



LAB 3

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

Họ tên và MSSV: Đỗ Khánh Toàn - B2012046

Nhóm học phần: Nhóm 7

- 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: là chương trình được sử dụng để thiết lập hoặc hiển thị tên máy chủ, tên miền hoặc tên nút hiện tại của hệ thống **Linux**. Những tên này được sử dụng bởi các phần mềm network để xác định trên hệ thống.

```
b2012046@localhost:~  
[b2012046@localhost ~]$ hostname  
localhost.localdomain  
[b2012046@localhost ~]$
```

hostname -I: dùng để hiển thị tất cả địa chỉ IP cho tài khoản host

```
b2012046@localhost:~  
[b2012046@localhost ~]$ hostname  
localhost.localdomain  
[b2012046@localhost ~]$ hostname -I  
172.168.121.209  
[b2012046@localhost ~]$
```

Whoami: bắt nguồn từ cụm từ tiếng Anh "Who am I" (tôi là ai), có chức năng xuất ra tên tài khoản người dùng đang đăng nhập. Lệnh có chức năng tương tự với lệnh Unix `í-un`,... Ví dụ nếu người dùng đăng nhập tài khoản John và su và root, whoami sẽ hiển thị root còn echo `$USER` xuất ra John

```
[b2012046@localhost ~]$ whoami  
b2012046
```

df -H: kích thước in ở định dạng con người có thể đọc được và sử dụng công suất 1000 không phải 1024

```
[b2012046@localhost ~]$ df -H  
Filesystem      Size  Used Avail Use% Mounted on  
devtmpfs        4.2M   0    4.2M   0% /dev  
tmpfs           929M   0    929M   0% /dev/shm  
tmpfs           372M  13M   359M   4% /run  
/dev/mapper/cs-root 20G  4.9G   15G  26% /  
/dev/sda1       1.1G  298M   766M  28% /boot  
tmpfs           186M  103k   186M   1% /run/user/1000
```

ps -eo pid,%mem,%cpu,comm --sort -rss | head -n 3: dùng để hiển thị tiếng trình đầu tiên của việc sử dụng tài nguyên của hệ thống (mem, cpu) và sắp xếp theo chiều thử tự giảm dần

```
[b2012046@localhost ~]$ ps -eo pid,%mem,%cpu,comm --sort -rss | head -n 3  
PID %MEM %CPU COMMAND  
4189 15.4  3.0 gnome-shell  
1464 11.6  1.2 packagekitd  
[b2012046@localhost ~]$
```

(KHÔNG CÓ KHOẢNG TRẮNG SAU DẤU PHẢI)

