



### LAB 3

## SỬ DỤNG SHELL SCRIPTING, QUẢN LÝ TIẾN TRÌNH, TẬP TIN NHẬT KÝ HỆ THỐNG

Họ tên và MSSV: Lê Tuấn Kiệt-B1909935

Nhóm học phần: 2

*- Các sinh viên bị phát hiện sao chép bài của nhau sẽ nhận 0đ cho tất cả bài thực hành của môn này.*

*- Bài nộp phải ở dạng PDF, hình minh họa phải rõ ràng chi tiết.*

#### 1. Cài đặt CentOS

Thực hiện cài đặt CentOS 6 (hoặc CentOS 7,8) vào máy tính cá nhân (hoặc máy ảo) của bạn.

#### 2. Quản trị với shell scripting

##### 2.1.

##### 2.2. Thực hiện các lệnh bên dưới và cho biết ý nghĩa của chúng (chụp hình minh họa):

`hostname`

`hostname -I`

`whoami`

`df -H`

`ps -eo pid,%mem,%cpu,comm --sort -rss | head -n 3`

(KHÔNG CÓ KHOẢNG TRẮNG SAU DẤU PHẨY)

- `hostname` : Kiểm tra host hiện tại

```
[b1909935@zen ~]$ hostname  
zen
```

- `hostname -I` : Hiển thị tất cả địa chỉ IPv4 mạng của host ()

```
[b1909935@zen ~]$ hostname -I  
192.168.2.176 192.168.122.1
```

- `whoami` (*Who Am I*): hiển thị ra người dùng đang thao tác

```
[b1909935@zen ~]$ whoami  
b1909935
```

- `df -H` : Kiểm tra thông tin tất cả ổ cứng

```
[b1909935@zen ~]$ df -H
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        6.1G   0    6.1G   0% /dev
tmpfs           6.2G   56M   6.1G   1% /dev/shm
tmpfs           6.2G   19M   6.2G   1% /run
tmpfs           6.2G    0    6.2G   0% /sys/fs/cgroup
/dev/sda2       64G   7.0G   53G   12% /
/dev/sda1       210M   7.3M   203M   4% /boot/efi
tmpfs           1.3G   1.2M   1.3G   1% /run/user/42
tmpfs           1.3G   4.9M   1.3G   1% /run/user/1000
```

- `ps -eo pid,%mem,%cpu,comm --sort -rss | head -n 3` : *Giải thích bên dưới*

```
[b1909935@zen ~]$ ps -eo pid,%mem,%cpu,comm --sort -rss | head -n 3
  PID %MEM %CPU COMMAND
  4259   3.6  14.7 Web Content
  3796   3.0  14.8 firefox
```

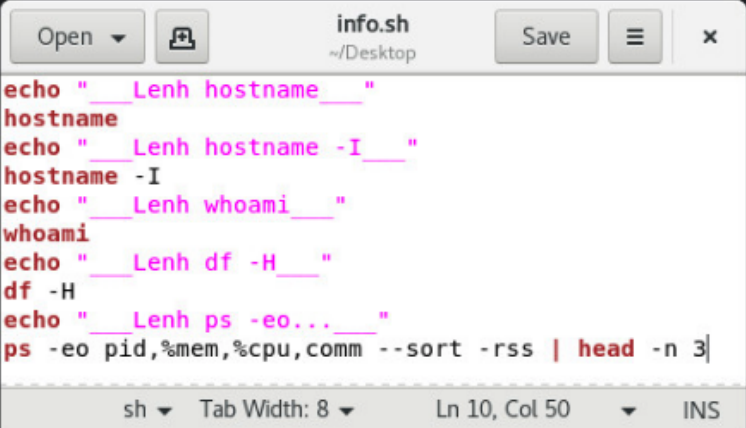
#### 💡 Giải thích chi tiết :

- `ps` (*Process Status*) : công cụ quản lí tiến trình
- `-eo` : Hiển thị tất cả tiến trình có chọn lọc
- `pid` (*Process ID*) : Mã tiến trình
- `%mem,%cpu` : Sử dụng bộ nhớ và vi xử lý tiến trình ( % )
- `comm` : đường dẫn của tiến trình
- `--sort` : sắp xếp tiến trình
- `-rss` (*resident set size*) : đi chung với `vsz` , hiển thị **dung lượng bộ nhớ được phân bổ** cho tiến trình trong **RAM**
- `| head -n 3` : sử dụng **ống dẫn** ( *pipe* ) để hiển thị 3 dòng đầu tiên

