

CÔNG TY AN NINH MẠNG VIETTEL

WEB SECURITY



BÁO CÁO TUẦN 01

Họ và tên : Lê Đức Anh

Mentor : Nguyễn Tuấn Anh

Hà Nội, 01/2024

I. MỤC LỤC

I. MỤC LỤC.....	1
II. TÌM HIỂU MỘT SỐ CÂU LỆNH LINUX	2
1. Các lệnh xem cấu hình máy	2
2. Quản lí tiến trình	5
3. Tìm kiếm file	7
4. Đặt lệnh chạy định kì.....	8
5. Tạo người dùng/nhóm và phân quyền	10
III. TÌM HIỂU VỀ WEB SERVER, VIRTUAL HOST.....	14
1. Khái niệm, cách hoạt động.....	14
2. Giới thiệu Apache	15
3. Cài đặt Web Server Apache	15
4. Cấu hình Virtual Hosts cho Web Server	17
IV. TÌM HIỂU VỀ LẬP TRÌNH SHELL SCRIPT.....	22
1. Khái niệm Shell Script	22
2. Thực hành	23
V. TÌM HIỂU VỀ HTTP CLIENT.	30
1. Cấu trúc của dòng Request line.....	30
2. Các HTTP Header thường gặp.....	31
3. Các mã trả về của HTTP Response	33

II. TÌM HIỂU MỘT SỐ CÂU LỆNH LINUX

1) Các lệnh xem cấu hình máy

- **Xem thông tin OS:**

Để xem thông tin hệ điều hành, ta sử dụng “**uname**” :

```
leediay@ubuntu:~$ uname
Linux
```

- **Xem tên, phiên bản, kiến trúc:**

Ta sử dụng câu lệnh “**uname -a**” để in ra toàn bộ thông tin...

```
leediay@ubuntu:~$ uname -a
Linux ubuntu 5.15.0-88-generic #98~20.04.1-Ubuntu SMP Mon Oct 9 16:43:45 UTC 2023
x86_64 x86_64 x86_64 GNU/Linux
```

⇒ Tên server là “ubuntu”, phiên bản 5.15.0-88-generic, Ubuntu 20.04, x86_64 thể hiện máy tính có kiến trúc 64 bit.

- **Danh sách các gói phần mềm được cài đặt:**

Sử dụng lệnh “**sudo apt list --installed**” để kiểm tra:

```
leediay@ubuntu:~$ sudo apt list --installed
[sudo] password for leediay:
Listing... Done
accountsservice/focal-updates,focal-security,now 0.6.55-0ubuntu12~20.04.6 amd64 [i
n
stalled,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 [i
n
stalled,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/focal-updates,focal-updates,now 1.2.2-1ubuntu0.13 all [installed,au
t
omatic]
alsa-utils/focal-updates,now 1.2.2-1ubuntu2.1 amd64 [installed,automatic]
amd64-microcode/focal-updates,focal-security,now 3.20191218.1ubuntu1.2 amd64 [ins
t
alled,automatic]
anacron/focal,now 2.3-29 amd64 [installed,automatic]
apache2-bin/focal-updates,focal-security,now 2.4.41-4ubuntu3.15 amd64 [installed,
a
utomatic]
apache2-data/focal-updates,focal-updates,focal-security,focal-security,now 2.4.41
4
ubuntu3.15 all [installed,automatic]
```

- **Hiển thị thông tin của CPU như số core, tốc độ, dung lượng bộ nhớ cache...**

Sử dụng câu lệnh “**lscpu**” hoặc “**cat /proc/cpuinfo**”

```
leedlay@ubuntu:~$ lscpu
Architecture:                x86_64
CPU op-mode(s):              32-bit, 64-bit
Byte Order:                  Little Endian
Address sizes:               45 bits physical, 48 bits virtual
CPU(s):                      2
On-line CPU(s) list:         0,1
Thread(s) per core:          1
Core(s) per socket:          1
Socket(s):                   2
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:                     1190.401
BogoMIPS:                     2380.80
Hypervisor vendor:           VMware
Virtualization type:         full
L1d cache:                   96 KiB
L1i cache:                   64 KiB
L2 cache:                     1 MiB
L3 cache:                     12 MiB
NUMA node0 CPU(s):           0,1
```

- Để xem các thông tin của RAM, ta có thể sử dụng những câu lệnh sau:

+ *free*

+ *top*

+ *vmstat*

+ *cat /proc/meminfo*

```
leedlay@ubuntu:~$ free -h
              total        used        free      shared  buff/cache   available
Mem:           1.9Gi       1.1Gi         64Mi       1.0Mi       703Mi       595Mi
Swap:          1.4Gi          59Mi       1.3Gi

leedlay@ubuntu:~$ vmstat
procs -----memory----- --swap--  -----io----- -system--  -----cpu-----
 r  b   swpd   free   buff   cache   si   so    bi    bo    in    cs   us   sy   id   wa   st
0  0   60940   65316   15696   705464    1   33   839   433   171   286    7    7   86    0    0

leedlay@ubuntu:~$ cat /proc/meminfo
MemTotal:        1974564 kB
MemFree:         64812 kB
MemAvailable:    610292 kB
Buffers:         15704 kB
Cached:          619852 kB
SwapCached:      2844 kB
Active:          438216 kB
Inactive:        864068 kB
Active(anon):    41412 kB
Inactive(anon):  627276 kB
Active(file):    396804 kB
Inactive(file):  236792 kB
Unreclaimable:   0 kB
Mlocked:         0 kB
SwapTotal:       1435264 kB
SwapFree:        1374324 kB
Dirty:           0 kB
Writeback:       0 kB
AnonPages:       664804 kB
Mapped:          192532 kB
Shmem:           1952 kB
KReclaimable:    86196 kB
Slab:            217792 kB
SReclaimable:    86196 kB
SUnreclaim:     131596 kB
KernelStack:     11872 kB
PageTables:      15168 kB
NFS_Unstable:    0 kB
Bounce:          0 kB
WritebackTmp:    0 kB
```

- Để check các thông tin về ổ cứng, ta sử dụng lệnh “df”:

```
leedlay@ubuntu:~$ df
Filesystem      1K-blocks      Used Available Use% Mounted on
udev             946568          0    946568   0% /dev
tmpfs            197460         1788    195672   1% /run
/dev/sda5        30268356 14712380  13993096  52% /
tmpfs            987280          0    987280   0% /dev/shm
tmpfs             5120           4       5116   1% /run/lock
tmpfs            987280          0    987280   0% /sys/fs/cgroup
/dev/loop0       161280       161280          0 100% /snap/chromium/2686
/dev/loop1       161280       161280          0 100% /snap/chromium/2680
/dev/loop2       57088        57088          0 100% /snap/core18/2796
/dev/loop4        128          128          0 100% /snap/bare/5
/dev/loop5       65024        65024          0 100% /snap/core20/2015
/dev/loop6       65536        65536          0 100% /snap/core20/2105
/dev/loop7       75776        75776          0 100% /snap/core22/864
/dev/loop8       75904        75904          0 100% /snap/core22/1033
/dev/loop9       68224        68224          0 100% /snap/cups/980
/dev/loop12      358144       358144          0 100% /snap/gnome-3-38-2004/140
/dev/loop11      168832       168832          0 100% /snap/gnome-3-28-1804/198
/dev/loop13       68224        68224          0 100% /snap/cups/1024
/dev/loop10      358144       358144          0 100% /snap/gnome-3-38-2004/143
/dev/loop15      508928       508928          0 100% /snap/gnome-42-2204/141
/dev/loop14      508928       508928          0 100% /snap/gnome-42-2204/132
/dev/loop17       6528         6528          0 100% /snap/notepad-plus-plus/387
/dev/loop16       86144        86144          0 100% /snap/notion-snap-reborn/25
/dev/loop25      41856        41856          0 100% /snap/snapd/20290
/dev/loop22       7040         7040          0 100% /snap/notepad-plus-plus/389
/dev/loop18      506880       506880          0 100% /snap/wine-platform-runtime-cor
e20/95
/dev/loop26      41472        41472          0 100% /snap/snapd/20671
/dev/loop24      47104        47104          0 100% /snap/snap-store/599
/dev/loop21      93952        93952          0 100% /snap/gtk-common-themes/1535
/dev/loop27      507008       507008          0 100% /snap/wine-platform-runtime-cor
e20/88
/dev/loop23      464000       464000          0 100% /snap/wine-platform-7-devel-cor
e20/24
/dev/loop20      12672        12672          0 100% /snap/snap-store/959
/dev/sda1        523248         4       523244   1% /boot/efi
/dev/loop28      57088        57088          0 100% /snap/core18/2812
```

- Địa chỉ IP, Gateway, DNS...