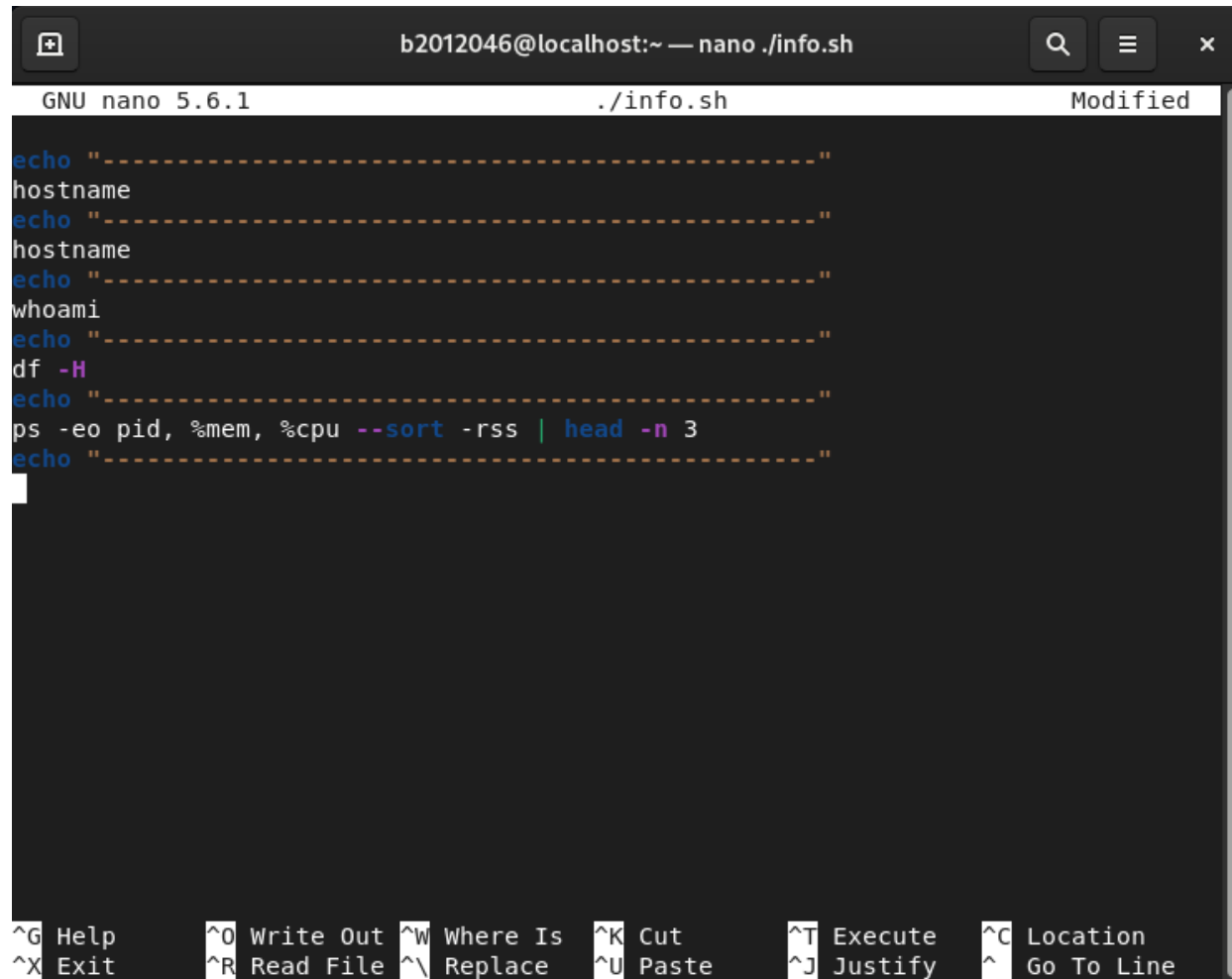
2.2. Viết shell script có tên `info.sh` thực hiện tất cả các lệnh ở 2.1 (chụp hình minh hoạ)

=>

1. TẠO SCRIPT

```
[b2012046@localhost ~]$ nano ./info.sh
```

2. THỰC HIỆN CÁC LỆNH SHELL Ở BÀI 2.1

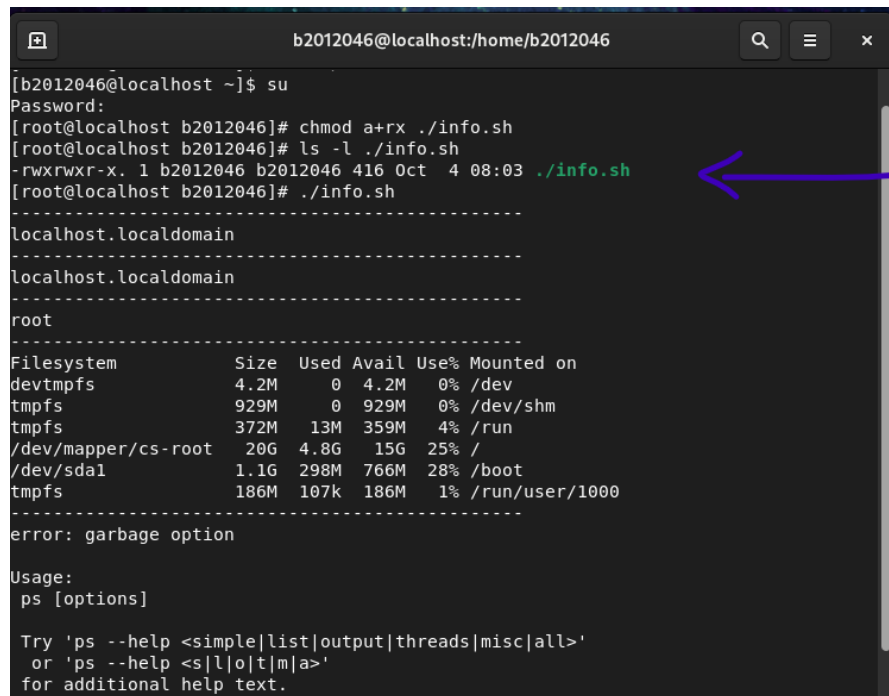


```
GNU nano 5.6.1                               ./info.sh                               Modified

echo "-----"
hostname
echo "-----"
hostname
echo "-----"
whoami
echo "-----"
df -H
echo "-----"
ps -eo pid, %mem, %cpu --sort -rss | head -n 3
echo "-----"

^G Help      ^O Write Out ^W Where Is   ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

3. Cấp quyền và chạy lệnh

A terminal window titled 'b2012046@localhost:/home/b2012046'. The user runs 'su' and enters a password. Then they run 'chmod a+rx ./info.sh'. Next, they run 'ls -l ./info.sh', which shows permissions '-rwxrwxr-x. 1 b2012046 b2012046 416 Oct 4 08:03 ./info.sh'. A blue arrow points to this line. Finally, they run './info.sh', which displays the output of the 'df' command. The output shows disk usage for various filesystems. Below the table, it says 'error: garbage option' and 'Usage: ps [options]'. At the bottom, it provides help text for the 'ps' command.