**2.3.** Viết shell script có tên *info.sh* thực hiện tất cả các lệnh ở 2.1. (chụp hình minh hoạ)

- Sử dụng `gedit + <file>` để tạo file như sau :

```
[b1909935@zen Desktop]$ gedit info.sh
```



```
echo "__Lenh hostname__"
hostname
echo "__Lenh hostname -I__"
hostname -I
echo "__Lenh whoami__"
whoami
echo "__Lenh df -H__"
df -H
echo "__Lenh ps -eo...__"
ps -eo pid,%mem,%cpu,comm --sort -rss | head -n 3
```

- Sử dụng lệnh `bash + <file>` để thực thi file :

```
[b1909935@zen Desktop]$ bash info.sh
__Lenh hostname__
zen
__Lenh hostname -I__
192.168.2.176 192.168.122.1
__Lenh whoami__
b1909935
__Lenh df -H__


| Filesystem | Size | Used | Avail | Use% | Mounted on     |
|------------|------|------|-------|------|----------------|
| devtmpfs   | 6.1G | 0    | 6.1G  | 0%   | /dev           |
| tmpfs      | 6.2G | 54M  | 6.1G  | 1%   | /dev/shm       |
| tmpfs      | 6.2G | 19M  | 6.2G  | 1%   | /run           |
| tmpfs      | 6.2G | 0    | 6.2G  | 0%   | /sys/fs/cgroup |
| /dev/sda2  | 64G  | 7.0G | 53G   | 12%  | /              |
| /dev/sda1  | 210M | 7.3M | 203M  | 4%   | /boot/efi      |
| tmpfs      | 1.3G | 1.2M | 1.3G  | 1%   | /run/user/42   |
| tmpfs      | 1.3G | 6.3M | 1.3G  | 1%   | /run/user/1000 |


__Lenh ps -eo...__


| PID  | %MEM | %CPU | COMMAND     |
|------|------|------|-------------|
| 4259 | 3.4  | 3.4  | Web Content |
| 4335 | 3.1  | 12.6 | Web Content |

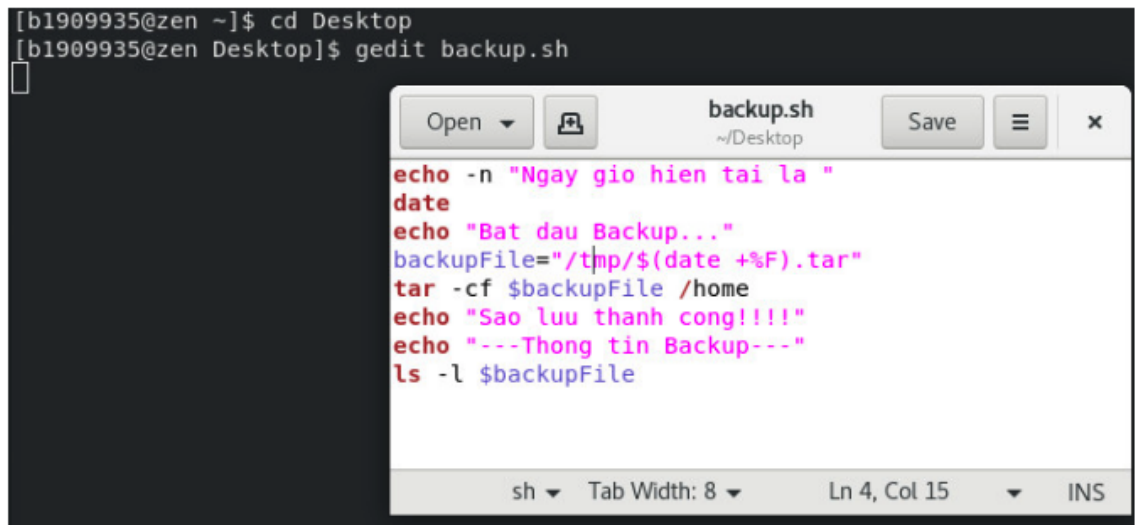

```

**2.4.** Viết shell script có tên *backup.sh* thực hiện:

- In ra ngày giờ hiện tại
- Nén toàn bộ thư mục **/home** thành tập tin **/tmp/<YYYY-MM-DD>.tar** (YYYY-MM-DD là ngày hiện tại, ví dụ: 2020-04-22.tar)
- In thông tin đầy đủ của tập tin **/tmp/<YYYY-MM-DD>.tar**
- In thông báo "Sao lưu thành công!!!!"

