



LAB 3

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

Họ tên và MSSV: Trương Đăng Trúc Lâm

Nhóm học phần: B2111933

- 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 9 Stream vào máy tính cá nhân (hoặc máy ảo) của bạn **nếu cần** (KHÔNG cần chụp hình minh họa).

2. Quản trị với shell scripting

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

hostname

```
B2111933@localhost:~  
[B2111933@localhost ~]$ hostname  
localhost.localdomain  
[B2111933@localhost ~]$
```

Lệnh **hostname** dùng để thiết lập hoặc hiển thị tên máy chủ, tên miền hoặc tên nút hiện tại của hệ thống Linux.

hostname -I

```
B2111933@localhost:~  
[B2111933@localhost ~]$ hostname -I  
10.0.2.15  
[B2111933@localhost ~]$
```

Lệnh **hostname -I** dùng để hiển thị tất cả địa chỉ IP của máy chủ.

whoami

```
B2111933@localhost:~  
[B2111933@localhost ~]$ whoami  
B2111933  
[B2111933@localhost ~]$
```

Lệnh **whoami** có chức năng xuất ra tên tài khoản người dùng đang đăng nhập

df -H

```
[B2111933@localhost ~]$ df -H
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.2M   0    4.2M   0% /dev
tmpfs           2.0G   0    2.0G   0% /dev/shm
tmpfs           769M   9.6M  759M   2% /run
/dev/mapper/cs-root 24G   6.4G   17G  28% /
/dev/sda1       1.1G  386M   678M  37% /boot
tmpfs           385M  115k   384M   1% /run/user/1000
/dev/sdb1       8.4G   29k   8.0G   1% /data
[B2111933@localhost ~]$
```

Lệnh **df -H** được sử dụng để hiển thị tóm tắt đầy đủ về việc sử dụng không gian đĩa cứng còn sẵn và được sử dụng của hệ thống tập tin trên hệ thống Linux. Tham số **-H** cho phép ta hiển thị thông số dưới dạng lũy thừa của 1000 (VD: 1.1gb).

ps -eo pid,%mem,%cpu,comm --sort -rss | head -n 3
(KHÔNG CÓ KHOẢNG TRẮNG SAU DẤU PHẨY)

```
[B2111933@localhost ~]$ ps -eo pid,%mem,%cpu,comm --sort -rss | head -n 3
PID %MEM %CPU COMMAND
4937  7.4  0.2 gnome-shell
4350  4.3  0.0 packagekitd
[B2111933@localhost ~]$
```

Lệnh **ps -eo pid,%mem,%cpu,comm --sort -rss | head -n 3** cho phép ta hiển thị **ID tiến trình**, **% ram sử dụng**, **% cpu sử dụng** và **lệnh thực thi** của **2 tiến trình sử dụng nhiều bộ nhớ nhất**.

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

```
[B2111933@localhost ~]$ nano ./info.sh
```

Dùng công cụ **nano** để tạo ra tập tin **info.sh**.

```
GNU nano 5.6.1 ./info.sh Modified
echo "swim swim swim swim swim"
hostname
echo "swim swim swim swim swim"
hostname -I
echo "swim swim swim swim swim"
whoami
echo "swim swim swim swim swim"
df -H
echo "swim swim swim swim swim"
ps -eo pid,%mem,%cpu,comm --sort -rss | head -n 3
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute
^X Exit      ^R Read File ^_ Replace   ^U Paste     ^J Justify
```

Nhập vào các lệnh trên.

```

B2111933@localhost:~
[B2111933@localhost ~]$ sudo chmod 755 ./info.sh
[B2111933@localhost ~]$ ./info.sh
swim swim swim swim swim swim
localhost.localdomain
swim swim swim swim swim swim
10.0.2.15
swim swim swim swim swim swim
B2111933
swim swim swim swim swim swim
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.2M   0  4.2M   0% /dev
tmpfs           2.0G   0  2.0G   0% /dev/shm
tmpfs           769M  9.6M  759M   2% /run
/dev/mapper/cs-root 24G  6.4G   17G  28% /
/dev/sdal       1.1G  386M  678M  37% /boot
tmpfs           385M  115k  384M   1% /run/user/1000
/dev/sdb1       8.4G   29k  8.0G   1% /data
swim swim swim swim swim swim
  PID %MEM %CPU COMMAND
    4937  7.4  0.3 gnome-shell
    4350  4.3  0.0 packagekitd
[B2111933@localhost ~]$

