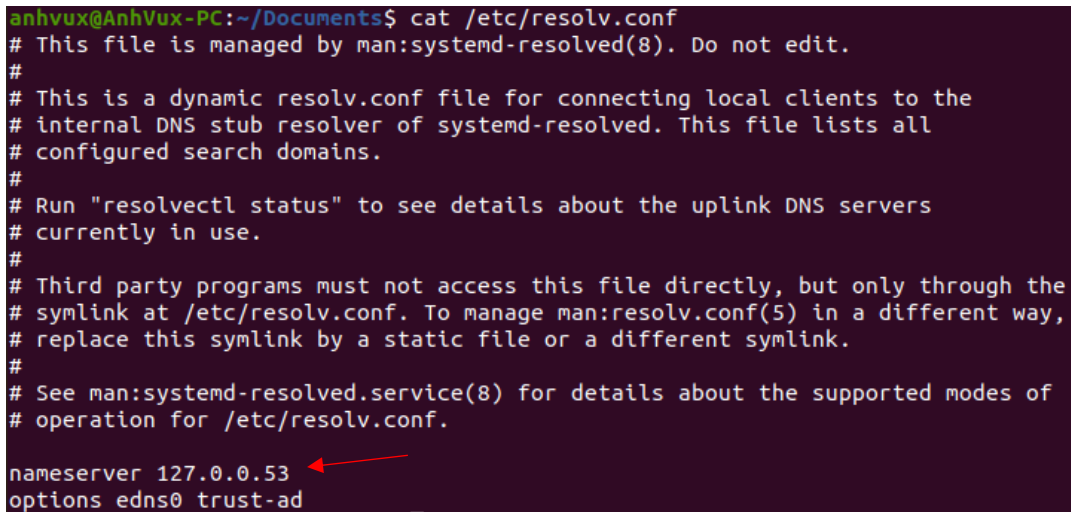
### 3. Xem thông tin về mạng:

- Địa chỉ IP, Gateway, DNS...:



```
anhvux@AnhVux-PC:~/Documents$ hostname -I
192.168.1.86 2402:800:61ae:a0de:b0f6:64e3:d4a7:451f 2402:800:61ae:a0de:5a97:b487:dcdf:6c6e
```

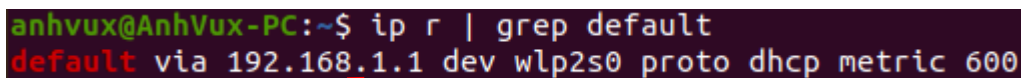
⚠ IP ở đây là **IP private**, cũng có thể dùng *ifconfig* để lấy; để lấy **IP public**, dùng *curl ifconfig.me*



```
anhvux@AnhVux-PC:~/Documents$ cat /etc/resolv.conf
# This file is managed by man:systemd-resolved(8). Do not edit.
#
# This is a dynamic resolv.conf file for connecting local clients to the
# internal DNS stub resolver of systemd-resolved. This file lists all
# configured search domains.
#
# Run "resolvectl status" to see details about the uplink DNS servers
# currently in use.
#
# Third party programs must not access this file directly, but only through the
# symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a different way,
# replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

nameserver 127.0.0.53
options edns0 trust-ad
```

⚠ Check DNS



```
anhvux@AnhVux-PC:~$ ip r | grep default
default via 192.168.1.1 dev wlp2s0 proto dhcp metric 600
```

⚠ Check Gateway

- Các kết nối đang mở và tiến trình tương ứng:

```
anhvux@AnhVux-PC:~$ netstat -atup
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN      -
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN      -
tcp        0      0 192.168.1.86:56824      server-13-224-161:https ESTABLISHED 1943/chrome --type=
tcp        0    110 192.168.1.86:43624      192.168.1.12:8009      ESTABLISHED 1943/chrome --type=
tcp        0      0 192.168.1.86:43706      tm-in-f94.1e100.n:https ESTABLISHED 1943/chrome --type=
tcp        0      0 192.168.1.86:58084      162.252.214.11:https   ESTABLISHED 1943/chrome --type=
tcp        0      0 192.168.1.86:47938      e2a.google.com:https   ESTABLISHED 1943/chrome --type=
tcp        0      0 192.168.1.86:37712      unassigned.psychz:https ESTABLISHED 1943/chrome --type=
tcp        0      0 192.168.1.86:43960      ec2-16-162-38-172:https ESTABLISHED 1943/chrome --type=
tcp        0      0 192.168.1.86:54022      ns555277.ip-54-39:https ESTABLISHED 1943/chrome --type=
tcp        0      0 192.168.1.86:42982      47.57.145.2:https      ESTABLISHED 1943/chrome --type=
tcp        0      0 192.168.1.86:35824      aeab55d76dd13c9bb:https ESTABLISHED 1943/chrome --type=
tcp        0      0 192.168.1.86:48764      ec2-16-162-194-81:https ESTABLISHED 1943/chrome --type=
tcp        0      0 192.168.1.86:54026      ns555277.ip-54-39:https ESTABLISHED 1943/chrome --type=
tcp        0      0 192.168.1.86:57120      192.168.1.12:8008      ESTABLISHED 1943/chrome --type=
tcp        0      0 192.168.1.86:39578      ns535844.ip-144-2:https ESTABLISHED 1943/chrome --type=
tcp6       0      0 ip6-localhost:ipp      [::]:*                 LISTEN      -
```

🔗 *netstat* để in ra các kết nối mạng hiện tại, các tùy chọn *-atup* để lần lượt hiện thêm socket, TCP, UDP, và process



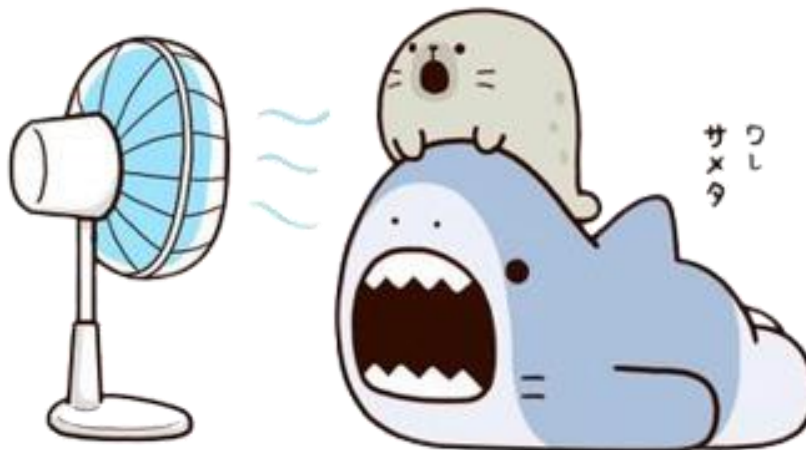
## II. QUẢN LÝ TIẾN TRÌNH:

- Xem danh sách các tiến trình đang chạy: Tên, PID, User, lệnh để chạy tiến trình:

```
top - 12:13:53 up 21 min, 1 user, load average: 1,95, 1,69, 1,11
Tasks: 272 total, 3 running, 269 sleeping, 0 stopped, 0 zombie
%Cpu(s): 5,5 us, 1,8 sy, 0,0 ni, 92,6 id, 0,0 wa, 0,0 hi, 0,1 si, 0,0 st
MiB Mem : 7719,7 total, 3889,3 free, 2032,0 used, 1798,4 buff/cache
MiB Swap: 2048,0 total, 2048,0 free, 0,0 used. 4994,2 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1974	anhvux	20	0	32,7g	216712	139408	R	18,7	2,7	3:13.59	chrome
3365	anhvux	20	0	1124,5g	185620	104292	S	13,7	2,3	1:50.51	chrome
1617	anhvux	20	0	4549040	271552	105156	R	10,3	3,4	1:55.53	gnome-shell
1487	anhvux	20	0	480440	78432	46980	S	7,7	1,0	1:17.12	Xorg
2626	anhvux	20	0	1132,6g	359692	101008	S	5,7	4,6	4:08.16	chrome
2974	anhvux	20	0	1128,6g	432848	141812	S	5,0	5,5	1:04.05	chrome
1395	anhvux	9	-11	2010392	20204	15424	S	3,3	0,3	0:25.36	pulseaudio
2606	anhvux	20	0	1128,5g	139676	94064	S	2,3	1,8	0:23.12	chrome
237	root	-51	0	0	0	0	D	1,3	0,0	0:02.94	irq/35-DELL097C
519	root	0	-20	0	0	0	I	1,0	0,0	0:06.98	kworker/u17:2-i915_flip
2335	anhvux	20	0	32,8g	74704	63236	S	1,0	0,9	0:07.02	chrome
3797	root	20	0	0	0	0	I	1,0	0,0	0:00.12	kworker/6:1-events
1929	anhvux	20	0	32,6g	294188	172352	S	0,7	3,7	1:06.99	chrome
13	root	20	0	0	0	0	I	0,3	0,0	0:02.36	rcu_sched
75	root	20	0	0	0	0	I	0,3	0,0	0:00.31	kworker/2:1-events
1976	anhvux	20	0	32,3g	107828	85792	S	0,3	1,4	0:19.43	chrome
3788	anhvux	20	0	15032	4192	3380	R	0,3	0,1	0:00.12	top
1	root	20	0	167540	11496	8292	S	0,0	0,1	0:00.90	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_par_gp
5	root	20	0	0	0	0	I	0,0	0,0	0:00.31	kworker/0:0-events

⚠ lệnh *top* dùng để liệt kê các process đang chạy của Linux; ngoài ra, còn có lệnh *ps* cũng để liệt kê (cụ thể là gửi một snapshot) về các tiến trình đang chạy



- Tắt tiến trình theo PID:

```
anhvux@AnhVux-PC:~$ kill -9 9392
anhvux@AnhVux-PC:~$ Vim: Caught deadly signal HUP
Vim: preserving files...
Vim: Finished.
Hangup
ps
  PID TTY          TIME CMD
  2033 pts/0        00:00:00 bash
  3189 pts/0        00:00:00 less
 10875 pts/0        00:00:00 crontab
 10876 pts/0        00:00:00 sh
 10877 pts/0        00:00:00 sensible-editor
 10879 pts/0        00:00:00 vim.tiny
 10957 pts/0        00:00:00 crontab
 10958 pts/0        00:00:00 sh
 10959 pts/0        00:00:00 sensible-editor
 10961 pts/0        00:00:00 vim.tiny
 12945 pts/0        00:00:00 crontab
 12946 pts/0        00:00:00 sh
 12947 pts/0        00:00:00 sensible-editor
 12949 pts/0        00:00:00 vim.tiny
 13012 pts/0        00:00:00 crontab
 13013 pts/0        00:00:00 sh
 13014 pts/0        00:00:00 sensible-editor
 13016 pts/0        00:00:00 vim.tiny
 14299 pts/0        00:00:00 ps
[2]  Killed                  crontab -e
```

⚡ lệnh `kill <PID>` sẽ **tắt tiến trình theo ID**, tùy chọn `-9` để **buộc tắt** vì một số process sẽ không tắt được 1 cách thông thường

- Tắt tiến trình theo tên:

```
anhvux@AnhVux-PC:~$ ps
  PID TTY          TIME CMD
  3585 pts/0        00:00:00 bash
  5932 pts/0        00:00:00 ps
anhvux@AnhVux-PC:~$ pkill bash
```

⚡ lệnh `pkill <process_name>` sẽ **tắt tiến trình theo tên**

### III. TÌM KIẾM FILE:

- Tìm theo tên (phân biệt / không phân biệt hoa thường):

```
anhvux@AnhVux-PC:~/Documents$ find -type f -iname hello.txt
./Hello.txt
./hello.txt
```

⚠ *tùy chọn -iname* (case-insensitive name) để **không phân biệt chữ hoa hay chữ thường** trong tên file

- Tìm theo owner / group

```
anhvux@AnhVux-PC:~/Documents$ find -user userA
./hello.txt
./D2
./D1
anhvux@AnhVux-PC:~/Documents$ find -group groupX
./hello.txt
./D2
./D1
```

- Tìm theo thời gian chỉnh sửa / truy cập cuối:

```
anhvux@AnhVux-PC:~/Documents$ find -mtime 7
anhvux@AnhVux-PC:~/Documents$ find -atime 7
```

⚠ *find -mtime <day>* để tìm **file được chỉnh sửa** trong <day> ngày gần nhất; *find -atime <day>* tương tự nhưng là **file được truy cập** gần nhất

- Tìm theo dung lượng:

```
anhvux@AnhVux-PC:~$ find -size 10M
./.config/google-chrome/Default/Extensions/amfojhdiedpdnljjbhjnhokbnohdfdb/6.6.40_0/js/home.bundle.js
./.vscode/extensions/ms-vscode.cpptools-1.10.5-linux-x64/debugAdapters/bin/System.Private.CoreLib.dll
find: './Documents/D2': Permission denied
```

⚠ *find* kèm *tùy chọn -size* để **tìm theo dung lượng**; 10M là 10MB; có thể viết +10M là những files size lớn hơn 10M, -10M thì nhỏ hơn 10M

- Tìm theo nội dung file:

```
anhvux@AnhVux-PC:~/Documents$ grep "chi P" *.txt  
hello.txt:chi P lui deadline co  
Hello.txt:chi P oi lui deadline ik :<
```

🔗 *grep* để tìm và đánh dấu chuỗi sau đó, \*.txt để quét các files .txt



#### IV. ĐẶT LỊCH CHẠY ĐỊNH KỲ:

- Chạy lệnh X định kỳ vào 0h00 mỗi ngày
- Chạy lệnh Y định kỳ vào 8h00 từ thứ hai đến thứ sáu
- Chạy lệnh Z định kỳ mỗi 3h một lần vào ngày 15 của tháng

🔗 Dùng lệnh crontab -e