Sử dụng lệnh “ifconfig” để check địa chỉ:

```
leedlay@ubuntu:~$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
    ether 02:42:6d:0d:3c:f3 txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.23.129 netmask 255.255.255.0 broadcast 192.168.23.255
    inet6 fe80::a572:48d:7a6c:d054 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:50:e6:88 txqueuelen 1000 (Ethernet)
    RX packets 68682 bytes 102540265 (102.5 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 5916 bytes 440939 (440.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 333 bytes 31354 (31.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 333 bytes 31354 (31.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

⇒ Địa chỉ IP của máy: 192.168.23.129

- Sử dụng “`resolvectl status | grep Current`” để check DNS:

```
leediay@ubuntu:~$ resolvectl status | grep Current
Current Scopes: none
Current Scopes: DNS
Current DNS Server: 192.168.23.2
```

- Các kết nối đang mở và tiến trình tương ứng

Câu lệnh “`top`” và “`ps`” dùng để hiển thị các tiến trình đang chạy trên Ubuntu:

```
top - 19:15:25 up 48 min, 1 user, load average: 0.00, 0.02, 0.05
Tasks: 306 total, 1 running, 305 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.7 us, 0.8 sy, 0.0 ni, 98.3 id, 0.0 wa, 0.0 hi, 0.2 si, 0.0 st
MiB Mem : 1928.3 total, 86.0 free, 1160.1 used, 682.2 buff/cache
MiB Swap: 1401.6 total, 1332.8 free, 68.8 used, 596.2 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
4960	leediay	20	0	12012	3912	3112	R	1.0	0.2	0:00.05	top
1228	root	20	0	1355700	20552	5028	S	0.7	1.0	0:01.90	contain+
3198	leediay	20	0	289496	34948	13396	S	0.7	1.8	0:05.86	Xorg
996	root	20	0	239460	6180	5240	S	0.3	0.3	0:05.95	vmtoolsd
3367	leediay	20	0	3749988	193700	57096	S	0.3	9.8	0:23.42	gnome-s+
3544	leediay	20	0	144664	33188	21240	S	0.3	1.7	0:05.26	vmtoolsd
4857	leediay	20	0	814496	49412	36704	S	0.3	2.5	0:01.80	gnome-t+
1	root	20	0	169960	12720	7744	S	0.0	0.6	0:07.48	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.05	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+
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	slub_fl+
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker+
10	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_perc+
11	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tas+
12	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tas+

```
leediay@ubuntu:~$ ps
PID TTY          TIME CMD
 4866 pts/0        00:00:00 bash
 4963 pts/0        00:00:00 ps
```

PID – Process ID: Mỗi Process có 5 ký tự số. Những số này có thể hết (hết số) và bắt đầu lại, nhưng tại bất kỳ thời điểm nào, không có hơn 1 PID trong hệ thống.

2) Quản lý tiến trình

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

Như ở trên, để xem danh sách các tiến trình đang chạy, ngoài lệnh “`top`” ta còn có thể dùng lệnh “`htop`” để có 1 cái nhìn trực quan, sinh động và dễ hiểu hơn:

1 [||| 2.0%] Tasks: 130, 307 thr; 1 running
 2 [||| 0.7%] Load average: 0.07 0.02 0.04
 Mem[||||| 1.13G/1.88G] Uptime: 00:51:30
 Swp[||| 68.8M/1.37G]

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
4970	leediay	20	0	10832	4168	3232	R	2.0	0.2	0:00.14	htop
3367	leediay	20	0	3662M	189M	57096	S	1.3	9.8	0:24.05	/usr/bin/gnome-
4857	leediay	20	0	795M	49412	36704	S	0.7	2.5	0:02.04	/usr/libexec/gn
996	root	20	0	233M	6180	5240	S	0.7	0.3	0:06.23	/usr/bin/vmtool
3198	leediay	20	0	282M	34948	13396	S	0.0	1.8	0:06.27	/usr/lib/xorg/X
1676	root	20	0	1268M	24856	8288	S	0.0	1.3	0:00.14	/usr/bin/docker
3544	leediay	20	0	141M	33448	21240	S	0.0	1.7	0:05.64	/usr/bin/vmtool
1	root	20	0	165M	12720	7744	S	0.0	0.6	0:07.48	/sbin/init auto
393	root	19	-1	69084	15140	13480	S	0.0	0.8	0:01.90	/lib/systemd/sy
457	root	20	0	24852	5948	3448	S	0.0	0.3	0:00.75	/lib/systemd/sy
466	root	20	0	147M	268	16	S	0.0	0.0	0:00.00	vmware-vmblock-
467	root	20	0	147M	268	16	S	0.0	0.0	0:00.00	vmware-vmblock-
464	root	20	0	147M	268	16	S	0.0	0.0	0:00.00	vmware-vmblock-
978	systemd-r	20	0	24808	12336	8684	S	0.0	0.6	0:00.40	/lib/systemd/sy
1005	systemd-t	20	0	90908	5804	5088	S	0.0	0.3	0:00.00	/lib/systemd/sy
979	systemd-t	20	0	90908	5804	5088	S	0.0	0.3	0:00.22	/lib/systemd/sy
994	root	20	0	50148	8884	7628	S	0.0	0.4	0:00.09	/usr/bin/VGAuth
1003	root	20	0	233M	6180	5240	S	0.0	0.3	0:00.22	/usr/bin/vmtool
1004	root	20	0	233M	6180	5240	S	0.0	0.3	0:00.00	/usr/bin/vmtool
1036	root	20	0	233M	6180	5240	S	0.0	0.3	0:00.00	/usr/bin/vmtool
1121	root	20	0	233M	7460	6424	S	0.0	0.4	0:00.16	/usr/lib/connm

→ “htop” thể hiện chi tiết hơn nhiều so với “top”

- Tắt tiến trình

Để tắt 1 tiến trình đang chạy trong Ubuntu, ta có thể sử dụng các lệnh như “kill”, “killall” và “pkill”. Sự khác biệt chính giữa các công cụ này là **kill** chấm dứt các tiến trình **dựa trên số ID** của tiến trình (PID), trong khi các lệnh **killall** và **pkill** chấm dứt các tiến trình đang chạy **dựa trên tên** của chúng và các thuộc tính khác.

Tắt tiến trình theo PID

Để tắt 1 tiến trình đang hoạt động, ta sẽ dùng lệnh “kill + PID”

```
leediay@leediay:~$ kill 1919
leediay@leediay:~$
```

Tắt tiến trình theo tên

Để xem PID của tên 1 chương trình, ta sử dụng “pidof + tên_tiến_trình”

```
leediay@leediay:~$ pidof firefox
9763 9759 9740 9738 9682 9638 9621 9576
```

Lần này để tắt 1 tiến trình, ta sẽ không sử dụng PID của nó nữa, mà sử dụng bằng chính tên của tiến trình. Để làm như vậy, ta dùng lệnh “killall + tên_tiến_trình”

```
leediay@leediay:~$ killall firefox
```

Để xóa hết tất cả các tiến trình thuộc về chương trình đó, ta còn có thể sử dụng tham số “-9” trong câu

lệnh “killall”

3) Tìm kiếm file

- Tìm kiếm file theo tên

Để có thể tìm kiếm 1 file theo tên, có rất nhiều câu lệnh, cụ thể như:

+ *locate*

+ *which*

+ *whereis*

+ *find*

Lệnh “**which**” và “**whereis**” khác nhau ở chỗ “**whereis**” cho biết vị trí các trang nhị phân, nguồn và trang man của 1 lệnh, còn lệnh “**which**” chỉ hiển thị vị trí nhị phân của lệnh.

```
leediay@leediay:~$ which firefox
/usr/bin/firefox
leediay@leediay:~$ whereis firefox
firefox: /usr/bin/firefox /usr/lib/firefox /etc/firefox /usr/share/man/man1/firefox.1.gz
leediay@leediay:~$
```

- Tìm theo owner / group

+ Tìm file với user

Tìm file có tên là “ducanh.txt” thuộc sở hữu của user leediay trong thư mục /home:

“find / -user leediay -name ducanh.txt”

+ Tìm nhiều file với user

Tìm tất cả file thuộc sở hữu của user leediay trong thư mục /home:

“find /home -user leediay”

+ Tìm nhiều file với group

Tìm tất cả các file thuộc sở hữu của group “ducanh” nằm trong thư mục /home:

“find /home -group ducanh”

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

+ Tìm file được chỉnh sửa thời gian gần nhất

Tìm tất cả các file được chỉnh sửa trong thời gian 7 ngày gần nhất.