```

Chỉnh sửa các quyền trên tập tin **info.sh** với lệnh **chmod** và chạy tập tin **info.sh**.

2.3. 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 **/tmpn/<YYYY-MM-DD>.tar**
- In thông ra thông báo “Sao lưu thành công!!!!”

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

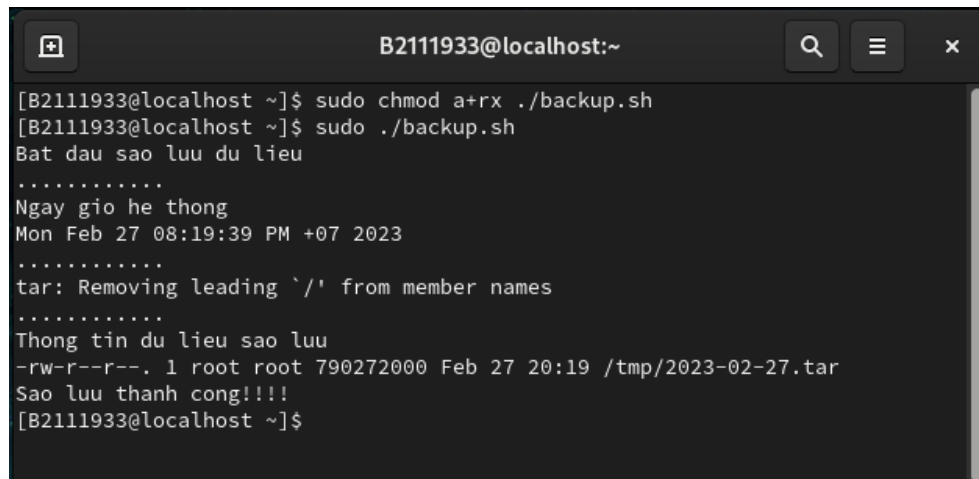
```

B2111933@localhost:~ — nano ./backup.sh
GNU nano 5.6.1      ./backup.sh      Modified
echo "Bat dau sao lưu du lieu"
echo "....."
echo "Ngày giờ hệ thống"
date
echo "....."
filename="/tmp/$(date +%F).tar"
tar -cf $filename /home
echo "....."
echo "Thông tin du lieu sao lưu"
ls -l $filename
echo "Sao lưu thành công!!!!"

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify

```

1. Lệnh **date** cho phép ta in ra ngày giờ hiện tại
2. Đặt tên cho thư mục tập tin chuẩn bị nén **filename = "/tmp/\$(date +%F).tar"** (lệnh **date +%F** hiển thị theo dạng **yyyy-mm-dd**).
3. Tạo file nén **.tar** thư mục **home** với câu lệnh **tar -cf \$filename /home**.
4. Liệt kê thông tin đầy đủ của tập tin với **ls -l \$filename**.



```
B2111933@localhost:~
[B2111933@localhost ~]$ sudo chmod a+rx ./backup.sh
[B2111933@localhost ~]$ sudo ./backup.sh
Bat dau sao luu du lieu
.....
Ngày gio he thong
Mon Feb 27 08:19:39 PM +07 2023
.....
tar: Removing leading '/' from member names
.....
Thông tin du lieu sao luu
-rw-r--r--. 1 root root 790272000 Feb 27 20:19 /tmp/2023-02-27.tar
Sao luu thanh cong!!!!
[B2111933@localhost ~]$
```

Thay đổi quyền với **chmod** và chạy tập tin **./backup.sh** để xem kết quả.