```
0 0 * * * cat /Documents/hello.txt

0 8 * * 1,2,3,4,5 cat /Documents/hello.txt

0 */3 15 * * cat /Documents/hello.txt
```

\* \* \* \* \* ứng với số phút (0-59), số giờ (0-23), ngày trong tháng (1-31), tháng trong năm (1-12), thứ trong tuần (1-7); nếu để \* sẽ hiểu là "mỗi"; nếu để \*/<no.> sẽ hiểu là lặp mỗi <no.> một lần

Dòng đầu: 12:00 a.m mỗi ngày

Dòng hai: 8:00 a.m từ thứ 2-6 mỗi tuần

Dòng ba: 3 giờ / 1 lần vào ngày 15 mỗi tháng





## V. PHÂN QUYỀN:

- Tạo mới 3 user: UserA và UserB thuộc GroupX, UserC thuộc GroupY:

```
anhvux@AnhVux-PC:~$ sudo useradd userA
anhvux@AnhVux-PC:~$ sudo useradd userB
anhvux@AnhVux-PC:~$ sudo useradd userC
anhvux@AnhVux-PC:~$ sudo groupadd groupX
anhvux@AnhVux-PC:~$ sudo groupadd groupY
```

```
anhvux@AnhVux-PC:~$ sudo usermod -a -G groupX userA
anhvux@AnhVux-PC:~$ sudo usermod -a -G groupX userB
anhvux@AnhVux-PC:~$ sudo usermod -a -G groupY userC
```

🔗 `useradd <username>` để **tạo user**; `groupadd <groupname>` để **tạo group**;  
`usermod` để **điều chỉnh user**, các tùy chọn `-a` và `-G` để **add user vào nhóm**; các câu lệnh trên phải được thực hiện bởi **user root** --> dùng thêm lệnh `sudo` - thực thi lệnh bằng quyền của root

- Phân quyền file F1 chỉ cho phép thực thi bởi UserA / GroupX:

```
anhvux@AnhVux-PC:~/Documents$ sudo chown userA:groupX hello.txt
anhvux@AnhVux-PC:~/Documents$ ls -l
total 4
-rw-rw-r-- 1 userA groupX 18 Thg 6 27 00:10 hello.txt
anhvux@AnhVux-PC:~/Documents$ sudo chmod u=x,g=x hello.txt
anhvux@AnhVux-PC:~/Documents$ ls -l
total 4
---x--xr-- 1 userA groupX 18 Thg 6 27 00:10 hello.txt
```

🔗 lệnh `chown` dùng để **đổi owner (và group)** của 1 file hoặc folder (lệnh này cần thực hiện bởi root --> `sudo`); lệnh `chmod` để **điều chỉnh các mode** (lệnh này cũng cần thực hiện bởi root)

- Phân quyền thư mục D1 cho phép mọi user có quyền đọc file bên trong thư mục nhưng chỉ UserA / GroupX được tạo file mới:

```
anhvux@AnhVux-PC:~/Documents$ sudo chmod u=rw,g=rw,o=r D1
anhvux@AnhVux-PC:~/Documents$ ls
D1 D2 hello.txt
anhvux@AnhVux-PC:~/Documents$ ls -l
total 12
drw-rw-r-- 2 userA groupX 4096 Thg 6 27 01:08 D1
drwxrwxr-x 2 anhvux anhvux 4096 Thg 6 27 01:08 D2
---x--xr-- 1 userA groupX 18 Thg 6 27 00:10 hello.txt
```

- Phân quyền thư mục D2 chỉ cho phép UserA / Group X được xem danh sách file trong thư mục đó:

```
anhvux@AnhVux-PC:~/Documents$ sudo chmod u=r,g=r,o-rx D2
anhvux@AnhVux-PC:~/Documents$ ls -l
total 12
drw-rw-r-- 2 userA groupX 4096 Thg 6 27 01:08 D1
dr--r----- 2 userA groupX 4096 Thg 6 27 01:08 D2
---x--xr-- 1 userA groupX 18 Thg 6 27 00:10 hello.txt
```



## VI. ĐỊNH HƯỚNG LẠI VÀO RA CƠ CHẾ PIPE:

- Tạo một tệp có nội dung là tên và thuộc tính của các thư mục và tệp trong một thư mục bất kỳ:

```
anhvux@AnhVux-PC:~/Documents$ ls -l >> hello.txt
anhvux@AnhVux-PC:~/Documents$ cat hello.txt

total 16
drw-rw-r-- 2 userA groupX 4096 Thg 6 27 01:08 D1
dr--r--r-- 2 userA groupX 4096 Thg 6 27 01:08 D2
---x--xrw- 1 userA groupX 1 Thg 6 27 16:51 hello.txt
-rw-rw-r-- 1 anhvux anhvux 16 Thg 6 27 16:48 touch
```

- Đếm số lượng tệp và thư mục trong một thư mục:

```
anhvux@AnhVux-PC:~$ ls | wc -l
9
anhvux@AnhVux-PC:~$ ls
Desktop Documents Downloads Music Pictures Public snap Templates Videos
```

¶ `ls` kết hợp `wc` với tùy chọn `-l` để **đếm số lần xuống dòng** (ở đây, mỗi files hoặc folders được **liệt kê sẽ được xuống dòng** => đếm số files và folders cũng như đếm số lần xuống dòng)

- Đếm số lượng thư mục con trong một thư mục:

```
anhvux@AnhVux-PC:~$ find -type d | wc -l
3070
```

¶ lệnh `find` với tùy chọn `-type d` dùng để **tìm các dir trong dir** hiện tại; lệnh `wc` kèm tùy chọn `-l` để **đếm số lần xuống dòng**

- Cho một tệp văn bản, hãy in ra dòng thứ  $n$  bất kỳ của tệp và đếm số lượng từ trong dòng này:

```
anhvux@AnhVux-PC:~/Documents$ cat hello.txt
abcd efg
hahahaha
iloveu
billie eilish

ssss
huhihaho
mong chi P lui deadline cho em :<
A+ giai tich get go

thra AC 1 bai khong nhung buon ma con rat buon
anhvux@AnhVux-PC:~/Documents$ head -n 8 hello.txt | tail -n 1
mong chi P lui deadline cho em :<
anhvux@AnhVux-PC:~/Documents$ head -n 8 hello.txt | tail -n 1 | wc -w
8
```

¶ lệnh *head* và *tail* lần lượt lấy ra **phần đầu** và **phần cuối của file**, tùy chọn *-n* <*no.*> để lấy ra **no. dòng**; kết hợp 2 commands để lấy được dòng mong muốn; kết hợp cùng *wc* với tùy chọn *-w* để **đếm số từ trong dòng đó**



- Liệt kê tất cả các tệp và thư mục trong thư mục hiện hành được tạo ra trong tháng 9:

```
anhvux@AnhVux-PC:~$ ls -lt | grep 'Thg 6'
drwxr-xr-x 2 anhvux anhvux 4096 Thg 6 26 20:33 Documents
drwxr-xr-x 2 anhvux anhvux 4096 Thg 6 26 07:36 Desktop
drwxr-xr-x 2 anhvux anhvux 4096 Thg 6 22 21:58 Pictures
drwxr-xr-x 2 anhvux anhvux 4096 Thg 6 22 21:29 Downloads
drwx----- 4 anhvux anhvux 4096 Thg 6 22 21:22 snap
drwxr-xr-x 2 anhvux anhvux 4096 Thg 6 14 00:36 Music
drwxr-xr-x 2 anhvux anhvux 4096 Thg 6 14 00:36 Public
drwxr-xr-x 2 anhvux anhvux 4096 Thg 6 14 00:36 Templates
drwxr-xr-x 2 anhvux anhvux 4096 Thg 6 14 00:36 Videos
anhvux@AnhVux-PC:~$ ls -lt | grep 'Thg 9'
anhvux@AnhVux-PC:~$
```

🔗 *ls* và tùy chọn *-lt* để list tất cả files và folders theo long format và được sorted theo thứ tự mới nhất; lệnh *grep* để in các dòng đúng theo một kiểu nào đó, ở đây là các dòng chứa "Thg 6"; kết hợp 2 commands để lấy files và folders được tạo ra trong tháng 6 (không có files hay folders nào được tạo ra trong tháng 9)

- Liệt kê sự khác nhau về tên tệp trong hai thư mục bất kỳ:

```
anhvux@AnhVux-PC:~/Documents$ diff /home/anhvux/Documents/hello.txt /home/anhvux/Downloads/Hello.txt
1,3c1,2
< hello
< hello
< hello
---
> chỉ P oi lui deadline ik :<
> aloooo
5,7c4,5
< hehe
< hihi
< chỉ P lui deadline co
---
> huhueijzdc
> nvk
```

🔗 Số dòng file1 + chữ cái + số dòng file2 để thể hiện sự khác nhau giữa 2 files, < để in dòng của file1, > để in dòng file2 (chữ cái: a – add, c – change, d – delete)

- Liệt kê sự khác nhau về tên tệp giữa thư mục linux3 và thư mục linux1:

```
anhvux@AnhVux-PC:~/Documents$ diff /home/anhvux/Documents /home/anhvux/Downloads
Only in /home/anhvux/Documents: D1
Only in /home/anhvux/Documents: D2
Only in /home/anhvux/Downloads: google-chrome-stable_current_amd64.deb
Only in /home/anhvux/Documents: hello.txt
Only in /home/anhvux/Downloads: Hello.txt
Only in /home/anhvux/Downloads: Linux CLI (1).pdf
Only in /home/anhvux/Downloads: Linux CLI.pdf
Only in /home/anhvux/Downloads: Untitled document.pdf
```

- Hãy đếm tổng số tiến trình đang có trong hệ thống:

```
anhvux@AnhVux-PC:~/Documents$ ps -e | wc -l
276
```

- Đếm số lượng người sử dụng đã đăng kí với hệ thống:

```
anhvux@AnhVux-PC:~$ getent passwd | wc -l
50
```

🔗 lệnh *getent passwd* để **liệt kê** các users

- Đếm số người sử dụng đang đăng nhập vào hệ thống:

```
anhvux@AnhVux-PC:~$ who
anhvux      :0                2022-06-26 21:08 (:0)
anhvux@AnhVux-PC:~$ who | wc -l
1
```

🔗 lệnh *who* để **kiểm tra** các users hiện tại **đang đăng nhập**

- Sử dụng lệnh `env` để xem giá trị các biến môi trường:

```
anhvux@AnhVux-PC:~/Documents$ env
SHELL=/bin/bash
SESSION_MANAGER=local/AnhVux-PC:@/tmp/.ICE-unix/1611,unix/AnhVux-PC:/tmp/.ICE-unix/1611
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
LC_ADDRESS=vi_VN
GNOME_SHELL_SESSION_MODE=ubuntu
LC_NAME=vi_VN
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=ubuntu
LC_MONETARY=vi_VN
SSH_AGENT_PID=1572
GTK_MODULES=gail:atk-bridge
PWD=/home/anhvux/Documents
LOGNAME=anhvux
XDG_SESSION_DESKTOP=ubuntu
XDG_SESSION_TYPE=x11
GPG_AGENT_INFO=/run/user/1000/gnupg/S.gpg-agent:0:1
XAUTHORITY=/run/user/1000/gdm/Xauthority
WINDOWPATH=2
HOME=/home/anhvux
USERNAME=anhvux
IM_CONFIG_PHASE=1
LC_PAPER=vi_VN
LANG=en_US.UTF-8
```

🔗 lệnh `env` để liệt kê các **biến môi trường**

## FILE CODE

File bash là gì? So sánh file .bat trên Windows

- Bash: là một ngôn ngữ shell
- Bash script: một chuỗi các lệnh viết trong một file
- Là file có đuôi .sh trên Linux (và .bat trên Windows)
- Dùng để thực thi các câu lệnh bên trong nó mà không cần phải gõ hết các câu lệnh đó ra mỗi khi dùng
- So sánh với .bat:

Cài môi trường lập trình và trình bày cách biên dịch và thực thi file .c trên Linux

- Sử dụng trình biên dịch **GCC**, để tải GCC: dùng lệnh *sudo apt-get install gcc*

```
anhvux@AnhVux-PC:~/Documents/code_C$ sudo apt-get install gcc
[sudo] password for anhvux:
Reading package lists... Done
Building dependency tree
Reading state information... Done
gcc is already the newest version (4:9.3.0-1ubuntu2).
The following packages were automatically installed and are no longer required:
  libfwupdplugin1 libgnutls-dane0 libgnutls-openssl27 libpq5 libunbound8
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

- Mở trình soạn thảo văn bản **nano** để viết code
- Để biên dịch file .c sang file thực thi .obj (hoặc .o): dùng lệnh *gcc -c <filename>.c -o <filename>.o*
- Để chạy/thực thi file .o: dùng *./<filename>*



## BASH CƠ BẢN

Bài 1: Nhập 1 số n và so sánh số đó với 207. Đưa kết quả so sánh ra màn hình (2 cách: if fi và case)

Cách 1: Dùng if fi:

```
#!/bin/bash

if [ $1 -gt 207 ]
then
    echo $1 lon hon 207
elif [ $1 -lt 207 ]
then
    echo $1 nho hon 207
else
    echo $1 bang 207
fi
```