Thực thi *backup.sh* để kiểm tra (chụp hình minh họa).

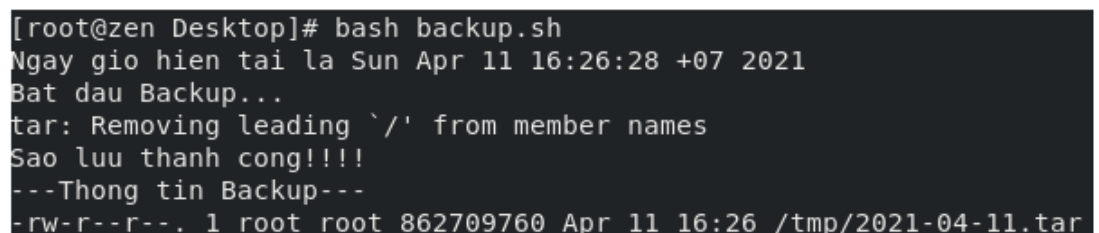
- Sử dụng `gedit + <file>` để tạo file như sau :



```
[b1909935@zen ~]$ cd Desktop
[b1909935@zen Desktop]$ gedit backup.sh

echo -n "Ngày giờ hiện tại là "
date
echo "Bắt đầu Backup..."
backupFile="/tmp/$(date +%F).tar"
tar -cf $backupFile /home
echo "Sao lưu thành công!!!!"
echo "---Thông tin Backup---"
ls -l $backupFile
```

- Sử dụng lệnh `bash + <file>` để thực thi file :



```
[root@zen Desktop]# bash backup.sh
Ngày giờ hiện tại là Sun Apr 11 16:26:28 +07 2021
Bắt đầu Backup...
tar: Removing leading `/' from member names
Sao lưu thành công!!!!
---Thông tin Backup---
-rw-r--r--. 1 root root 862709760 Apr 11 16:26 /tmp/2021-04-11.tar
```

**3. Lên lịch công việc định kỳ với cron**

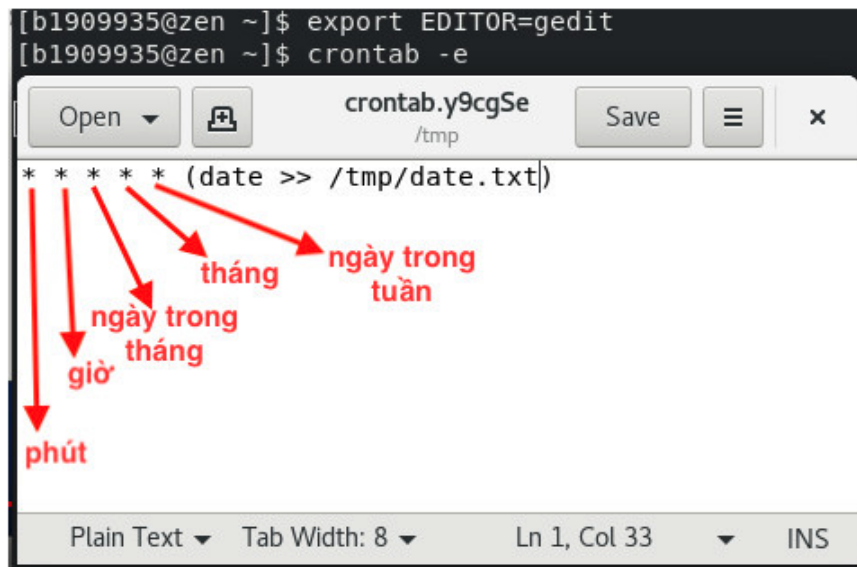
Cron là một tiện ích trong Linux cho phép máy tính thiết lập thời gian biểu để thực hiện công việc một cách định kỳ. Một crontab file chứa danh sách các lệnh sẽ được thực thi, kèm theo thông tin về thời điểm lặp lại việc thực thi. Để hiệu chỉnh file crontab với trình soạn thảo nano, ta dùng các lệnh sau:

```
export EDITOR=nano
crontab -e
```