```
[b2012046@localhost ~]$ su
Password:
[root@localhost b2012046]# chmod a+rx ./info.sh
[root@localhost b2012046]# ls -l ./info.sh
-rwxrwxr-x. 1 b2012046 b2012046 416 Oct 4 08:03 ./info.sh
[root@localhost b2012046]# ./info.sh

-----
localhost.localdomain
-----
localhost.localdomain
-----
root
-----
Filesystem            Size  Used Avail Use% Mounted on
devtmpfs               4.2M   0   4.2M   0% /dev
tmpfs                  929M   0   929M   0% /dev/shm
tmpfs                  372M  13M   359M   4% /run
/dev/mapper/cs-root    20G   4.8G   15G   25% /
/dev/sda1              1.1G  298M   766M   28% /boot
tmpfs                  186M  107k   186M   1% /run/user/1000
-----
error: garbage option

Usage:
ps [options]

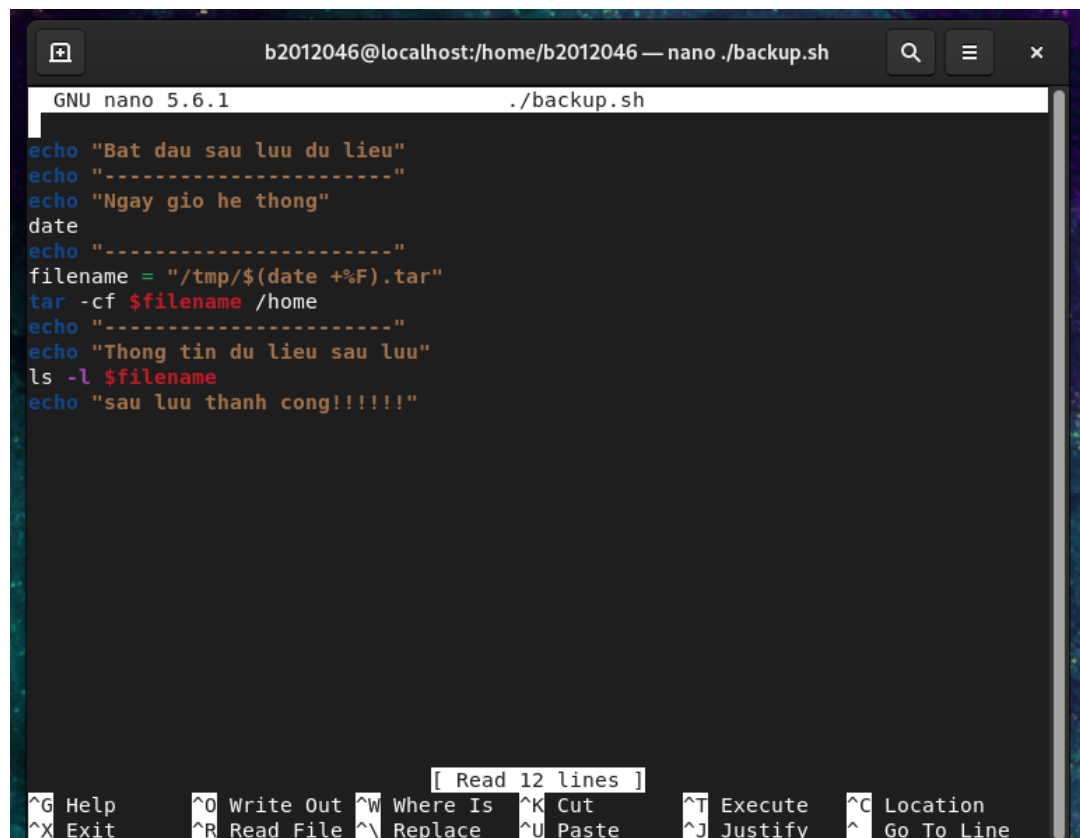
Try 'ps --help <simple|list|output|threads|misc|all>'
or 'ps --help <s|l|o|t|m|a>'
for additional help text.
```

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 `/tmp/<YYYY-MM-DD>.tar`
 - In thông báo "Sao lưu thành công!!!!"
- Thực thi `backup.sh` để kiểm tra (chụp hình minh họa).

=>

1. Tạo thư mục `backup.sh` và nội dung