“find /home -mtime 7”

+ Tìm tất cả các file được truy cập theo thời gian

Tìm tất cả các file được truy cập trong thời gian 7 ngày gần nhất.

“find /home -atime 7”

+ **Tìm tất cả file được chỉnh sửa trong mốc thời gian**

Tìm tất cả các file được chỉnh sửa trong khoảng 1 tháng trước.

“find /home -mtime +30 -mtime -60”

+ **Tìm file được chỉnh sửa theo giờ**

Tìm tất cả các file được chỉnh sửa trong khoảng 2 tiếng trước.

“find /home -mmin -60”

- **Tìm theo dung lượng**

+ **Tìm tất cả các file có dung lượng lớn hơn dung lượng cho trước**

Tìm tất cả các file có dung lượng lớn hơn 10M trong thư mục /home

“find /home -size +10M”

+ **Tìm tất cả các file có dung lượng bé hơn dung lượng cho trước**

Tìm tất cả các file có dung lượng bé hơn 50M trong thư mục /home

“find /home -size -50M”

- **Tìm theo nội dung file**

+ **Tìm kiếm tất cả các file có định dạng .sh trong thư mục /home**

“find /home -type f -name "*.sh”

4) Đặt lệnh chạy định kì

- Để thực hiện các tác vụ một cách tự động theo định kì, ở chế độ nền của hệ thống thì ta sẽ sử dụng tiện ích **Crontab**. Crontab là một file chứa định dạng biểu (schedule) của các entries được chạy. đơn giản là một text file. Mỗi người dùng có một cron schedule riêng, file này thường nằm ở **/var/spool/cron**. Crontab files không cho phép tạo hoặc chỉnh sửa trực tiếp với bất kỳ trình text editor nào, trừ khi dùng lệnh **crontab**.
- Cài đặt crontab: **“sudo apt-get install cron”**

```
leediay@leediay:~$ sudo apt-get install cron
Reading package lists... Done
Building dependency tree
Reading state information... Done
cron is already the newest version (3.0pl1-136ubuntu1).
cron set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 64 not upgraded.
```

- Khởi động crontab: **“service crond start”**
- Một số lệnh thường dùng:
 - + **crontab -e**: tạo, chỉnh sửa các crontab.
 - + **crontab -l**: xem các crontab đã tạo.
 - + **crontab -r**: xóa file crontab.
- Cấu trúc của file crontab: **gồm 5 trường** xác định thời gian, sau đó là lệnh sẽ được chạy định kì:


```
* * * * * lệnh/tệp tin_chạy
```

- - - - -

| | | | |

| | | | +----- day of week (0 - 6) (Sunday=0)

| | | +----- month (1 - 12)

| | +----- day of month (1 - 31)

| +----- hour (0 - 23)

+----- min (0 - 59)
- Nếu một cột được gán ký tự *, nó có nghĩa là tác vụ sau đó sẽ được chạy ở mọi giá trị cho cột đó.

Field	Giải thích	Giá trị cho phép
MIN	phút	0 to 59
HOUR	Giờ	0 to 23
DOM	Ngày trong tháng	1-31
MON	Tháng	1-12
DOW	Ngày trong tuần	0-6
CMD	Lệnh	Các lệnh có thể thực hiện

- Ví dụ:
 - + Chạy script 10 phút 1 lần: **“10 * * * * script”**
 - + Chạy script vào 5 giờ sáng mỗi ngày: **“0 5 * * * script”**
 - + Tạo 1 tác vụ thực hiện 2 lần trong 1 ngày, như chạy script 2 lần 1 ngày vào lúc 5h sáng và 22h tối:

“00 05,22 * * * script”
 - + Tạo một tác vụ chỉ thực hiện vào các giờ cụ thể, như chạy script từ 8->9h từ thứ 3 đến thứ 5:

“00 8-9 * * 2-4 script”

5) Tạo người dùng/nhóm và phân quyền

- Trong Linux có 3 dạng đối tượng:
 - + **User:** tài khoản người dùng. User gồm user và super user (root). Root có quyền cao nhất trong hệ thống.
 - + **Group:** nhóm người dùng.
 - + **Other:** Những người dùng khác trong hệ thống.
- Quản lí người dùng và nhóm:
 - + **useradd || adduser:** Thêm một người dùng mới vào hệ thống.
 - + **userdel || deluser:** Xóa một người dùng khỏi hệ thống.
 - + **groupadd:** Thêm một nhóm mới vào hệ thống.
 - + **groupdel:** Xóa một nhóm khỏi hệthoong
 - + **passwd:** Đặt hoặc thay đổi mật khẩu của người dùng.
 - + **usermod:** Sửa đổi thông tin người dùng, như thay đổi tên người dùng hoặc nhóm
 - + **groupmod:** Sửa đổi thông tin nhóm
- **Ví dụ: Tạo mới 3 user: UserA và UserB thuộc GroupX, UserC thuộc GroupY**
 - + Đầu tiên ta sẽ tạo 3 user: UserA, UserB và UserC bằng lệnh:

“*sudo useradd + tên_user*” (tạo mới user chưa có mật khẩu)

Hoặc có thể sử dụng **“adduser”** khi muốn đặt luôn mật khẩu cho user...
 - + Sau đó ta tạo 2 group mới là GroupX và GroupY bằng lệnh tương tự như tạo user:

“*sudo groupadd + tên_group*”
 - + Bây giờ ta sẽ thêm UserB vào GroupX, UserC vào GroupC bằng cách sử dụng lệnh:

“*sudo usermod -g tên_group tên_user*”

```
leediay@ubuntu: ~
leediay@ubuntu:~$ sudo useradd UserA
[sudo] password for leediay:
leediay@ubuntu:~$ sudo useradd UserB
leediay@ubuntu:~$ sudo useradd UserC
leediay@ubuntu:~$ sudo groupadd GroupX
leediay@ubuntu:~$ sudo groupadd GroupY
leediay@ubuntu:~$ sudo usermod -g GroupX UserB
leediay@ubuntu:~$ sudo usermod -g GroupY UserC
leediay@ubuntu:~$
```

+ Để kiểm tra thông tin của 1 user, ta sẽ dùng lệnh: “*id tên_user*”

```
leediay@ubuntu:~$ id UserB
uid=1009(UserB) gid=1112(GroupX) groups=1112(GroupX)
leediay@ubuntu:~$
```

=> Như vậy, ở trên ta đã tạo thành công UserB thuộc GroupX

Phân quyền trong Linux:

- Hệ thống file của Linux/Unix cho phép cài đặt một số quyền truy nhập tới file và thư mục. Các quyền phổ biến được dùng đó là đọc (r), ghi (w) và thực thi (x). Ý nghĩa cụ thể của các quyền này đôi khi lệ thuộc vào đối tượng giám sát cụ thể:
 - + **Đọc:** Nếu là một file thì quyền này cho phép bạn mở file đó lên và đọc. Nếu là một thư mục thì nó cho phép bạn liệt kê danh sách file hay thư mục trong thư mục đó.
 - + **Ghi:** Quyền ghi cho phép sửa đổi nội dung của file. Nếu là thư mục thì nó cho phép có thể thêm, xóa và đổi tên các file trong thư mục đó.
 - + **Thực thi:** Với Windows có thể chạy với một file có đuôi ".exe" một cách dễ dàng. Khác so với Windows, trong Linux không thể chạy khi nó không được cấp quyền thực thi. Còn đối với thư mục thì không thể truy cập(cd) nếu không có quyền thực thi nó.
- Để liệt kê, xem quyền và chủ sở hữu các file hay thư mục, sử dụng “**ls -l**”

```
leediay@ubuntu:~$ ls -l
total 92
-rwxrwxr-x 1 leediay leediay 807 Nov 6 08:13 createUser.sh
-rwxrwxr-x 1 leediay leediay 268 Nov 6 08:14 deleteUser.sh
drwxr-xr-x 2 leediay leediay 4096 Oct 25 2022 Desktop
drwxr-xr-x 2 leediay leediay 4096 Oct 25 2022 Documents
drwxr-xr-x 2 leediay leediay 4096 Nov 12 23:36 Downloads
-rw-r--r-- 1 root root 19873 Oct 25 2022 get-docker.sh
drwxrwxr-x 2 leediay leediay 4096 Nov 5 08:48 john
drwxr-xr-x 2 leediay leediay 4096 Oct 25 2022 Music
drwxrwxr-x 4 leediay leediay 4096 Oct 25 2022 OnlineJudgeDeploy
drwxrwxrwx 8 leediay leediay 4096 Oct 25 2022 OnlineJudgeFE
-rw-rw-r-- 1 leediay leediay 86 Oct 25 2022 package-lock.json
-rwxrwxr-x 1 leediay leediay 65 Nov 6 08:03 phanQuyen.sh
drwxr-xr-x 2 leediay leediay 4096 Aug 29 07:51 Pictures
drwxr-xr-x 2 leediay leediay 4096 Oct 25 2022 Public
drwxrwxr-x 6 leediay leediay 4096 May 6 2023 sherlock
drwx----- 3 leediay leediay 4096 May 7 2023 snap
drwxrwxr-x 11 leediay leediay 4096 Aug 29 07:16 sqlmap-dev
drwxr-xr-x 2 leediay leediay 4096 Oct 25 2022 Templates
drwxr-xr-x 2 leediay leediay 4096 Oct 25 2022 Videos
```

+ Đầu tiên trong phần permissions bạn sẽ biết nó là dạng file hay thư mục.

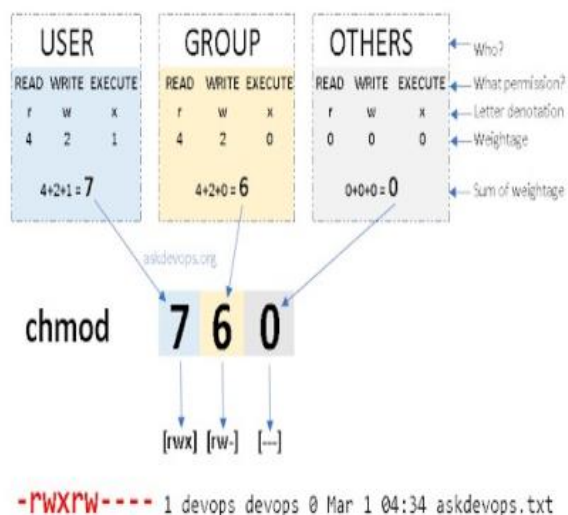
+ 9 kí hiệu tiếp theo, cứ mỗi 3 kí hiệu đại diện cho quyền của từng owner trong hệ thống, lần lượt là User, Group và Other.

+ Có 2 cách phân quyền bằng cách sử dụng “**chmod**” là phân quyền bằng chữ và phân quyền bằng số)

- **Cách 1: Phân quyền bằng chữ:**

Ký hiệu	Ý nghĩa	Ký hiệu	Ý nghĩa
		r	Đọc
d	Thư mục	w	Ghi
-	File	x	Thực thi
		-	Không có quyền

- **Cách 2: Phân quyền bằng số:**



Số	Ký hiệu	Ý nghĩa
0	---	Không có quyền
1	--x	Thực thi
2	-w-	Ghi
3	-wx	Thực thi + Ghi
4	r--	Đọc
5	r-x	Đọc + Thực thi
6	rw-	Đọc + Ghi
7	rwX	Đọc + Ghi + Thực thi

- **Ví dụ: Phân quyền file F1.txt chỉ cho phép thực thi bởi UserA/GroupX**

+ Đầu tiên ta sẽ tạo 1 file F1 bằng lệnh “**nano F1.txt**”, để phân quyền file F1 chỉ cho phép thực thi bởi UserA thì ta sẽ phải đổi quyền sở hữu file F1 sang cho UserA. Ta sẽ cần dùng lệnh:

“sudo chown tên_user tên_file”

```
leediay@leediay:~$ nano F1.txt
leediay@leediay:~$ cat F1.txt
hello
leediay@leediay:~$ sudo chown UserA F1.txt
```

+ Sau đó ta kiểm tra xem các quyền truy cập bằng lệnh “**ls -l**”

```
leediay@leediay:~$ ls -l
total 40
drwxr-xr-x 2 leediay leediay 4096 Thg 1 15 18:02 Desktop
drwxr-xr-x 2 leediay leediay 4096 Thg 1 15 18:02 Documents
drwxr-xr-x 2 leediay leediay 4096 Thg 1 15 18:02 Downloads
-rw-rw-r-- 1 UserA leediay 7 Thg 1 22 13:44 F1.txt
drwxr-xr-x 2 leediay leediay 4096 Thg 1 15 18:02 Music
drwxr-xr-x 2 leediay leediay 4096 Thg 1 15 18:02 Pictures
drwxr-xr-x 2 leediay leediay 4096 Thg 1 15 18:02 Public
drwx----- 3 leediay leediay 4096 Thg 1 15 18:03 snap
drwxr-xr-x 2 leediay leediay 4096 Thg 1 15 18:02 Templates
drwxr-xr-x 2 leediay leediay 4096 Thg 1 15 18:02 Videos
leediay@leediay:~$
```

+ Để phân quyền file F1 chỉ cho phép thực thi bởi UserA ta dùng: “**sudo chmod 766 F1.txt**”

```
leediay@leediay:~$ sudo chmod 766 F1.txt
leediay@leediay:~$ ls -l
total 40
drwxr-xr-x 2 leediay leediay 4096 Thg 1 15 18:02 Desktop
drwxr-xr-x 2 leediay leediay 4096 Thg 1 15 18:02 Documents
drwxr-xr-x 2 leediay leediay 4096 Thg 1 15 18:02 Downloads
-rwxrw-rw- 1 UserA leediay 7 Thg 1 22 13:44 F1.txt
drwxr-xr-x 2 leediay leediay 4096 Thg 1 15 18:02 Music
drwxr-xr-x 2 leediay leediay 4096 Thg 1 15 18:02 Pictures
drwxr-xr-x 2 leediay leediay 4096 Thg 1 15 18:02 Public
drwx----- 3 leediay leediay 4096 Thg 1 15 18:03 snap
drwxr-xr-x 2 leediay leediay 4096 Thg 1 15 18:02 Templates
drwxr-xr-x 2 leediay leediay 4096 Thg 1 15 18:02 Videos
leediay@leediay:~$
```

- Ví dụ: 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.

Lệnh “`sudo chmod u=rwx,g=r,o=r D1`” hoặc “`sudo chmod 744 D1`” sẽ giải quyết vấn đề này:

```
leediay@leediay:~$ mkdir D1
leediay@leediay:~$ sudo chown UserA D1
leediay@leediay:~$ sudo chmod 744 D1
leediay@leediay:~$ ls -l
total 44
drwxr--r-- 2 UserA leediay 4096 Thg 1 22 13:57 D1
```

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

Cũng như trên, ta dùng “`sudo chmod 711 D2`”

```
leediay@leediay:~$ mkdir D2
leediay@leediay:~$ sudo chown UserA D2
leediay@leediay:~$ sudo chmod 711 D2
leediay@leediay:~$ ls -l
total 48
drwxr--r-- 2 UserA leediay 4096 Thg 1 22 13:57 D1
drwx--x--x 2 UserA leediay 4096 Thg 1 22 13:59 D2
drwxr-xr-x 2 leediay leediay 4096 Thg 1 15 18:02 Desktop
drwxr-xr-x 2 leediay leediay 4096 Thg 1 15 18:02 Documents
drwxr-xr-x 2 leediay leediay 4096 Thg 1 15 18:02 Downloads
```