Giải thích:

- *\$1* là **tham số truyền vào** khi mở file
- *-gt* là greater than (lớn hơn), *-lt* là less than (nhỏ hơn)
- Dùng lệnh *echo* để in ra kết quả

Cách 2: Dùng case:

```
#!/bin/bash

case $1 in
    [0-9] | [1-9][0-9] | 1[0-9][0-9] | 20[1-6])
        echo $1 nho hon 207
        ;;
    207)
        echo $1 bang 207
        ;;
    *)
        echo $1 lon hon 207
        ;;
esac
```

Giải thích:

- \$1 là **tham số truyền vào** khi mở file
- Xét trường hợp \$1 nhỏ hơn 207: ở đây dùng **so sánh chuỗi**, các số bên trong [...] đại diện cho **từng chữ cái**; ký hiệu “|” hiểu là “**hoặc**”
  - + [0-9]: đại diện cho các số 0-9
  - + [1-9][0-9]: đại diện cho các số 10-99
  - + 1[0-9][0-9]: đại diện cho các số 100-199
  - + 20[1-6]: đại diện cho các số 200-206
 Nếu input nhập vào là các trường hợp trên thì sẽ *echo* ra kết quả
- Xét trường hợp \$1 bằng 207: nó là case duy nhất, input là 207 thì *echo* luôn
- Xét trường hợp \$1 lớn 207: là các trường hợp còn lại, dấu “\*” đại diện cho các input còn lại

Kết quả:

```
anhvux@AnhVux-PC:~/Documents/bash-shell-basic$ ./b1.sh 9
9 nho hon 207
anhvux@AnhVux-PC:~/Documents/bash-shell-basic$ ./b1.sh 208
208 lon hon 207
anhvux@AnhVux-PC:~/Documents/bash-shell-basic$ ./b1.sh 207
207 bang 207
```

Bài 2: Tính tổng  $S = 1+2+3+4+.....+n$ . Sử dụng vòng lặp. (3 cách: for, while và until)

Cách 1: dùng while

```
#!/bin/bash
read n
c=1
sum=0
while [ $c -le $n ]
do
    sum=$((sum+$c))
    let c++
done
echo $sum
```

Giải thích:

- Đầu tiên, đọc input từ user bằng *read*
- *c* là biến lặp, *sum* là biến lưu kết quả
- *-le* là less than or equal to (nhỏ hơn hoặc bằng)
- Điều kiện lặp: **chừng nào** *c* còn nhỏ hơn hoặc bằng *n*
- Trong vòng lặp:
  - + Ta tính biến *sum* bằng câu lệnh như trên, toán tử *((...))* (double parentheses) dùng để thực hiện phép toán bên trong nó, sau đó gán giá trị của phép toán vào *sum*
  - + Tăng biến *c* lên bằng *let* để không bị lặp vô hạn

Cách 2: Dùng until:

```
#!/bin/bash

read n
c=1
sum=0
until [ $c -gt $n ]
do
    sum=$(( $sum+$c ))
    (( c++ ))
done
echo $sum
```

Giải thích:

- Đầu tiên, đọc input từ user bằng *read*
- *c* là biến lặp, *sum* là biến lưu kết quả
- *-gt* là greater than (lớn hơn)
- Điều kiện lặp: **cho đến khi** *c* lớn hơn *n*
- Trong vòng lặp:
  - + Ta tính biến *sum* bằng câu lệnh như trên, toán tử *((...))* (double parentheses) dùng để thực hiện phép toán bên trong nó, sau đó gán giá trị của phép toán vào *sum*
  - + Tăng biến *c* lên bằng *let* để không bị lặp vô hạn

Cách 3: Dùng for:

```
#!/bin/bash

read n
sum=0
for i in $(seq $n)
do
    sum=$((sum+$i))
done
echo $sum
```

Giải thích:

- Đầu tiên, đọc input từ user bằng *read*
- *sum* là biến lưu kết quả
- Lệnh *seq \$n* dùng để **liệt kê các số từ 1 đến n**, tận dụng output của nó bằng *\$(...)* (command substitution) để lặp *i* từ số đầu (là 1) đến *n*

Kết quả:

```
anhvux@AnhVux-PC:~/Documents/bash-shell-basic$ ./b2.sh
4
10
```

(1+2+3+4=10)

Bài 3: Tạo Menu Food gồm các tùy chọn: Chicken, Pizza, Noodles, Hamburger và lựa chọn (2 cách: select item in list và mảng)

Cách 1: Dùng select:

```
#!/bin/bash

menu="Chicken Pizza Noodles Hamburger Quit"
PS3="What do you want to eat? "
select food in $menu
do
    if [ $food == 'Quit' ]
    then
        break
    fi
    echo "You have selected $food"
done
echo "Thanks for ordering"
```

Giải thích:

- Khởi tạo một cái *list* tên là *menu*
- Biến *PS3* là một biến đặc biệt, dùng để in một đoạn prompt mỗi lần lặp của *select*
- Vòng lặp của *select* sẽ kết thúc khi user chọn tùy chọn có chứa nội dung là “Quit”

Cách 2: Dùng mảng:

```
#!/bin/bash

menu=("Chicken" "Pizza" "Noodles" "hamburger" "Quit")
while true
do
    echo "What do you want to eat? "
    i=0
    while [ $i -le 4 ]
    do
        let x=$i+1
        echo "$x) ${menu[$i]}"
        let i++
    done
    read choice
    if [ $choice -eq 5 ]
    then
        echo "Thanks for ordering"
        break
    else
        let x=choice-1
        echo "You have selected ${menu[$x]}"
    fi
done
```

Giải thích:

- Khởi tạo một *mảng* tên là *menu*
- Ta lặp liên tục (*while true*) và mỗi lần lặp sẽ prompt người dùng chọn một trong các options ở menu và lưu vào biến *choice*
- Vòng *while* bên trong để in menu

Kết quả:

```
anhvux@AnhVux-PC:~/Documents/bash-shell-basic$ ./b3.sh
1) Chicken
2) Pizza
3) Noodles
4) Hamburger
5) Quit
What do you want to eat? 1
You have selected Chicken
What do you want to eat? 2
You have selected Pizza
What do you want to eat? 3
You have selected Noodles
What do you want to eat? 4
You have selected Hamburger
What do you want to eat? 5
Thanks for ordering
```

Bài 4: Nhập 1 số n và tính tổng  $S=1!+2!+3!+\dots+n!$  (sử dụng function)

```
#!/bin/bash

gt(){
    c=1
    sum=1
    while [ $c -le $1 ]
    do
        let "sum = sum * c"
        let c++
    done
    return $sum
}

read n
i=1
s=0
while [ $i -le $n ]
do
    gt $i
    let "s = $s + $?"
    let i++
done
echo $s
```

Giải thích:

- Ta có hàm *gt()* để tính giai thừa của một số \$1 tương ứng với tham số truyền vào hàm
  - + Dùng *while* để lặp cho đến khi c nhỏ hơn hoặc bằng \$1, mỗi vòng sẽ nhân sum với biến lặp
  - + Hàm sẽ return lại giá trị của sum là **exit status**
- Đến phần chính, sau khi lấy input vào n, ta lặp từ i đến n bằng while
  - + *gt \$i* tức là truyền tham số i vào hàm *gt()* bên trên để tính giai thừa của i
  - + Tính xong ta cộng dồn i vào biến s
  - + Lưu ý tăng biến lặp để không bị lặp vô hạn

Kết quả:

```
anhvux@AnhVux-PC:~/Documents/bash-shell-basic$ ./b4.sh
5
153
```

$$(1! + 2! + 3! + 4! + 5! = 153)$$



### Bài 5: Giải phương trình bậc 2 với 3 số a, b, c nhập từ bàn phím

```
#!/bin/bash

read a b c
if [ $a -eq 0 ]
then
    if [ $b -eq 0 ]
    then
        if [ $c -eq 0 ]
        then
            echo Phương trình vô số nghiệm
        else
            echo Phương trình vô nghiệm
        fi
    else
        x=$(echo "scale=2; -($c/$b)" | bc)
        echo "Phương trình có 1 nghiệm x=$x"
    fi
else
    let delta="$b*$b-4*$a*$c"
    if [ $delta -lt 0 ]
    then
        echo Phương trình vô nghiệm
    elif [ $delta -eq 0 ]
    then
        x=$(echo "scale=2; (-1)*$b/(2*$a)" | bc)
        echo "Phương trình có nghiệm kép x=$x"
    else
        x1=$(echo "scale=2; -($b+sqrt($delta))/2*$a" | bc)
        x2=$(echo "scale=2; -($b-sqrt($delta))/2*$a" | bc)
        echo "Phương trình có 2 nghiệm phân biệt x1=$x1, x2=$x2"
    fi
fi
```

Giải thích:

- Nhập 3 số a, b, c lần lượt là hệ số của phương trình bậc 2
- Xét các if elif else để giải phương trình bậc 2
- `echo "scale=2; -($c/$b)" | bc` là lệnh để in ra thương của  $-c/b$  dưới dạng thập phân với 2 chữ số sau dấu phẩy; ta pipe thêm lệnh `bc` (basic calculator) để thực hiện phép toán (nếu bỏ `bc` đi thì output của echo sẽ là một chuỗi các ký tự bên trong quotes, và  $\$<var>$  sẽ là giá trị của biến tương ứng  $\Rightarrow$  không phải là kết quả cần)

Kết quả:

```
anhvux@AnhVux-PC:~/Documents/bash-shell-basic$ ./b5.sh
0 -1 1
Phương trình có 1 nghiệm x=1.00
anhvux@AnhVux-PC:~/Documents/bash-shell-basic$ ./b5.sh
0 -1 -1
Phương trình có 1 nghiệm x=-1.00
anhvux@AnhVux-PC:~/Documents/bash-shell-basic$ ./b5.sh
1 2 3
Phương trình vô nghiệm
anhvux@AnhVux-PC:~/Documents/bash-shell-basic$ ./b5.sh
3 2 1
Phương trình vô nghiệm
anhvux@AnhVux-PC:~/Documents/bash-shell-basic$ ./b5.sh
1 3 2
Phương trình có 2 nghiệm phân biệt x1=-2.00, x2=-1.00
anhvux@AnhVux-PC:~/Documents/bash-shell-basic$ ./b5.sh
1 -3 2
Phương trình có 2 nghiệm phân biệt x1=1.00, x2=2.00
```

Bài 6: Nhập 1 số từ bàn phím, kiểm tra số đó có phải số nguyên tố hay không

```
#!/bin/bash

snt(){
    if [ $1 -lt 2 ]
    then
        return 0
    fi
    i=2
    while [ $1 -le $(echo "sqrt($1)" | bc) ]
    do
        if [ $((($1%i))) -eq 0 ]
        then
            return 0
        fi
        let i++
    done
    return 1
}

read n
snt $n
if [ $(echo $? ) -eq 0 ]
then
    echo "$n không phải số nguyên tố"
else
    echo "$n là số nguyên tố"
fi
```

Giải thích:

- Ta có hàm *snt()* để kiểm tra tham số truyền vào *\$1* có phải là số nguyên tố không
  - + Nếu số đó nhỏ hơn 2 => exit status là 0
  - + Nếu  $\geq 2$  thì đến với vòng lặp check số nguyên tố quen thuộc
- Sau khi lấy input vào *n* và truyền *n* vào hàm *snt()*, kiểm tra exit status của hàm để kết luận
- *\$?* là tham số đặc biệt, có giá trị là **exit status** của hàm được gọi gần nó nhất, ở đây *\$?* chỉ có thể là 0 hoặc 1 phụ thuộc vào giá trị trả về của hàm *snt()*

Kết quả:

```
anhvux@AnhVux-PC:~/Documents/bash-shell-basic$ ./b6.sh
5
5 là số nguyên tố
anhvux@AnhVux-PC:~/Documents/bash-shell-basic$ ./b6.sh
6
6 không phải số nguyên tố
```

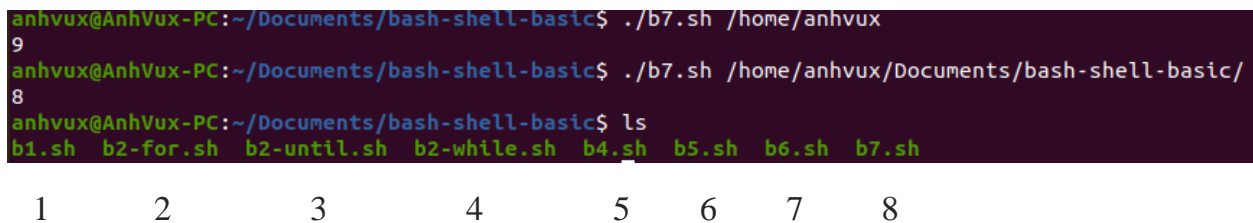
Bài 7: Tạo biến cnt = số các tệp/thư mục trong thư mục bất kỳ. In giá trị cnt ra màn hình

```
#!/bin/bash
x=$(ls $1 | wc -l)
echo $x
```

Giải thích:

- Hàm `ls $1 | wc -l` output ra **số tệp và thư mục có ở trong thư mục \$1** (bản chất `$1` là một **tham số truyền vào**, trong bài này nó sẽ là **địa chỉ, đường dẫn** đến một thư mục bất kỳ mà mình muốn đếm số tệp và thư mục con trong nó)
- Tận dụng output của hàm trên, ta lưu kết quả output vào biến x bằng **command substitution** `$(...)`
- Sau đó xuất kết quả của x ra màn hình bằng `echo`