Cho biết cú pháp để thực hiện các yêu cầu sau từ crontab file:

**3.1. Chạy lệnh `date` mỗi phút một lần, sau đó ghi kết quả vào cuối tập tin `/tmp/date.txt` (chụp hình minh hoạ)**

Dùng lệnh `export EDITOR=<tên editor>`, để thay đổi trình chỉnh sửa mặc định



💡 Giải thích :

`crontab -e` : chỉnh sửa file `crontab`

`crontab -l` : xem file `crontab`

```
[b1909935@zen ~]$ crontab -l
* * * * * (date >> /tmp/date.txt)
```

`crontab -r` : xoá file `crontab`

==> Sau một phút tự động sẽ cập nhật giờ trong file `/tmp/data.txt` :

```
[b1909935@zen ~]$ cat /tmp/date.txt
Sun Apr 11 16:39:01 +07 2021
```

```
[b1909935@zen ~]$ cat /tmp/date.txt
Sun Apr 11 16:39:01 +07 2021
Sun Apr 11 16:40:01 +07 2021
Sun Apr 11 16:41:01 +07 2021
Sun Apr 11 16:42:01 +07 2021
Sun Apr 11 16:43:01 +07 2021
Sun Apr 11 16:44:01 +07 2021
Sun Apr 11 16:45:01 +07 2021
```

3.2. Thực thi *backup.sh* ở 2.3 vào 23:00 giờ ngày 10, 20 và 30 hàng tháng (chụp hình minh họa).

```
[b1909935@zen ~]$ crontab -l
* * * * * (date >> /tmp/date.txt)
* 23 10,20,30 * * (/home/b1909935/Desktop/backup.sh)
```

==> Cứ mỗi 23h ngày 10,20,30 hằng tháng tự động sẽ thực thi file *backup.sh*

#### 4. Thao tác với tiến trình

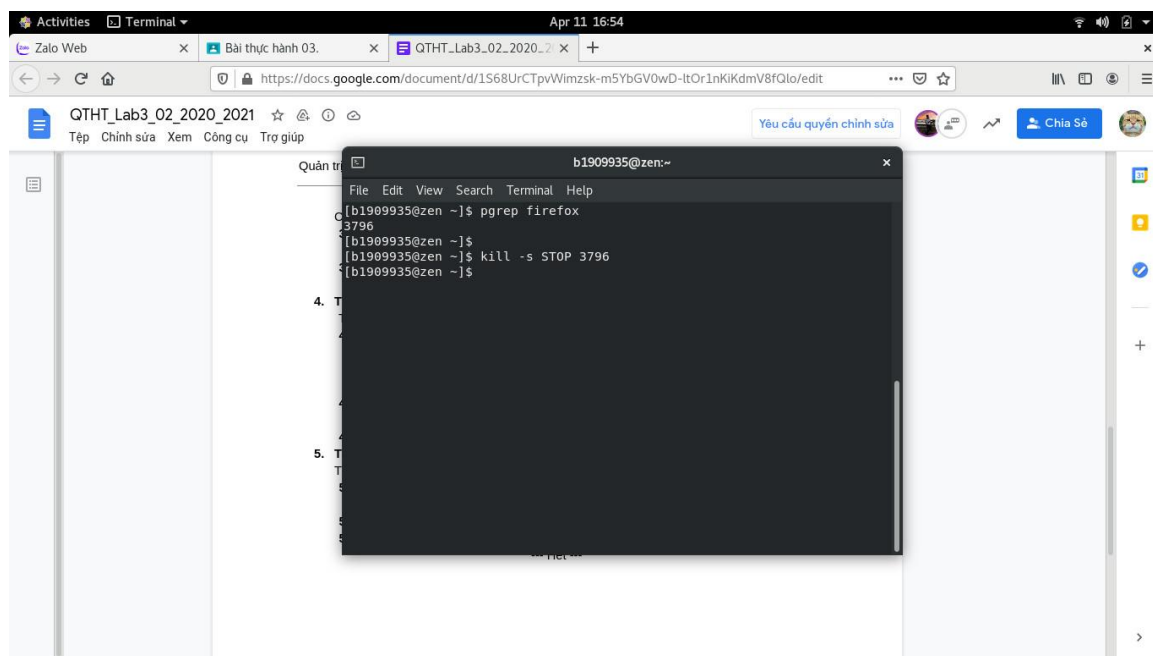
Tìm hiểu và thực hiện các yêu cầu sau:

4.1. Mở trình duyệt Firefox. Sau đó dùng lệnh **ps -aux** hoặc **pgrep** tìm PID của firefox. Tiếp theo, dùng lệnh **kill** để tạm dừng tiến trình firefox (chụp hình minh họa). Điều gì xảy ra khi bạn dịch chuyển cửa sổ firefox hoặc nhấn chọn menu của nó ngay lúc này?

- Sử dụng lệnh **pgrep** để tìm ID tiến trình :

```
[b1909935@zen ~]$ pgrep firefox
3796
```

- Sử dụng lệnh **kill -s STOP <PID>** sẽ tạm dừng tiến trình :



💡 Ngay lập tức ta không thể thao tác được gì từ *Firefox*

**4.2.** Dùng lệnh **kill** để phục hồi trạng thái trước đó của firefox và quan sát kết quả (chụp hình minh hoạ).

Sử dụng lệnh sau đây để khôi phục tiến trình vừa dùng :

```
$ kill -s CONT <PID>
```

```
[b1909935@zen ~]$ kill -s CONT 3796
```

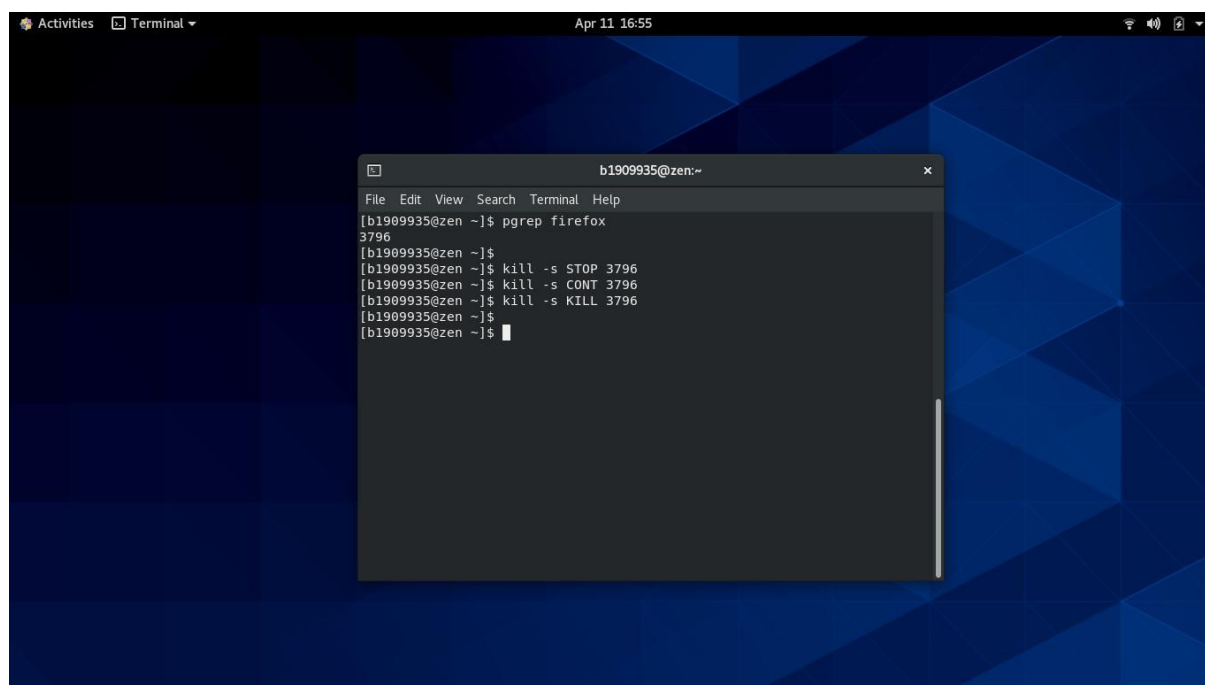
💡 Ngay lập tức ta có thể thao tác **Firefox** lại bình thường