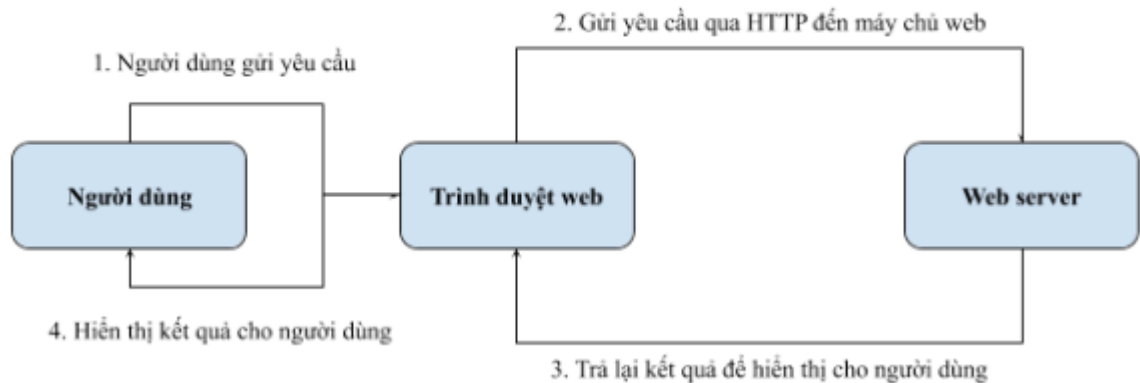
III. TÌM HIỂU VỀ WEB SERVER, VIRTUAL HOST

1) Khái niệm, cách hoạt động

- Web Server là **máy chủ** cài đặt các chương trình phục vụ các ứng dụng web. Web Server có khả

năng **tiếp nhận request** từ các trình duyệt web và **gửi phản hồi** đến client thông qua giao thức HTTP hoặc các giao thức khác. Có nhiều web server phổ biến hiện nay như: Apache, Nginx, IIS, ...

- **Cách thức hoạt động:** Bất cứ khi nào bạn xem một trang web trên internet, có nghĩa là bạn đang yêu cầu trang đó từ một web server. Khi bạn nhập URL trên trình duyệt của mình (ví dụ: <https://ducanh.com>) nó sẽ tiến hành các bước sau để gửi lại phản hồi cho bạn.



- + Bước 1: Người dùng gửi yêu cầu (requests)
- + Bước 2: Trình duyệt web gửi yêu cầu tới WebServer để xử lý
- + Bước 3: Máy chủ Web kiểm tra, trả về kết quả và trình duyệt hiển thị kết quả cho người dùng.

2) Giới thiệu về Apache

- Apache là **Web Server** được sử dụng rộng rãi nhất thế giới. Apache được phát triển và duy trì bởi một cộng đồng mã nguồn mở dưới sự bảo trợ của Apache Software Foundation. Apache được phát hành với giấy phép Apache License là được sử dụng tự do, miễn phí.
- Tính đến tháng 8 năm 2018, Apache ước tính phục vụ cho 54.2% các trang web đang hoạt động và 53.3% số máy chủ hàng đầu. Apache chạy trên các hệ điều hành như Windows, Linux, Unix, MacOS ...

3) Cài đặt Web Server Apache

- Đầu tiên, mở Terminal, ta sẽ update gói tin bằng lệnh: **“sudo apt update”**

```
leediay@leediay:~$ sudo apt update
[sudo] password for leediay:
Hit:1 http://vn.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://vn.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://vn.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
65 packages can be upgraded. Run 'apt list --upgradable' to see them.
leediay@leediay:~$
```

- Tiếp theo, ta sẽ tải Apache2 về máy bằng lệnh: **“sudo apt-get install apache2”**

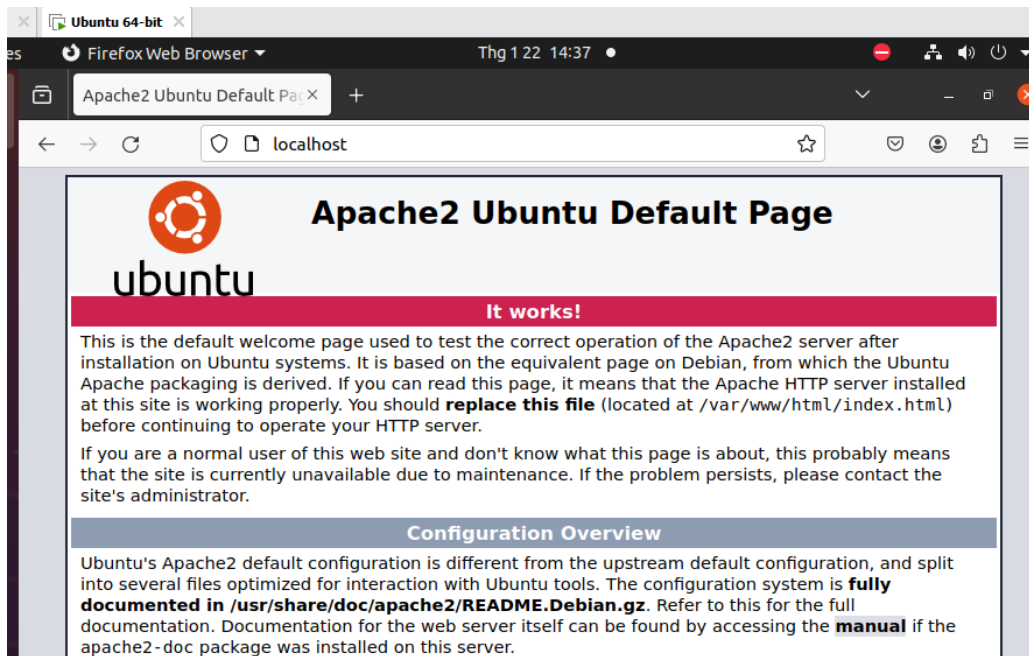
```
leediay@leediay:~$ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
apache2 is already the newest version (2.4.41-4ubuntu3.15).
0 upgraded, 0 newly installed, 0 to remove and 65 not upgraded.
```

- Để kiểm tra apache2 đã khởi động thành công hay chưa, mở: **“sudo systemctl status apache2”** để check:

```
leediay@leediay:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese>
   Active: active (running) since Mon 2024-01-22 13:37:56 +07; 57min ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 979 (apache2)
      Tasks: 55 (limit: 2218)
     Memory: 5.5M
    CGroup: /system.slice/apache2.service
            └─979 /usr/sbin/apache2 -k start
              └─980 /usr/sbin/apache2 -k start
                └─981 /usr/sbin/apache2 -k start

Thg 1 22 13:37:55 leediay systemd[1]: Starting The Apache HTTP Server...
Thg 1 22 13:37:56 leediay apache2[947]: AH00558: apache2: Could not reliably >
Thg 1 22 13:37:56 leediay systemd[1]: Started The Apache HTTP Server.
lines 1-15/15 (END)
```

- Vào browser, truy cập “**http://localhost**”, nếu hiển thị “**It works!**” thì Apache2 đã được cài đặt thành công trên máy:



- Tiếp theo, di chuyển đến thư mục gốc của webserver: “**cd var/www/html/**”
- Khởi động apache2, dùng lệnh “**sudo systemctl start apache2**”
- Dùng “**ls -l**” để xem các file hiện có trong đây:

```
leediay@leediay:/var/www/html$ ls -l
total 12
-rw-r--r-- 1 root root 10918 Thg 1 16 15:46 index.html
```

- Khởi động apache2: “**sudo systemctl start apache2**”
- Xóa file “index.html” mặc định trên máy đi: “**sudo rm -rf index.html**”
- Đến đây, ta sẽ tạo 1 file index.html mới bằng lệnh: “**sudo nano index.html**”

```
leediay@leediay:/var/www/html$ sudo systemctl start apache2
leediay@leediay:/var/www/html$ sudo rm -rf index.html
leediay@leediay:/var/www/html$ sudo nano index.html
leediay@leediay:/var/www/html$
```

- Viết 1 chương trình đơn giản:

```
leediay@leediay: /var/www/html
GNU nano 4.8 index.html
<h1> Hello World, Duc Anh Le! </h1>
```

- Lưu file bằng Ctrl + O và thoát khỏi nano bằng Ctrl + X.