```
b2012046@localhost:/home/b2012046 — nano ./backup.sh
GNU nano 5.6.1                               ./backup.sh
echo "Bat dau sau luu du lieu"
echo "-----"
echo "Ngay gio he thong"
date
echo "-----"
filename = "/tmp/${date +%F}.tar"
tar -cf $filename /home
echo "-----"
echo "Thong tin du lieu sau luu"
ls -l $filename
echo "sau luu thanh cong!!!!!"

[ Read 12 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

2. Chạy và cho kết quả:

```
[root@localhost b2012046]# nano ./backup.sh
[root@localhost b2012046]# chmod a+rx ./backup.sh
[root@localhost b2012046]# ./backup.sh
Bat dau sau luu du lieu
-----
Ngay gio he thong
Tue Oct  4 11:30:41 AM EDT 2022
-----
./backup.sh: line 7: filename: command not found
tar: Cowardly refusing to create an empty archive
Try 'tar --help' or 'tar --usage' for more information.
-----
Thong tin du lieu sau luu
total 8
-rwxr-xr-x. 1 root      root      294 Oct  4 11:28 backup.sh
drwxrwxr-x. 3 b2012046 b2012046  18 Sep 15 12:09 CT179
drwxr-xr-x. 2 b2012046 b2012046   6 Sep 15 10:16 Desktop
drwxr-xr-x. 2 b2012046 b2012046   6 Sep 15 10:16 Documents
drwxr-xr-x. 2 b2012046 b2012046  27 Sep 15 13:19 Downloads
-rwxrwxr-x. 1 b2012046 b2012046 416 Oct  4 08:03 info.sh
drwxr-xr-x. 2 b2012046 b2012046   6 Sep 15 10:16 Music
drwxr-xr-x. 2 b2012046 b2012046 100 Oct  4 07:58 Pictures
drwxr-xr-x. 2 b2012046 b2012046   6 Sep 15 10:16 Public
drwxr-xr-x. 2 b2012046 b2012046   6 Sep 15 10:16 Templates
drwxrwxr-x. 2 b2012046 b2012046   6 Sep 15 12:50 toanvippro
drwxr-xr-x. 2 b2012046 b2012046   6 Sep 15 10:16 Videos
sau luu thanh cong!!!!!!
[root@localhost b2012046]#
```

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).