Kết quả:



```
anhvux@AnhVux-PC:~/Documents/bash-shell-basic$ ./b7.sh /home/anhvux
9
anhvux@AnhVux-PC:~/Documents/bash-shell-basic$ ./b7.sh /home/anhvux/Documents/bash-shell-basic/
8
anhvux@AnhVux-PC:~/Documents/bash-shell-basic$ ls
b1.sh  b2-for.sh  b2-until.sh  b2-while.sh  b4.sh  b5.sh  b6.sh  b7.sh
```

1          2          3          4          5          6          7          8

Source code:

[https://drive.google.com/drive/folders/1tistm72IaEPkb6PWTs9w7WsYK9\\_6kda6?usp=sharing](https://drive.google.com/drive/folders/1tistm72IaEPkb6PWTs9w7WsYK9_6kda6?usp=sharing)

## BASH CƠ BẢN 2

Bài 1: Lấy thông tin hệ thống:

```
#!/bin/bash

echo -e "-----\tSystem Information\t-----"
echo -e "Ten may:" $(hostname)
echo -e "Ten ban phan phoi:" $(hostnamectl | grep Ubuntu | awk '{print $3, $4, $5}')
echo
echo -e "-----\tCPU/Memory Usage\t-----"
echo -e "Ten CPU:" $(lscpu | grep name | awk '{print $3, $4, $5, $6, $7, $8}')
echo -e "RAM vat ly: $(free -m | grep Mem | awk '{print $2}')MB"
echo
echo -e "-----\tDisk Usage\t\t-----"
df -m
echo
echo -e "-----\tSystem Main IP\t\t-----"
echo $(hostname -I)
echo
echo -e "-----\tUser list\t\t-----"
sort /etc/passwd | cut -d: -f 1
echo
echo -e "-----\tRoot running process\t\t-----"
ps -U root -u root u
echo -e "-----\tOpening port\t\t-----"
netstat | sort
echo -e "-----\tFolder with -w permission\t-----"
find -perm -o=w
echo
echo -e "-----\tInstalled package\t\t-----"
apt list --installed
```

## Bài 2: Xử lý file:

```
#!/bin/bash

createFile(){
    if [ ! -e "$1" ]
    then
        touch $1
    fi
}

checkTextFile(){
    check=$(file -i $1 | grep -o "text/plain")
    echo "$1" >> checketc.txt
    if [ $check = "text/plain" ]
    then
        head -n 10 "$1" >> checketc.txt
    fi
}

checkNewFile(){
    old=$1
    current=$2
    while read -r line
    do
        check=$(grep -h -w "$line" -m 1 $old)
        if [ ! -z "$check" ]
        then
            checkTextFile "$line" >> checketc.txt
        fi
    done < $current
}

checkDeleteFile(){
    old=$1
    current=$2
    while read -r line
    do
        check=$(grep -h -w "$line" -m 1 $current)
        if [ ! -z "$check" ]
        then
            echo "$line" >> checketc.txt
        fi
    done < $old
}

echo "[Log checketc - `date +%T` `date +%D`]" > checketc.txt
dir='/etc'
listOldFiles='/home/anhvux/Documents/bash-shell-basic-2/b2/old_files'
listCurrentFiles='/home/anhvux/Documents/bash-shell-basic-2/b2/current_files'
listCheckFiles='/home/anhvux/Documents/bash-shell-basic-2/b2/check_files'

createFile $listCheckFiles
createFile $listOldFiles
createFile $listCurrentFiles

find $dir -type f > $listCurrentFiles
find $dir -type f -cmin -30 > $listCheckFiles

echo "====Danh Sach file moi tao=====" >> checketc.txt
checkNewFile $listOldFiles $listCheckFiles
echo >> checketc.txt

echo "=== Danh sach file moi chinh sua ===" >> checketc.txt
filesEdit='sudo find /etc -mmin -30'

echo $filesEdit | sed 's/ /\n/g' >> checketc.txt
echo >> checketc.txt

echo "=====Danh sach file bi xoa===== " >> checketc.txt
checkDeleteFile $listOldFiles $listCurrentFiles
cat $listCurrentFiles > $listOldFiles
```

Giải thích:

- Biến *listCheckFiles* để lưu **các files bị thay đổi trong 30 phút gần nhất** (trong bài này thì lưu các files trước khi có **file mới được thêm vào**, vì khi file mới được thêm, **change time** sẽ thay đổi), *listCurrentFiles* để lưu **các files** của **phiên chạy hiện tại**, *listOldFiles* để lưu **các files** của **phiên chạy gần nhất**
- Để kiểm tra các files được tạo mới gần đây, gọi hàm tự tạo *CheckNewFile*:
  - + Hàm này truyền vào 2 tham số là *listOldFiles* và *listCheckFiles*
  - + Vòng lặp while để đọc từng dòng trong *listCheckFiles*, sau đó so sánh với các files trong *listOldFiles*; nếu trong *listCheckFiles* có tên file trong *listOldFiles*, biến *check* sẽ mang giá trị là tên của file đó (tức là biến *check* **KHÔNG rỗng**), còn nếu không có, biến *check* sẽ là biến **rỗng**
  - + Điều kiện trong *if* để kiểm tra nếu *check* rỗng hay không (-z trả về **đúng** nếu string rỗng); nếu **rỗng** (tức tồn tại file mới trong *listCheckFiles* mà *listOldFiles* không có) thì gọi hàm *CheckTextFile*; còn **KHÔNG rỗng** (tức *listCheckFiles* và *listOldFiles* đều có file này) thì đọc tiếp
  - + Hàm *CheckTextFile* để kiểm tra file được truyền vào có phải là text file hay không bằng cách chạy lệnh *file -i* để **xem format file**, sau đó pipe grep để tìm định dạng là text/plain; vì là file mới nên cứ echo vào *checketc.txt* đã, sau đó nếu nó đúng là file text thì in thêm 10 dòng đầu tiên
- Để kiểm tra các files mới được chỉnh sửa:
  - + Dùng lệnh *find /etc -mmin -30* (tìm trong /etc các file được chỉnh sửa trong vòng 30 phút gần nhất), sau đó lưu output vào *FilesEdit*
  - + *sed* là một text stream editor, trong trường hợp này dùng để **thay thế** (chữ s-substitution) các **đấu cách** thành **xuống dòng** (\n) và **thay thế tất cả** (chữ g-global replacement) trong *FilesEdit* sau đó ghi vào *checketc.txt*
- Để kiểm tra các files bị xóa, gọi hàm tự tạo *CheckDeleteFile*:
  - + Hàm này hoạt động tương tự *CheckNewFile*, chỉ khác tham số truyền vào và đọc từng dòng của file cũ rồi tìm trong file hiện tại
- Gửi mail định kỳ: \*/30 \* \* \* \* sendmail root@localhost < checketc.txt