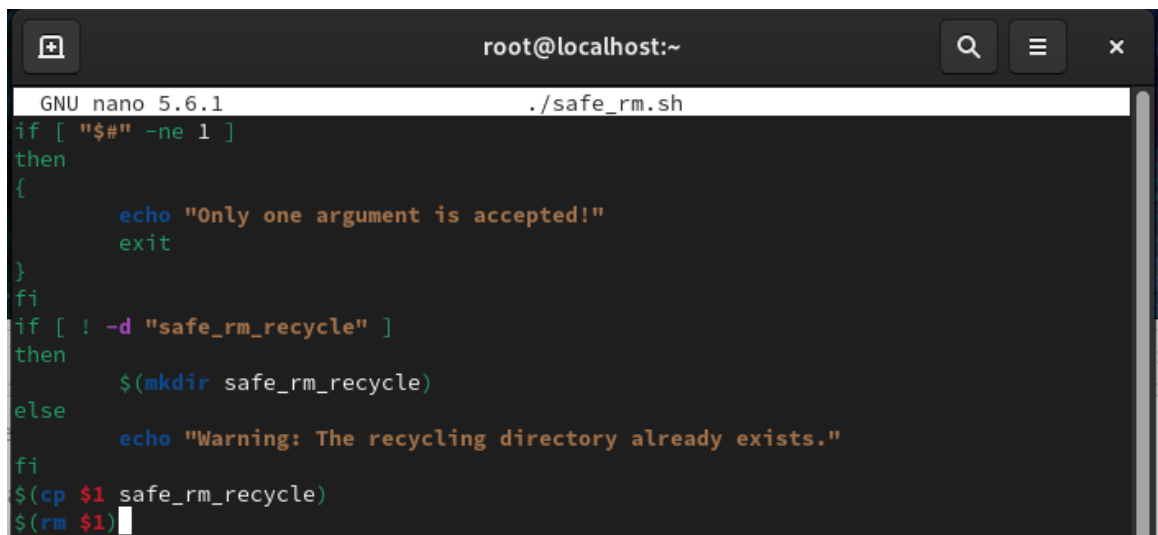
- 2.4.** Bổ sung dòng các lệnh cần thiết vào shell script `safe_rm.sh` bên dưới để thực hiện công việc sau:
- Nhận 01 tham số từ dòng lệnh. Thông báo lỗi và thoát khỏi chương trình nếu không có tham số hoặc nhiều hơn 01 tham số.
 - Tạo thư mục với tên "safe_rm_recycle" nếu chưa có. Sao chép tập tin với tên là tham số 01 vào thư mục "safe_rm_recycle" vừa tạo. Sau đó xóa tập tin.
 - Tạo tập tin có tên `mydata.txt`, thực thi `safe_rm.sh` để xóa tập tin `mydata.txt` (chụp hình minh họa).

`safe_rm.sh`

```
if [ "$#" -ne 1 ]
then
    echo "Only one argument is accepted!"
    #Thêm 1 dòng lệnh ở đây
fi
if [ ! -d "safe_rm_recycle" ]
then
    #Thêm 1 dòng lệnh ở đây
else
    echo "Warning: The recycling directory already
exists."
fi
#Thêm 1 dòng lệnh ở đây
#Thêm 1 dòng lệnh ở đây
```