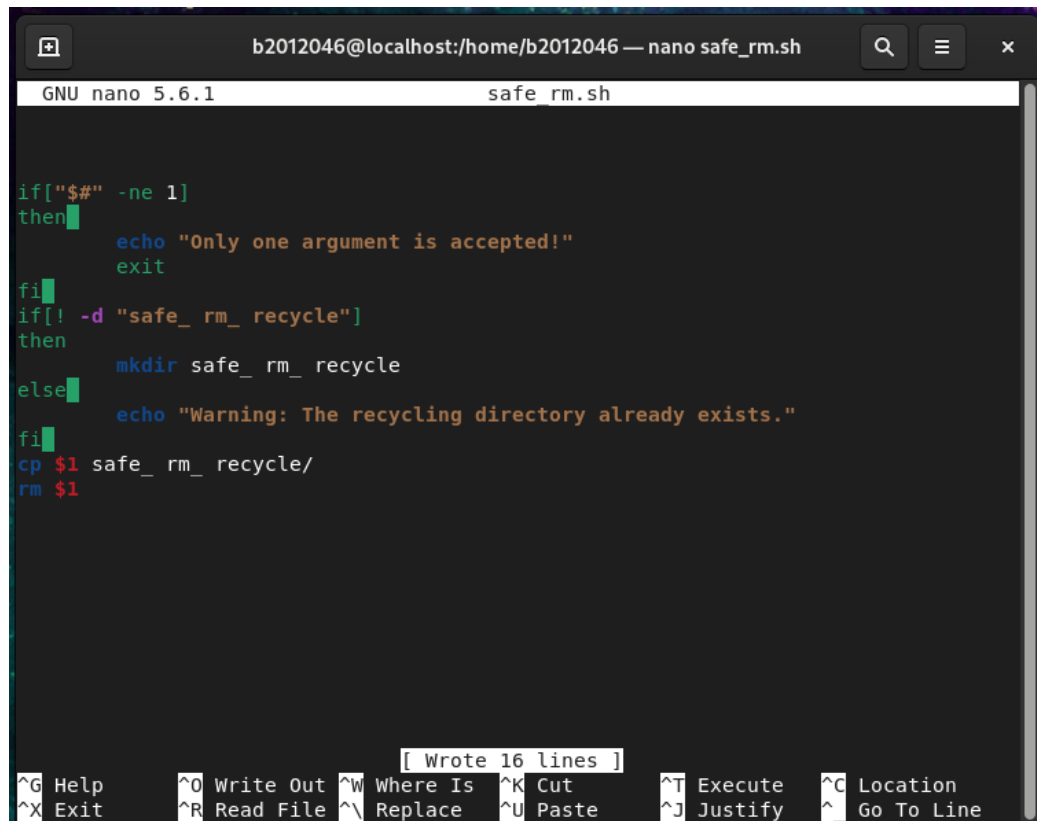
=>

`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
```

File safe_rm.sh



```
b2012046@localhost:/home/b2012046 — nano safe_rm.sh
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

[ Wrote 16 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

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:

```
root@localhost b2012046]# crontab -e
no crontab for root - using an empty one
crontab: no changes made to crontab
root@localhost b2012046]# export EDITOR=nan
root@localhost b2012046]# crontab -e
no crontab for root - using an empty one
crontab: no changes made to crontab
root@localhost b2012046]#
```

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)

Chạy lệnh `date` mỗi phút 1 lần và in ra bằng lệnh `crontab -l`

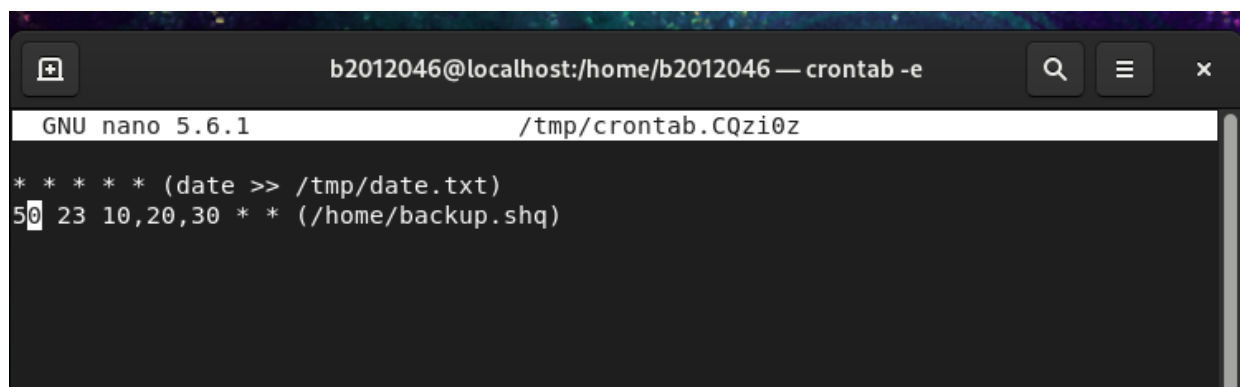
```
[root@localhost b2012046]# crontab -l
* * * * * (date >> /tmp/date.txt)
[root@localhost b2012046]#
```