**4.3.** Dùng lệnh **kill** để hủy tiến trình firefox (chụp hình minh hoạ).

Sử dụng lệnh sau đây để thoát tiến trình ngay lập tức :

```
$ kill -s KILL <PID>
```

```
[b1909935@zen ~]$ kill -s KILL 3796
```



💡 Tiến trình **Firefox** ngay lập tức sẽ bị tắt đi.



## 5. Tập tin log

Tìm hiểu và thực hiện các yêu cầu sau:

### 5.1. Tìm thông tin về người dùng, thời gian của lần đăng nhập sau cùng vào hệ thống (chụp hình minh họa).

Sử dụng lệnh `last` để hiển thị thông tin nhật ký đăng nhập trên máy:

```
[root@zen b1909935]# last
b1909935 tty2      tty2      Sun Apr 11 14:24  still logged in
reboot   system boot  4.18.0-240.15.1 Sun Apr 11 14:24  still running
b1909935 tty2      tty2      Sat Mar 27 13:58 - down (04:59)
reboot   system boot  4.18.0-240.15.1 Sat Mar 27 20:58 - 18:58 (-1:59)
b1909935 tty3      tty3      Sat Mar 27 13:46 - down (00:11)
b1909935 tty2      tty2      Sat Mar 27 13:42 - down (00:15)
reboot   system boot  4.18.0-240.15.1 Sat Mar 27 13:40 - 13:57 (00:16)
b1909935 tty3      tty3      Sat Mar 27 13:28 - crash (00:12)
b1909935 tty2      tty2      Sat Mar 27 13:26 - crash (00:14)
b1909935 tty2      tty2      Sat Mar 27 12:52 - 13:26 (00:34)
reboot   system boot  4.18.0-240.15.1 Sat Mar 27 12:51 - 13:57 (01:06)
b1909935 tty2      tty2      Mon Mar 15 21:49 - down (01:41)
reboot   system boot  4.18.0-240.15.1 Mon Mar 15 21:48 - 23:30 (01:41)
b1909935 tty2      tty2      Sun Mar 14 22:03 - down (00:08)
reboot   system boot  4.18.0-240.15.1 Mon Mar 15 05:02 - 22:12 (-6:50)
b1909935 tty2      tty2      Sun Mar 14 21:19 - down (00:43)
b1909935 tty2      tty2      Sun Mar 14 21:16 - 21:18 (00:02)
b1909935 tty2      tty2      Sun Mar 14 20:56 - 21:15 (00:19)
b1909935 tty2      tty2      Sun Mar 14 20:55 - 20:56 (00:00)
reboot   system boot  4.18.0-240.el8.x Sun Mar 14 20:53 - 22:02 (01:09)

wtm begins Sun Mar 14 20:53:25 2021
```

thời gian đăng nhập

💡 Nhưng ta có thể sử dụng lệnh `last` kèm theo ống dẫn `head -n -1` để hiển thị thời gian đăng nhập cuối cùng như sau :

```
[root@zen b1909935]# last | head -n 1
b1909935 tty2      tty2      Sun Apr 11 14:24  still logged in
```

### 5.2. Tạo một người dùng mới.

Sử dụng lệnh `adduser` hoặc `useradd` để tạo người dùng mới (dùng tài khoản root) (Mình tạo tên `newUser`)

### 5.3. Tìm thời gian người dùng ở 5.2 được tạo ra (chụp hình minh họa).

Thông tin tạo người dùng nằm trong file `/var/log/secure`, sử dụng lệnh `cat` để in ra (dùng tài khoản root)

💡 Ta nên kèm theo ống dẫn để tìm kiếm nhanh hơn như sau :

```
[root@zen b1909935]# cat /var/log/secure | grep newUser
Apr 11 17:04:46 zen useradd[13374]: new group: name=newUser, GID=1004
Apr 11 17:04:46 zen useradd[13374]: new user: name=newUser, UID=1002, GID=1004,
home=/home/newUser, shell=/bin/bash
```

--- Hết ---