Đáp án: `safe_rm.sh`

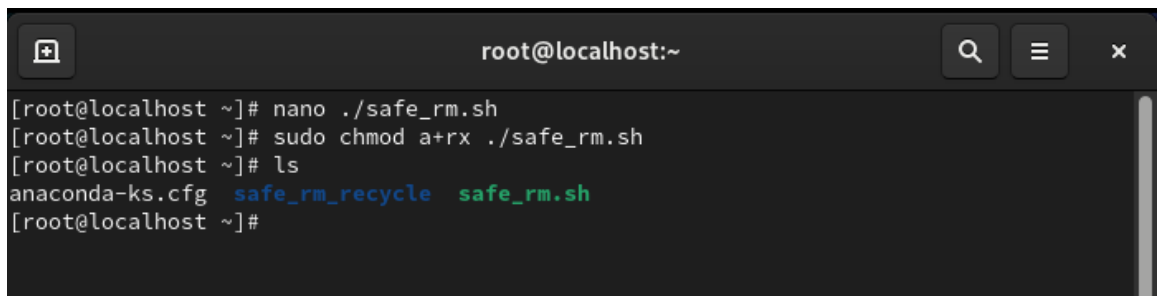
```
if [ "$#" -ne 1 ]
then
    echo "Only one argument is accepted!"
    exit
fi
if [ ! -d "safe_rm_recycle" ]
then
    $(mkdir safe_rm_recycle)
else
    echo "Warning: The recycling directory already
exists."
fi
$(cp $1 safe_rm_recycle)
$(rm $1)
```



root@localhost:~

GNU nano 5.6.1 ./safe_rm.sh

```
if [ "$#" -ne 1 ]
then
{
    echo "Only one argument is accepted!"
    exit
}
fi
if [ ! -d "safe_rm_recycle" ]
then
    $(mkdir safe_rm_recycle)
else
    echo "Warning: The recycling directory already exists."
fi
$(cp $1 safe_rm_recycle)
$(rm $1)
```



root@localhost:~

```
[root@localhost ~]# nano ./safe_rm.sh
[root@localhost ~]# sudo chmod a+rx ./safe_rm.sh
[root@localhost ~]# ls
anaconda-ks.cfg  safe_rm_recycle  safe_rm.sh
[root@localhost ~]#
```

Tạo tập tin **safe_rm.sh** với công cụ **nano** và chỉnh sửa quyền.

```
root@localhost:~  
[root@localhost ~]# ./safe_rm.sh dog monkey  
Only one argument is accepted!  
[root@localhost ~]# ./safe_rm.sh  
Only one argument is accepted!  
[root@localhost ~]# ./safe_rm.sh dog  
Warning: The recycling directory already exists.  
cp: cannot stat 'dog': No such file or directory  
rm: cannot remove 'dog': No such file or directory  
[root@localhost ~]#
```

Kiểm tra lại các câu lệnh.

```
root@localhost:~  
[root@localhost ~]# ls  
anaconda-ks.cfg  safe_rm_recycle  safe_rm.sh  
[root@localhost ~]# ls safe_rm_recycle  
[root@localhost ~]# touch mydata.txt  
[root@localhost ~]# ls  
anaconda-ks.cfg  mydata.txt  safe_rm_recycle  safe_rm.sh  
[root@localhost ~]# ls safe_rm_recycle  
[root@localhost ~]#
```

```
root@localhost:~  
[root@localhost ~]# ./safe_rm.sh mydata.txt  
Warning: The recycling directory already exists.  
[root@localhost ~]# ls  
anaconda-ks.cfg  safe_rm_recycle  safe_rm.sh  
[root@localhost ~]# ls safe_rm_recycle  
mydata.txt  
[root@localhost ~]#
```

Tạo tập tin có tên **mydata.txt**, thực thi **safe_rm.sh** để xóa tập tin **mydata.txt** và kiểm tra kết quả.