In tập tin ngày giờ hệ thống, cứ mỗi phút tập tin được thêm 1 dòng

```
[root@localhost b2012046]# cat /tmp/date.txt
Tue Oct  4 12:26:01 PM EDT 2022
```

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).

Mở crontab -e: thực thi `backup.sh` 23:50 giờ ngày 10, 20 và 30 hàng tháng



```
b2012046@localhost:/home/b2012046 — crontab -e
GNU nano 5.6.1 /tmp/crontab.CQzi0z
* * * * * (date >> /tmp/date.txt)
50 23 10,20,30 * * (/home/backup.sh)
```

Kết quả


```
crontab: Installing new crontab
[root@localhost b2012046]# crontab -l

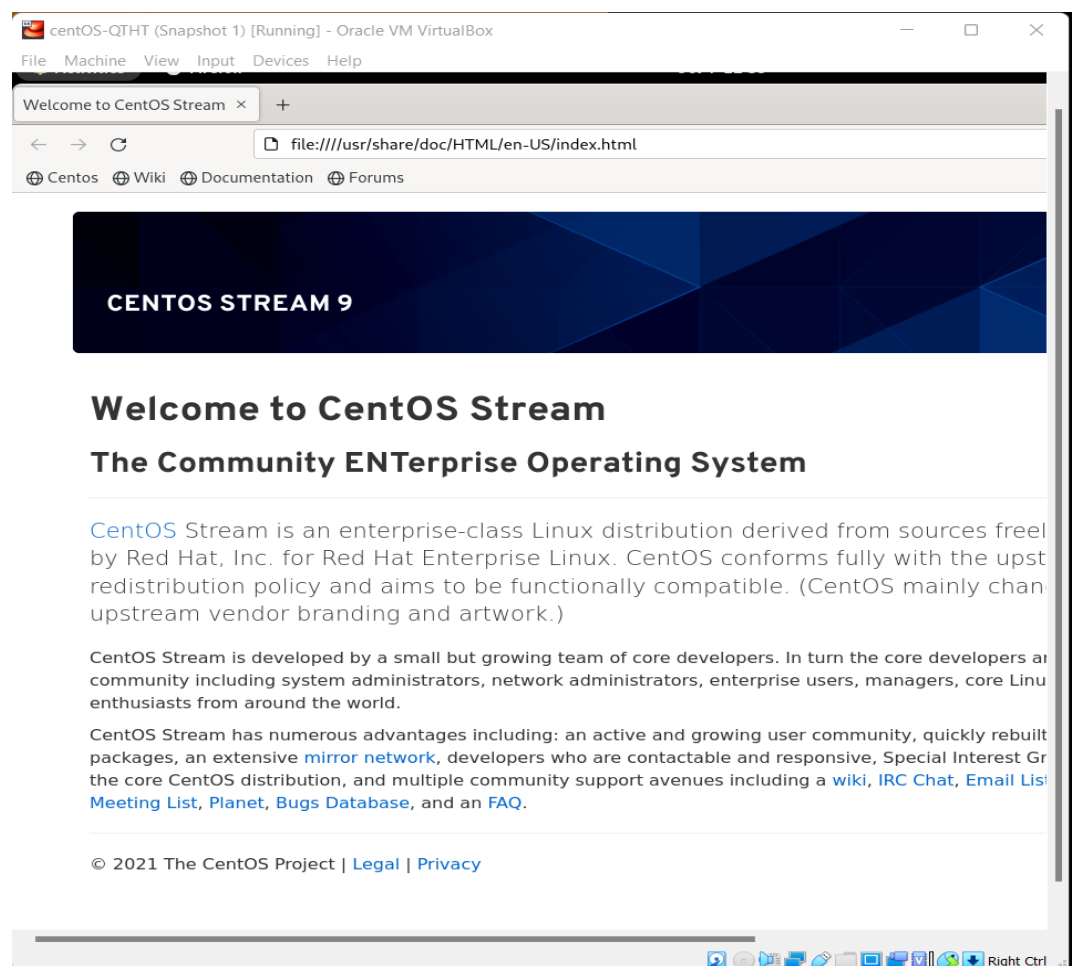
* * * * * (date >> /tmp/date.txt)
50 23 10,20,30 * * (/home/backup.sh)
[root@localhost b2012046]#
```