Kết quả:

```
anhvux@AnhVux-PC:~/Documents/bash-shell-basic-2/b2$ ./checketc.sh
find: '/etc/ssmtp': Permission denied
find: '/etc/ssl/private': Permission denied
find: '/etc/cups/ssl': Permission denied
find: '/etc/polkit-1/localauthority': Permission denied
find: '/etc/ssmtp': Permission denied
find: '/etc/ssl/private': Permission denied
find: '/etc/cups/ssl': Permission denied
find: '/etc/polkit-1/localauthority': Permission denied
anhvux@AnhVux-PC:~/Documents/bash-shell-basic-2/b2$ cat checketc.txt
[Log checketc - 01:20:00 07/02/22]
====Danh Sach file moi tao=====
/etc/test.txt
alo
1234
5678
hacking_is_fun

=== Danh sach file moi chinh sua ===
/etc
/etc/test.txt

=====Danh sach file bi xoa=====
/etc/terminfo/README
/etc/timezone
/etc/udisks2/udisks2.conf
/etc/vim/vimrc.tiny
/etc/vim/vimrc
/etc/shadow-
/etc/appstream.conf
/etc/sysctl.d/10-network-security.conf
/etc/sysctl.d/10-kernel-hardening.conf
```

## Bài 3: Monitor SSH:

```

1 #!bin/bash
2
3 old='/home/anhvux/Documents/bash-shell-basic-2/b3/phientruoc.txt'
4 now='/home/anhvux/Documents/bash-shell-basic-2/b3/phienbaygio.txt'
5 touch phienbaygio.txt phientruoc.txt
6
7 who > $now
8 while read -r line
9 do
10     check=$(grep -w "$line" -m 1 $old)
11     if [ ! -z "$check" ]
12     then
13         echo -e "User" "$(echo "$line" | awk '{print $1}')" "đang
nhập thành công vào thời gian" "$(echo "$line" | awk '{print $4}')" "$(echo
"$line" | awk '{print $3}')" > mail.txt
14     fi
15 done < $now
16 cat $now > $old

```

Giải thích:

- Biến *now* và *old* dùng để **lưu các phiên đăng nhập tương ứng** (*now* để lưu phiên đăng nhập hiện tại, *old* để lưu phiên đăng nhập của lần gần nhất)
- Lệnh *who* để đưa ra output về **thông tin của các users** đang đăng nhập, sau đó chuyển hướng output vào file *now*
- **Đọc từng dòng** trong file *now* bằng vòng lặp:
 

```
while read -r line do <commands> done < $now
```

  - + Biến *line* sẽ mang giá trị là từng dòng của *old* (ở đây là **tên của từng user** trong phiên đăng nhập hiện tại)
  - + Lệnh *grep -w "\$line" -m 1 \$old* để tìm xem **các tên của user hiện tại có tồn tại** trong **phiên đăng nhập lần gần nhất** không, nếu **KHÔNG** thì biến *check* sẽ mang giá trị là **tên của user** đó, còn **CÓ** thì *check* sẽ có giá trị rỗng
  - + *-z "\$check"* để kiểm tra xem xâu rỗng không (nếu rỗng thì trả về đúng) => thêm *toán tử !* để trả về đúng nếu xâu không rỗng (tức là có user mới đăng nhập)
  - + Nếu điều kiện trên là đúng (có user mới) thì in ra theo cú pháp trên
  - + Lệnh *awk '{print \$n}'* dùng để in ra **từ thứ n** trong output khi echo giá trị của *line* (\$1 sẽ là tên user, \$3 là thời gian, \$4 là ngày) rồi ghi vào file *mail.txt*
  - + Sau đó **ghi đè** lại file *now* vào file *old*
- Gửi mail định kỳ: `*/5 * * * * sendmail root@localhost < mail.txt`



Kết quả:

```
anhvux@AnhVux-PC:~/Documents/bash-shell-basic-2/b3$ ./sshmonitor.sh
anhvux@AnhVux-PC:~/Documents/bash-shell-basic-2/b3$ ls
mail.txt  phienbaygio.txt  phientruoc.txt  sshmonitor.sh
anhvux@AnhVux-PC:~/Documents/bash-shell-basic-2/b3$ cat mail.txt
User anhvux dang nhap thanh cong vao thoi gian 01:03:13 07/02/22
anhvux@AnhVux-PC:~/Documents/bash-shell-basic-2/b3$
```

Source code: <https://drive.google.com/drive/folders/1tbPsiLhu3Ny-TV1WWz06DpSBlxM0BgVu?usp=sharing>

## CODE C

Bài 1: Nhập số tự nhiên n. Liệt kê các số Fibonacci nhỏ hơn n là số nguyên tố

```
#include <stdio.h>
#include <math.h>

int prime(int n){
    if (n<2)
        return 0;
    for (int i=2; i<=sqrt(n); ++i)
        if (n%i==0)
            return 0;
    return 1;
}

int main(){
    int n;
    int a[1000]={0};
    a[1]=1;
    scanf("%d", &n);
    for (int i=2; a[i-1]+a[i-2]<n; ++i){
        a[i]=a[i-1]+a[i-2];
        if (prime(a[i]))
            printf("%d ", a[i]);
    }
    printf("\n");
}
```

Giải thích:

- Hàm prime để kiểm tra số nguyên tố
- Lưu các số Fibo vào mảng a bằng cách khởi tạo mảng với phép tính như trên
- Số mới được thiết lập với điều kiện nhỏ hơn n, nếu là số nguyên tố thì in ra màn hình luôn

Kết quả:

```
anhvux@AnhVux-PC:~/Documents/code_C$ gcc b1.c -o b1.o -lm
anhvux@AnhVux-PC:~/Documents/code_C$ ./b1.o
1000
2 3 5 13 89 233
```

Bài 2: Nhập chuỗi s1. Thực hiện in chuỗi s2 là đảo ngược của chuỗi s1

```
#include <stdio.h>
#include <string.h>
int main()
{
    char s[100];
    char res[100];
    fgets(s, sizeof(s), stdin);
    int n = strlen(s) - 1; //do strlen() đếm cả ký tự "\n"
    for (int i=0; i<n; ++i)
        res[i]=s[n-i-1];
    puts(res);
}
```

Giải thích:

- Hàm fgets dùng để nhập chuỗi tính cả dấu cách (và cả ký tự \n hoặc enter)
- Biến n có giá trị là strlen(s) – 1 (do strlen() đếm cả ký tự \n)
- Vòng lặp để gán giá trị cho từng chỉ số đầu của res với giá trị chỉ số cuối của s

Kết quả:

```
anhvux@AnhVux-PC:~/Documents/code_C$ gcc b2.c -o b2.o
anhvux@AnhVux-PC:~/Documents/code_C$ ./b2.o
nguyen anh vu
uv hna neyugn
```