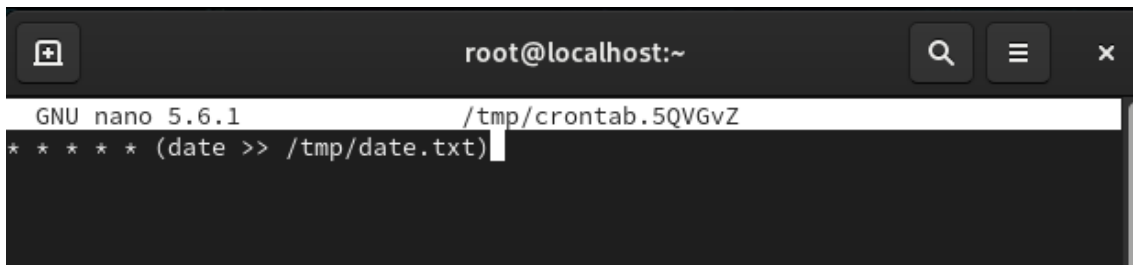
3. Lên lịch công việc định kỳ với cron

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
```

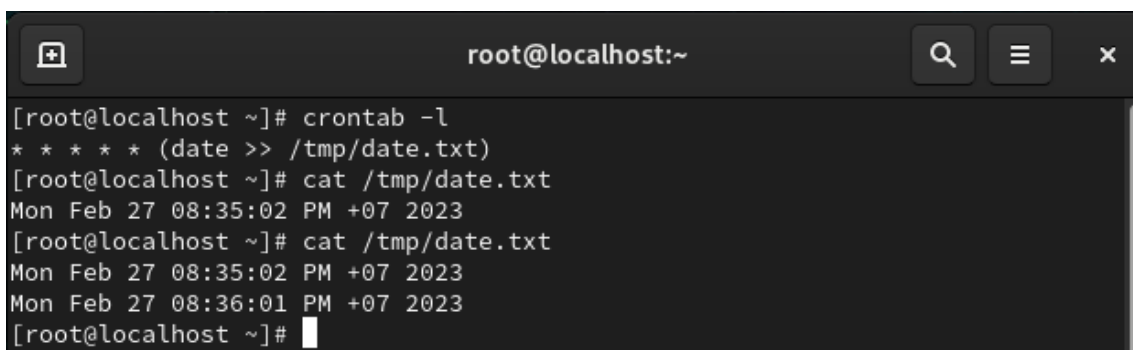
Cho biết cú pháp để thực hiện các yêu cầu sau từ crontab file:

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



```
root@localhost:~
GNU nano 5.6.1 /tmp/crontab.5QVGvZ
* * * * * (date >> /tmp/date.txt)
```

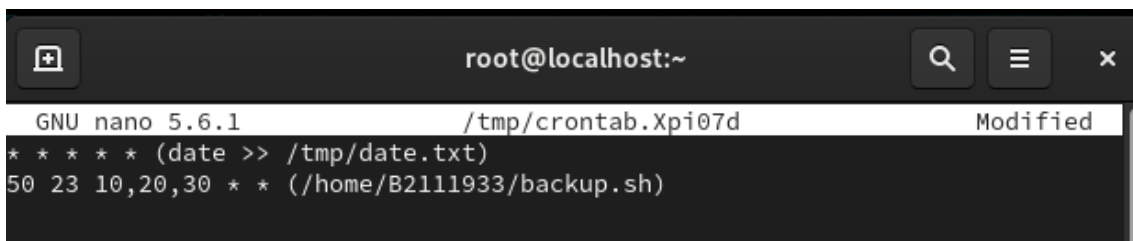
Để thực hiện **mỗi phút 1 lần**, ta bỏ qua **phút, giờ, ngày, tháng, thứ** tương ứng `* * * * *`
Ta dùng toán tử `>>` để xuất `date` ra tập tin `/tmp/date.txt`.