- Khởi động lại apache2: **“sudo systemctl restart apache2”**
- Cuối cùng, tải lại **“http://localhost”** và ta đã thành công:



Activate Windows
Go to Settings to activate W

4) Cấu hình Virtual Hosts cho Web Server

- **Virtual Hosts** là một phương thức lưu trữ **cho phép lưu nhiều tên miền khác nhau** trên cùng một máy chủ Server. Virtual Hosts có thể được xem như một giải pháp cho phép bạn nhúng rất nhiều tên miền trên một địa chỉ IP của một Server duy nhất. Tùy vào cách cài đặt, Server sẽ tự hiểu tên miền nào đang hoạt động bên trong vị trí lưu trữ của Server.
- Ví dụ, ta sẽ tạo 2 Virtual Host chạy 2 Web cùng lúc trên Server:

Bước 1: Tạo thư mục cho trang web

+ Đầu tiên, ta cần tạo ra hai thư mục để chứa tập tin và dữ liệu của hai trang web trên. Mặc định Apache sử dụng **/var/www** làm thư mục chủ. Và ta sẽ tạo ra hai thư mục ở đó để lưu trữ dữ liệu cho hai trang web. Bạn có thể tạo thư mục ở bất kỳ đâu mà bạn muốn, theo ví dụ ở đây ta sẽ có hai thư mục là **/var/www/web1.com** và **/var/www/web2.com**.

+ Thông thường người ta sẽ sử dụng một thư mục con để lưu trữ dữ liệu để xử lý trang web cho người truy cập là **public_html**. Mục đích của điều này là để phục vụ lưu trữ dữ liệu một cách linh hoạt, bạn có thể lưu dữ liệu chỉ dành cho admin ở một thư mục khác

+ Tạo hai thư mục bằng hai lệnh sau:

“sudo mkdir -p /var/www/web1.com/public_html

sudo mkdir -p /var/www/web2.com/public_html

```
leediay@leediay:/var/www/html$ sudo mkdir -p /var/www/web1.com/public_html
[sudo] password for leediay:
leediay@leediay:/var/www/html$ sudo mkdir -p /var/www/web2.com/public_html
leediay@leediay:/var/www/html$
```

Bước 2: Tạo tập tin cấu hình Virtual Host

- + Trên Ubuntu, các tập tin cấu hình virtual host của Apache sẽ nằm trong thư mục **“/etc/apache2/sites-available”**. Ta sẽ nhìn thấy hai virtual host mặc định là **“000-default.conf”** và **“default-ssl.conf”**, ở đây ta sẽ dựa trên 2 virtual host này để thêm cấu hình cho domain khác.
- + Để kiểm tra các virtual host có sẵn, sử dụng lệnh sau: **“ls”**

```
leediay@leediay:/var/www$ cd /etc/apache2/sites-available/
leediay@leediay:/etc/apache2/sites-available$ ls
000-default.conf  default-ssl.conf
leediay@leediay:/etc/apache2/sites-available$
```

- + Sao chép 2 virtual host mặc định cho hai tên miền mới:

“sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/web1.com.conf”

sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/web2.com.conf”

```
leediay@leediay:/etc/apache2/sites-available$ sudo cp /etc/apache2/sites-availab
le/000-default.conf /etc/apache2/sites-available/web1.com.conf
leediay@leediay:/etc/apache2/sites-available$ sudo cp /etc/apache2/sites-availab
le/000-default.conf /etc/apache2/sites-available/web2.com.conf
leediay@leediay:/etc/apache2/sites-available$ ls
000-default.conf  default-ssl.conf  web1.com.conf  web2.com.conf
leediay@leediay:/etc/apache2/sites-available$
```

- + Mở và chỉnh sửa tập tin cấu hình */etc/apache2/sites-available/web1.com.conf*

“sudo nano /etc/apache2/sites-available/web1.com.conf”

- + Ta sẽ nhìn thấy nội dung như sau:

```

leediay@leediay: /etc/apache2/sites-available
GNU nano 4.8 /etc/apache2/sites-available/web1.com.conf
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log

```

+ Ở trên ta sẽ có một số thông tin cần cấu hình lại:

- **ServerName:** ở đây cần sửa lại thành domain của mình, như ví dụ ở trên là `web1.com`. Nhớ phải bỏ dấu `#` ở trước đi.
- **ServerAdmin:** email của mình cho domain này.
- **DocumentRoot:** đây là thư mục sẽ lưu trữ trang web mà ta đã tạo ở trên, như ví dụ ở trên là `/var/www/web1.com/public_html`

+ Sau khi chỉnh sửa, ta sẽ có một tập tin cấu hình virtual host với nội dung như sau:

```

leediay@leediay: /etc/apache2/sites-available
GNU nano 4.8 /etc/apache2/sites-available/web1.com.conf
<VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
ServerName www.web1.com
ServerAdmin leducanh@gmail.com
DocumentRoot /var/www/web1.com/public_html

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

```

+ Lưu lại và làm tương tự như với web2.

Bước 3: Bật Virtual Host để các cấu hình hoạt động

+ Sử dụng lệnh sau để bật virtual host:

“sudo a2ensite web1.com.conf

sudo a2ensite web2.com.conf”

```

leediay@leediay:/etc/apache2/sites-available$ sudo a2ensite web1.com.conf
Enabling site web1.com.
To activate the new configuration, you need to run:
  systemctl reload apache2
leediay@leediay:/etc/apache2/sites-available$ sudo a2ensite web2.com.conf
Enabling site web2.com.
To activate the new configuration, you need to run:
  systemctl reload apache2
leediay@leediay:/etc/apache2/sites-available$

```

+ Khởi động lại máy chủ web Apache2: *“sudo service apache2 restart”*

Bước 4: Kiểm tra xem việc tạo Virtual Host trên Apache đã đúng chưa

+ Để kiểm tra xem việc tạo virtual host trên Apache xem đã đúng chưa, ta cần phải tạo 2 tập tin và chạy nó trên trình duyệt.

+ Tạo tập tin index.html trong */var/www/web1.com/public_html* và

`/var/www/web2.com/public_html`

“`sudo nano /var/www/web1.com/public_html/index.html`”

`sudo nano /var/www/web2.com/public_html/index.html`”

```
GNU nano 4.8 /var/www/web1.com/public_html/index.html
<html>
<head>
<title> Web 01 của Duc Anh </title>
</head>

<body>
<h1> Đây là web 01 </h1>
</body>
</html>
```

và

```
GNU nano 4.8 /var/www/web2.com/public_html/index.html
<html>
<head>
<title> Web 02 của Duc Anh </title>
</head>

<body>
<h1> Đây là web 02 </h1>
</body>
</html>
```

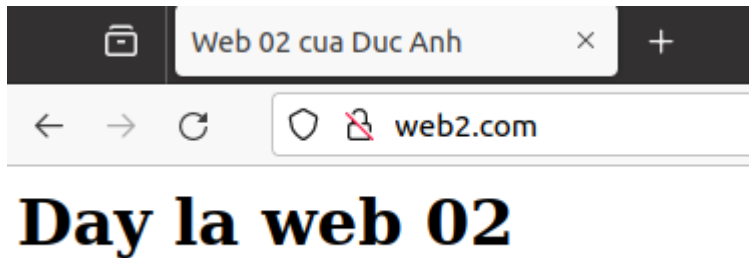
+ Tiếp theo, dùng lệnh **“`sudo nano /etc/hosts`”** để tạo 2 domain mới cho 2 web mình vừa tạo:

```
leediay@leediay: /etc/apache2/sites-available
GNU nano 4.8 /etc/hosts
127.0.0.1 localhost
127.0.1.1 leediay
127.0.0.1 web1.com
127.0.0.1 web2.com
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

+ Nhập và lưu domain lại như ảnh, cuối cùng thì khởi động lại apache2: **“`sudo systemctl restart apache2`”** và vào web để check và hoàn thành:



Và



IV. TÌM HIỂU VỀ LẬP TRÌNH SHELL SCRIPT

1) Khái niệm Shell Script

- Shell Script cũng giống như hầu hết các ngôn ngữ lập trình mà ta vẫn sử dụng như PHP, Python, Java, vv... Chỉ khác Shell Script là **ngôn ngữ chuyên dùng để thao tác với các Server Linux**. Shell Script được sử dụng cho nhiều mục đích khác nhau, chẳng hạn như thực thi lệnh shell, chạy nhiều lệnh cùng nhau, tùy chỉnh các tác vụ quản trị, thực hiện tự động hóa,... mà ta thường xuyên thực hiện trên máy tính của mình.
- Các tiện ích:
 - Có thể nhận đầu vào từ người dùng, tệp, hoặc kết quả từ màn hình.
 - Giúp cho chúng ta có thể tạo nhóm lệnh riêng.
 - Shell script giúp ta tiết kiệm thời gian.
 - Có khả năng thực hiện tự động một số công việc mà ta thường xuyên trên máy tính của mình.
- Các yêu cầu cơ bản khi viết:
 - + Các tập lệnh được lưu trữ dưới dạng các tập tin chúng ta có thể đặt tên tùy ý cho tập lệnh shell. Nhưng điều quan trọng đầu tiên là nó cần bắt đầu với một **shebang** ngay dòng đầu tiên (khi một Script được chạy, đầu tiên program loader sẽ dựa vào **shebang** để xác định Script đó sẽ được chạy bởi trình biên dịch nào)