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)?

- Mở Firefox



- **Tìm mã số tiến trình:** mã số của tiến trình là 8308

```
[root@localhost b2012046]# pgrep firefox
8308
[root@localhost b2012046]#
```

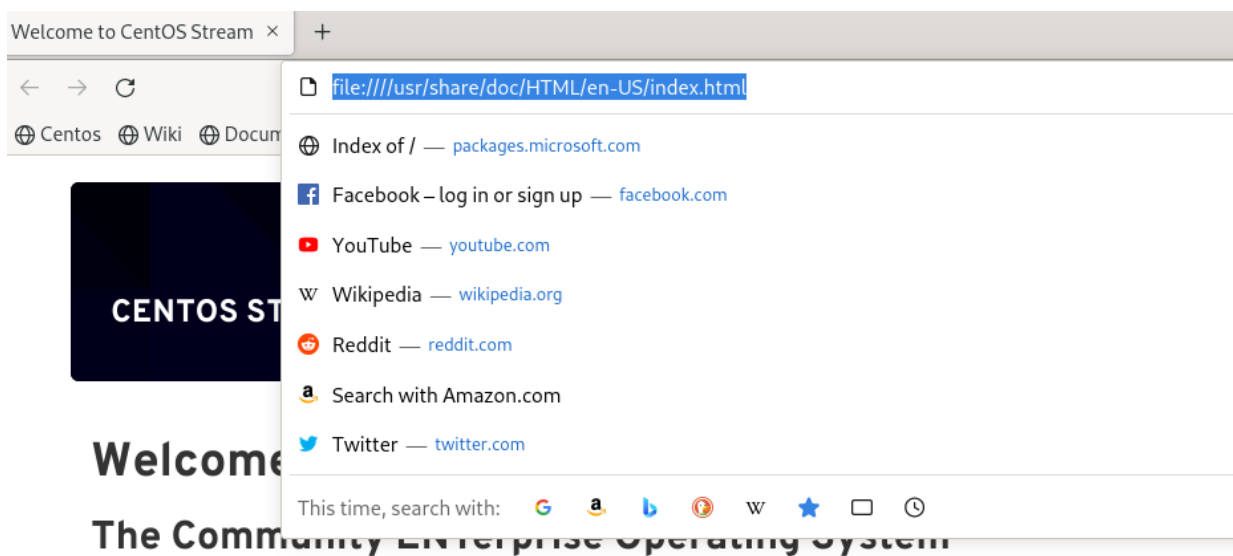
- Tạm dừng tiến trình với lệnh `kill` tham số `-s` lệnh cần gửi và mã số tiến trình

```
8308
[root@localhost b2012046]# kill -s STOP 8308
[root@localhost b2012046]#
```

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

- Tiếp tục tiến trình gửi tín hiệu `COUNT`

```
[root@localhost b2012046]# kill -s CONT 8308
[root@localhost b2012046]#
```



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

```

root      8654 16.0  2.7 360752 50348 ?        Sns1 12:40
root      8656  0.0  0.3 233424  6392 pts/0    R+    12:40
[root@localhost b2012046]# pgrep
pgrep: no matching criteria specified
Try `pgrep --help' for more information.
[root@localhost b2012046]# pgrep firefox
8308
[root@localhost b2012046]# kill -s STOP 8308
[root@localhost b2012046]# kill -s CONT 8308
[root@localhost b2012046]# kill -s STOP 8308
[root@localhost b2012046]# kill -s CONT 8308
[root@localhost b2012046]# kill -s KILL 8308
[root@localhost b2012046]#

```

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).