```
root@localhost:~
[root@localhost ~]# crontab -l
* * * * * (date >> /tmp/date.txt)
[root@localhost ~]# cat /tmp/date.txt
Mon Feb 27 08:35:02 PM +07 2023
[root@localhost ~]# cat /tmp/date.txt
Mon Feb 27 08:35:02 PM +07 2023
Mon Feb 27 08:36:01 PM +07 2023
[root@localhost ~]#
```

Thành quả sau mỗi phút.

- 3.2.** Thực thi `backup.sh` ở Câu 2.3 vào 23:50 giờ ngày 10, 20 và 30 hàng tháng (chụp hình minh họa).



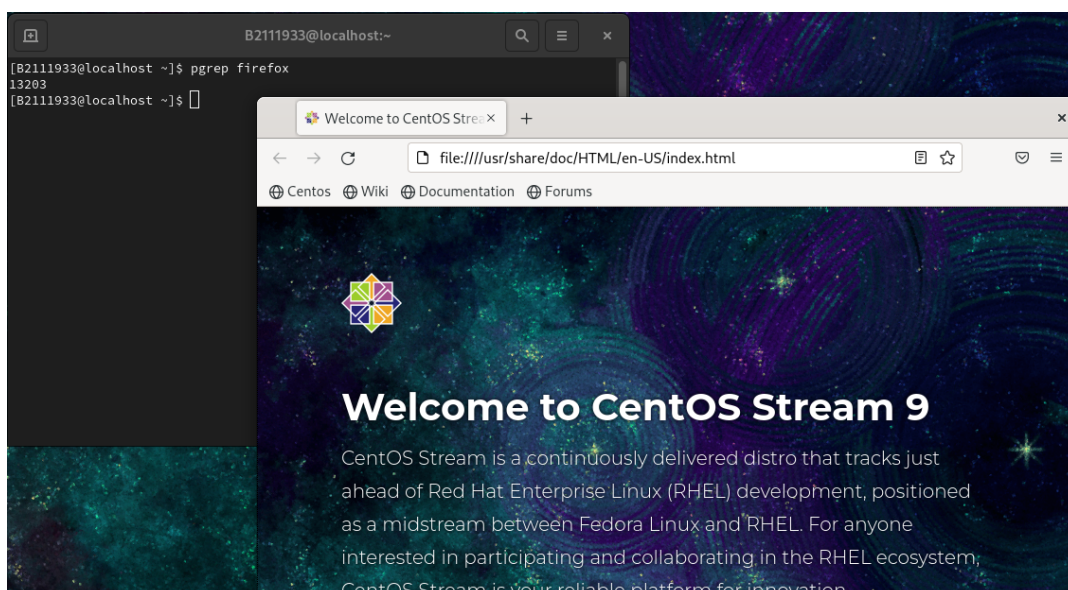
```
root@localhost:~
GNU nano 5.6.1 /tmp/crontab.Xpi07d Modified
* * * * * (date >> /tmp/date.txt)
50 23 10,20,30 * * (/home/B2111933/backup.sh)
```

Ta dùng lệnh `crontab -e` và thực hiện tương tự câu 3.1

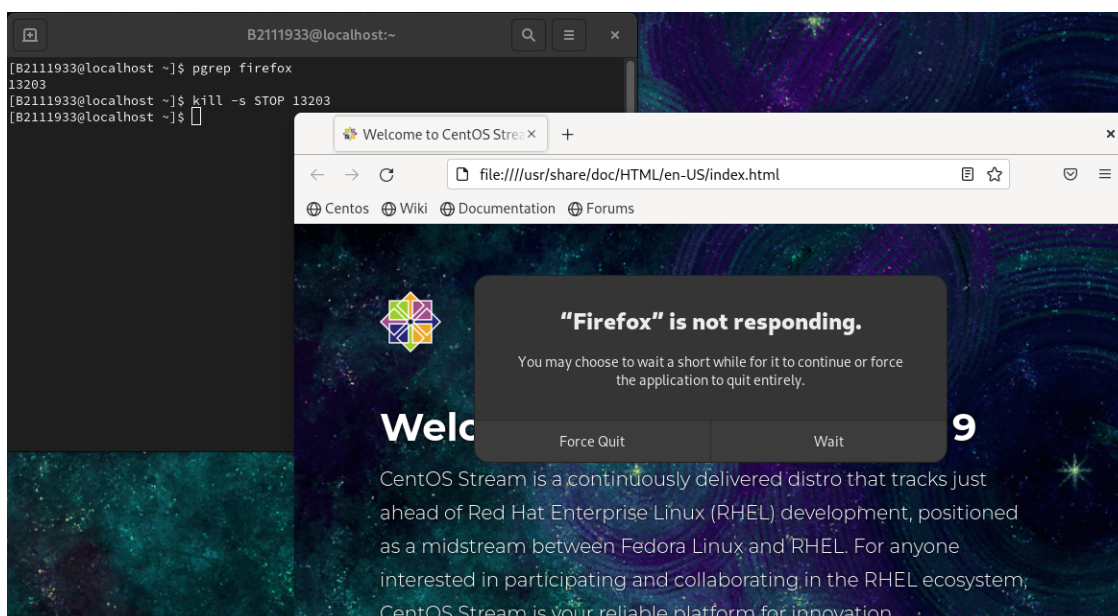