```
#!/bin/bash
```

+ Tiếp theo thì nó phải là một tập tin thực thi. Để có thể phân quyền cho tập tin có thể thực thi thì chúng ta sử dụng lệnh chmod: **“chmod +x myscript”**

+ Để thực thi tập tin, ta có thể sử dụng những cách sau:

“./myscript”

“sh myscript”

“bash myscript”

+ Đặt đuôi tập tin là **“.sh”**

- Ví dụ, tạo và chạy 1 shell script in ra màn hình dòng chữ “Hello World!”

```
leediay@leediay:~$ nano myscript.sh
leediay@leediay:~$ cat myscript.sh
#!/bin/bash
echo "Hello World"
leediay@leediay:~$ chmod +x myscript.sh
leediay@leediay:~$ ./myscript.sh
Hello World
leediay@leediay:~$
```

2) Thực hành

Bài 1: Nhập 1 số n và so sánh số đó với 100. Đưa kết quả so sánh ra màn hình.

Code:

```

GNU nano 4.8                                bai1.sh
#!/bin/bash
echo "Nhap n"
read n
if [ $n -eq 100 ];
then
    echo "Bang nhau"
elif [ $n -gt 100 ];
then
    echo "Lon hon"
else
    echo "Be hon"
fi

```

Kết quả:

```

leediay@leediay:~$ nano bai1.sh
leediay@leediay:~$ chmod +x bai1.sh
leediay@leediay:~$ sh bai1.sh
Nhap n
3
Be hon
leediay@leediay:~$ sh bai1.sh
Nhap n
100
Bang nhau
leediay@leediay:~$ sh bai1.sh
Nhap n
153
Lon hon
leediay@leediay:~$

```

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

Code dùng for:

```

GNU nano 4.8                                bai2_use_for.sh
#!/bin/bash

echo "Nhap N"
sum=0
read n
for (( i=1; i<=n; i++ ))
do
    sum=`expr $sum + $i`
done
echo "Tong: $sum"

```

Kết quả:

```
leediay@leediay:~$ nano bai2_use_for.sh
leediay@leediay:~$ chmod +x bai2_use_for.sh
leediay@leediay:~$ bash bai2_use_for.sh
Nhap N
10
Tong: 55
leediay@leediay:~$
```

Code dùng while:

```
GNU nano 4.8          bai2 use while.sh
#!/bin/sh
echo "Nhap n"
sum=0
read n
i=0
while [ $i -le $n ]
do
sum=`expr $sum + $i`
i=`expr $i + 1`
done
echo "Tong: $sum."
```

Kết quả:

```
leediay@leediay:~$ nano bai2_use_while.sh
leediay@leediay:~$ chmod +x bai2_use_while.sh
leediay@leediay:~$ ./bai2_use_while.sh
Nhap n
15
Tong: 120.
leediay@leediay:~$
```

Code dùng until:

```

GNU nano 4.8                                bai2_use_until.sh
#!/bin/bash
echo Nhap n
read n
sum=$n
until [ $n -eq 0 ];
do
    n=$((n - 1))
    sum=$((sum + n))
done
echo $sum

```

Kết quả:

```

leediay@leediay:~$ ./bai2_use_until.sh
Nhap n
13
91
leediay@leediay:~$

```

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

```

GNU nano 4.8                                bai3.sh
#!/bin/bash
echo Nhap a:
read a
echo Nhap b:
read b
echo Nhap c:
read c
delta=$((b*b - 4*a*c))
if [ $delta -lt 0 ];
then echo Phương trình vô nghiệm
elif [ $delta -eq 0 ];
then
    echo Phương trình có nghiệm kép
    x=$(echo "scale=2; -($b / 2*$a)" | bc )
    echo $x
else
    echo Phương trình có 2 nghiệm
    x1=$(echo "scale=2; ((-$b + sqrt($delta)) / (2*$a))" | bc )
    echo $x1
    x2=$(echo "scale=2; ((-$b - sqrt($delta)) / (2*$a))" | bc )
    echo $x2
fi

```

Kết quả:

```

leediay@leediay:~$ nano bai3.sh
leediay@leediay:~$ chmod +x bai3.sh
leediay@leediay:~$ ./bai3.sh
Nhap a:
1
Nhap b:
3
Nhap c:
4
Phuong trinh vo nghiem
leediay@leediay:~$ ./bai3.sh
Nhap a:
1
Nhap b:
2
Nhap c:
1
Phuong co nghiem kep
-1.00
leediay@leediay:~$ ./bai3.sh
Nhap a:
1
Nhap b:
5
Nhap c:
4
Phuong trinh co 2 nghiem
-1.00
-4.00
leediay@leediay:~$

```

Bài 4: Nhập 1 số từ bàn phím, kiểm tra số đó có phải số nguyên tố hay không.

Code:

```

GNU nano 4.8                                                                    bai4.sh
echo Nhap n
read n
if [ $n -lt 2 ];
then
    echo "$n khong phai la so nguyen to"
else
    sum=0
    for (( i=2;i <= $n;i++ ));
    do
        if [ $(( $n % $i )) -eq 0 ]
        then sum=$((sum+1))
        fi
    done
    if [ $sum -eq 1 ];
    then echo "$n la so nguyen to"
    else
    echo "$n khong phai la so nguyen to"
    fi
fi

```

Kết quả:

```

leediay@leediay:~$ nano bai4.sh
leediay@leediay:~$ chmod +x bai4.sh
leediay@leediay:~$ ./bai4.sh
Nhap n
2
2 la so nguyen to
leediay@leediay:~$ ./bai4.sh
Nhap n
10
10 khong phai la so nguyen to
leediay@leediay:~$ ./bai4.sh
Nhap n
-1
-1 khong phai la so nguyen to
leediay@leediay:~$

```

Bài 5: Viết shell script info.sh hiển thị các thông tin về hệ thống, bao gồm:

- Tên máy, tên bản phân phối
- Phiên bản hệ điều hành
- Thông tin CPU (tên, 32bit hay 64bit, tốc độ)
- Thông tin bộ nhớ vật lý (tổng bao nhiêu MB)
- Thông tin ổ đĩa còn trống bao nhiêu MB
- Danh sách địa chỉ IP của hệ thống
- Danh sách user trên hệ thống (sắp xếp theo thứ tự abc)

Code:

```

GNU nano 4.8 info.sh
#!/bin/bash

echo "          [THONG TIN HE THONG]"
echo "Ten may: `whoami`"
echo "Ban phan phoi: `cat /etc/os-release | grep -w "VERSION"|cut -d '=' -f 2`"
echo "Phien ban dieu hanh: `cat /etc/os-release | grep -w "VERSION_ID" |cut -d '=' -f 2`"
echo "Ten CPU: `cat /proc/cpuinfo | grep -w "model name" | uniq | cut -d ':' -f 2`"
echo "Thong tin CPU: `lscpu | grep -w "Architecture" | cut -d ':' -f 2`"
echo "Toc do CPU `lscpu | grep -w "CPU MHz" | cut -d ':' -f 2` Mhz"
echo "`free -h | grep "G" | uniq | cut -d 'G' -f 1` Gb"
echo "Dia chi ip: `ip addr |grep -w "192"|cut -d '/' -f 2 | tr -s ' ' | cut -d ' ' -f 3`"
users="Danh sach user: `cut -d : -f 1 /etc/passwd | sort`"
echo $users | sed 's/ /\n/4g'

```

Kết quả:

```

leediay@leediay:~$ nano info.sh
leediay@leediay:~$ chmod +x info.sh
leediay@leediay:~$ ./info.sh
[THONG TIN HE THONG]
Ten may: leediay
Ban phan phoi: "20.04.6 LTS (Focal Fossa)"
Phien ban dieu hanh: "20.04"
Ten CPU: Intel(R) Core(TM) i5-8500 CPU @ 3.00GHz
Thong tin CPU: x86_64
Toc do CPU 3000.006 Mhz
Mem: 1,9 Gb
Dia chi ip: 192.168.88.255
Danh sach user: _apt
avahi
avahi-autoipd
backup
bin
colord
cups-pk-helper
daemon
dnsmasq
fwupd-refresh

```

V. TÌM HIỂU VỀ HTTP CLIENT

1) Cấu trúc của dòng Request Line

- Một **HTTP Request** sẽ gồm 3 phần chính là: **Request line**, **Header** và **Body**.
- **Request line** là dòng đầu tiên của gói HTTP Request, bao gồm 3 trường: **phương thức** (method), **đường dẫn** (path – đa số là URL) và **phiên bản giao thức** (HTTP version).
- Trong đó:
 - + **Phương thức (method)** có thể là: **GET, POST, HEAD, PUT, DELETE...** Hai phương thức phổ biến nhất là **GET** và **POST**, trong ví dụ trên là phương thức **GET** thường dùng để yêu cầu tài nguyên cung cấp trong trường URL.
 - + **Đường dẫn (path)** dùng để định danh nguồn tài nguyên mà client yêu cầu, bắt buộc phải có ít nhất là dấu “/”.
 - + **Phiên bản giao thức (HTTP version)**: là phiên bản HTTP client đang sử dụng (thường là HTTP/1.0 hoặc HTTP/1.1).

- Ví dụ ta có 1 HTTP Request dưới đây:

Request Header	Value
(Request-Line)	GET / HTTP/1.1
Host	www.stdio.vn
User-Agent	Mozilla/5.0 (Windows NT 6.3; WOW64; rv:36.0) Gecko/20100101 Firefox/36.0
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language	vi-VN,vi;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding	gzip, deflate
Cookie	_ga=GA1.2.529740874.1426574792; PHPSESSID=0e61i2eb0mff46hejl7r0kfms2; _gat=1; ci_session=a%3...
Connection	keep-alive

2) Các HTTP Header thường gặp

- **HTTP Header** là một phần cốt lõi của **HTTP Request** và **HTTP Response**, trong mỗi HTTP Header đều kèm theo các thông tin phản hồi và yêu cầu của phía client và server, các dòng này là không bắt buộc, viết ở định dạng “**Name:Value**”.
- Ví dụ về các Header thường gặp:

```
GET /ajaxpro/EduSoft.Web.UC.DangNhap,EduSoft.Web.ashx HTTP/2
Host: qltd.ptit.edu.vn
Cookie: ASP.NET_SessionId=zhh2rwvdKsvgxa45rrpi5yaj
Sec-Ch-Ua: "Chromium";v="103", ".Not/A)Brand";v="99"
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.5060.134
Safari/537.36
Sec-Ch-Ua-Platform: "Windows"
Accept: */*
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: script
Referer: https://qltd.ptit.edu.vn/default.aspx?page=dangnhap
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
```

- + **Host**: cho biết bạn đang gửi truy vấn đến host nào. Điều này rất hữu ích nếu như nhiều trang web cùng đặt chung 1 web server. Nó bao gồm cả domain và subdomain.
- + **Cookie**: được gửi lại lên web server nếu như trước đây đã từng truy cập vào địa chỉ này.
- + **Referer**: cho ta biết rằng URL trên có nguồn gốc từ địa chỉ nào.

- + **User-Agent:** dùng để cung cấp cho web server biết là ta đang dùng trình duyệt web nào, hệ điều hành gì?
- + **Location:** Header này được sử dụng để chuyển hướng (redirect). Với status code là 301 hoặc 302.
- + **Content-Length:** Web server thông báo cho phía client về dung lượng của gói tin gửi cho phía client. Điều này rất tốt cho việc download vì trình duyệt có thể dự đoán mất bao nhiêu thời gian để download.
- + **Connection:** tùy chọn điều khiển cho kết nối hiện thời. Ví dụ: keep-alive, upgrade, close...
- + **Strict-Transport-Security: Bảo mật truyền tải nghiêm ngặt HTTP (HSTS)** là một cơ chế chính sách bảo mật web được thiết kế để bảo vệ các trang web **HTTPS** chống lại các cuộc tấn công hạ cấp và chiếm quyền điều khiển cookie. Máy chủ web được định cấu hình để sử dụng **HSTS** hướng dẫn trình duyệt web (hoặc phần mềm ứng dụng khác) chỉ sử dụng các kết nối **HTTPS** và không cho phép sử dụng giao thức **HTTP**.
- + **Cache-Control:** Mục đích của **Cache-Control** là để các nhà phát triển có thể kiểm soát tốt hơn nội dung trang web và giới hạn thời gian tải về. **Cache-Control** header bao gồm:
 - **max-age=[seconds]:** gán thời gian tối đa trước khi file hết hạn. Giá trị này tính bằng giây.
 - **s-maxage=[seconds] :** cũng tương tự như max-age nhưng chỉ áp dụng cho những cache được chia sẻ (ví dụ như proxy).
 - **public:** Đánh dấu rằng gói response này có thể cached. Thông thường người ta set thuộc tính public cho những thông tin không bảo mật và báo cho trình duyệt biết rằng có thể cache. Thông thường, nếu yêu cầu xác thực **HTTP** thì response sẽ tự động private.
 - **private:** Chỉ cho phép cache thông tin của một user nào đó vào trình duyệt của họ. Và những shared cache (như proxy) thì không được phép.
 - **no-store:** xác định content sẽ không được lưu vào cache.
 - **no-cache:** cho biết rằng cache có thể được duy trì nhưng content được lưu trong cache sẽ được validate lại từ server.

3) Các mã trả về của HTTP Response

Một số **loại** Status-Code thông dụng mà server trả về cho client như sau:

- **1xx: Information Message:** các status code này chỉ có tính chất tạm thời, client có thể không quan tâm.
- **2xx Successful:** khi đã xử lý thành công request của client, server trả về status dạng này:
 - **200 OK:** request thành công.
 - **202 Accepted:** request đã được nhận, nhưng không có kết quả nào trả về, thông báo cho client tiếp tục chờ đợi.
 - **204 No Content:** request đã được xử lý nhưng không có thành phần nào được trả về.
 - **205 Reset:** giống như 204 nhưng mã này còn yêu cầu client reset lại document view.
 - **206 Partial Content:** server chỉ gửi về một phần dữ liệu, phụ thuộc vào giá trị range header của client đã gửi.
- **3xx Redirection:** server thông báo cho client phải thực hiện thêm thao tác để hoàn tất request:
 - **301 Moved Permanently:** tài nguyên đã được chuyển hoàn toàn tới địa chỉ Location trong HTTP response.
 - **303 See other:** tài nguyên đã được chuyển tạm thời tới địa chỉ Location trong HTTP response.
 - **304 Not Modified:** tài nguyên không thay đổi từ lần cuối client request, nên client có thể sử dụng đã lưu trong cache.
- **4xx Client error:** Lỗi của client:
 - **400 Bad Request:** request không đúng dạng, cú pháp.
 - **401 Unauthorized:** client chưa xác thực.
 - **403 Forbidden:** client không có quyền truy cập.
 - **404 Not Found:** không tìm thấy tài nguyên.
 - **405 Method Not Allowed:** phương thức không được server hỗ trợ.
- **5xx Server Error:** lỗi của server:
 - **500 Internal Server Error:** có lỗi trong quá trình xử lý của server.
 - **501 Not Implemented:** server không hỗ trợ chức năng client yêu cầu.

- **503 Service Unavailable:** Server bị quá tải, hoặc bị lỗi xử lý.