4. Thao tác với tiến trình

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 (chụp hình minh họa)?

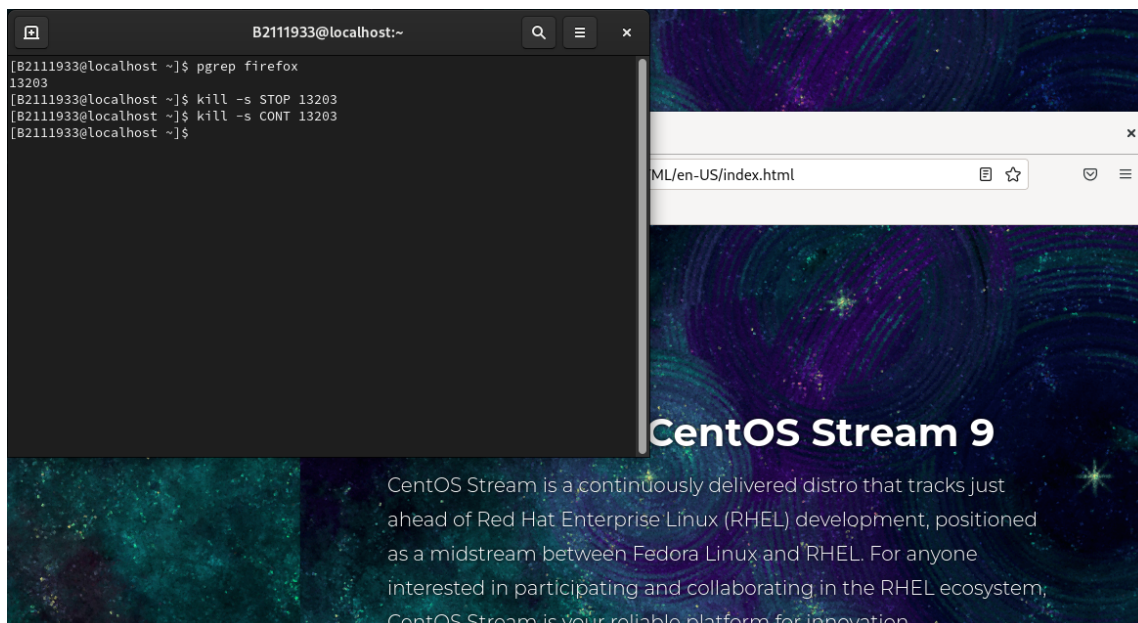


Ta dùng lệnh **pgrep** và tìm được id tiến trình **firefox** là **13203**.



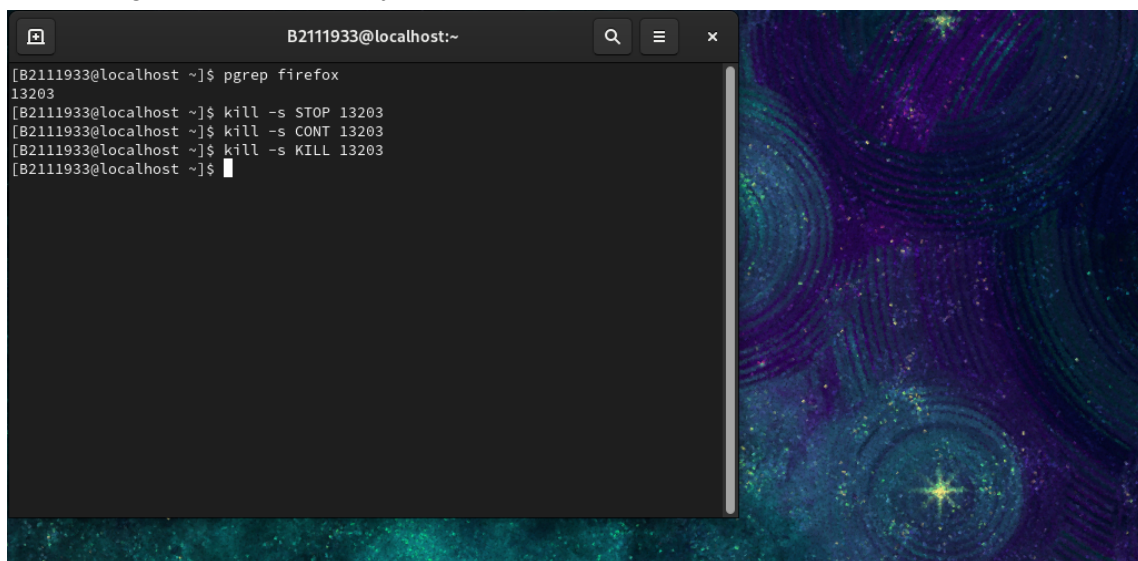
Dùng lệnh **kill -s STOP 13203** để dừng tiến trình **firefox**, lúc này ta không thể thao tác trên **firefox** được nữa.

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



Để tiếp tục tiến trình **firefox** ta dùng lệnh **kill -s CONT 13203**.

4.3. Dùng lệnh `kill` để hủy tiến trình `firefox` (chụp hình minh họa).



Ta dùng lệnh **kill -s KILL 13203** để hủy tiến trình **firefox**.

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).

```
root@localhost:~  
[root@localhost ~]# last  
B2111933 tty2          tty2          Mon Feb 27 11:18    still logged in  
B2111933 seat0         login screen   Mon Feb 27 11:18    still logged in  
reboot   system boot    5.14.0-252.el9.x Mon Feb 27 11:17    still running  
B2111933 tty2          tty2          Tue Feb  7 08:24 -  down (7+06:58)  
reboot   system boot    5.14.0-124.el9.x Tue Feb  7 08:16 - 15:22 (7+07:06)  
  
wtmp begins Tue Feb  7 08:16:24 2023  
[root@localhost ~]# last | head -n 1  
B2111933 tty2          tty2          Mon Feb 27 11:18    still logged in  
[root@localhost ~]#
```

Lệnh **last** dùng để hiển thị tất cả các lần đăng nhập.
Tham số | **head -n 1** dùng để lọc ra lần đăng nhập sau cùng.

5.2. Tạo một người dùng mới.

```
root@localhost:~  
[root@localhost ~]# adduser swimngu  
[root@localhost ~]#
```

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

```
root@localhost:~  
[root@localhost ~]# adduser swimngu  
[root@localhost ~]# less /var/log/secure
```

CentOS có một tập tin **log/secure** đặc biệt để chứa các sự kiện liên quan đến an ninh hệ thống.

```
root@localhost:~  
[root@localhost ~]# less /var/log/secure  
[1]+  Stopped                  less /var/log/secure  
[root@localhost ~]# cat /var/log/secure | grep swimngu  
Feb 27 21:04:25 localhost useradd[15148]: new group: name=swimngu, GID=1006  
Feb 27 21:04:25 localhost useradd[15148]: new user: name=swimngu, UID=1003, GID=1006,  
home=/home/swimngu, shell=/bin/bash, from=/dev/pts/0  
[root@localhost ~]#
```

Ta lấy thông tin người dùng **swimngu** bằng lệnh **cat /var/log/secure | grep swimngu**.
Có thể thấy người dùng **swimngu** được tạo ra vào **21:04:05** ngày **27/2** (thời gian máy ảo)

--- Hết ---