- Xem danh sách các file log trong folder `ls -l /var / log`

```

[root@localhost b2012046]# ls -l /var/log
total 1604
drwxr-xr-x. 2 root root 4096 Sep 15 10:13 anaconda
drwx----- 2 root root 23 Sep 15 10:15 audit
-rw----- 1 root root 2413 Oct 4 07:31 boot.log
-rw----- 1 root root 82765 Oct 1 01:56 boot.log-20221001
-rw----- 1 root root 16930 Oct 4 07:31 boot.log-20221004
-rw-rw---- 1 root utmp 384 Oct 1 02:00 btmp
-rw-rw---- 1 root utmp 0 Sep 15 10:05 btmp-20221001
drwxr-x--- 2 chrony chrony 6 Mar 24 2022 chrony
-rw----- 1 root root 9337 Oct 4 12:49 cron
-rw----- 1 root root 5963 Sep 19 07:01 cron-20221001
drwxr-xr-x. 2 lp sys 57 Sep 15 10:06 cups
-rw-r--r-- 1 root root 33346 Oct 4 12:48 dnf.librepo.log
-rw-r--r-- 1 root root 42436 Oct 4 12:48 dnf.log
-rw-r--r-- 1 root root 1006 Oct 4 12:40 dnf.rpm.log
-rw-r----- 1 root root 0 Sep 15 10:15 firewalld
drwx--x--x. 2 root gdm 6 Aug 3 15:55 gdm
-rw-r--r-- 1 root root 180 Oct 4 12:40 hawkey.log
-rw-r--r-- 1 root root 600 Sep 19 06:26 hawkey.log-20221001
-rw-r--r-- 1 root root 3936 Oct 4 07:31 kdump.log
-rw-rw-r-- 1 root utmp 292876 Oct 4 11:09 lastlog
-rw----- 1 root root 0 Oct 1 01:56 maillog
-rw----- 1 root root 0 Sep 15 10:06 maillog-20221001
-rw----- 1 root root 446639 Oct 4 12:49 messages
-rw----- 1 root root 787964 Oct 1 01:56 messages-20221001
drwx----- 2 root root 6 Sep 15 10:05 private
drwxr-xr-x. 2 root root 6 Aug 26 04:22 qemu-ga
lrwxrwxrwx. 1 root root 39 Sep 15 10:05 README -> ../../usr/share/doc/sys
temd/README.logs

```

```

b2012046@localhost:/home/b2012046
-rw----- 1 root root 5963 Sep 19 07:01 cron-20221001
drwxr-xr-x 2 lp sys 57 Sep 15 10:06 cups
-rw-r--r-- 1 root root 33346 Oct 4 12:48 dnf.librepo.log
-rw-r--r-- 1 root root 42436 Oct 4 12:48 dnf.log
-rw-r--r-- 1 root root 1006 Oct 4 12:40 dnf.rpm.log
-rw-r----- 1 root root 0 Sep 15 10:15 firewalld
drwx--x--x 2 root gdm 6 Aug 3 15:55 gdm
-rw-r--r-- 1 root root 180 Oct 4 12:40 hawkey.log
-rw-r--r-- 1 root root 600 Sep 19 06:26 hawkey.log-20221001
-rw-r--r-- 1 root root 3936 Oct 4 07:31 kdump.log
-rw-rw-r-- 1 root utmp 292876 Oct 4 11:09 lastlog
-rw----- 1 root root 0 Oct 1 01:56 maillog
-rw----- 1 root root 0 Sep 15 10:06 maillog-20221001
-rw----- 1 root root 446639 Oct 4 12:49 messages
-rw----- 1 root root 787964 Oct 1 01:56 messages-20221001
drwx----- 2 root root 6 Sep 15 10:05 private
drwxr-xr-x 2 root root 6 Aug 26 04:22 qemu-ga
lrwxrwxrwx 1 root root 39 Sep 15 10:05 README -> ../../usr/share/doc/sys
temd/README.logs
drwx----- 3 root root 17 Sep 15 10:05 samba
-rw----- 1 root root 25693 Oct 4 12:49 secure
-rw----- 1 root root 18879 Sep 19 07:48 secure-20221001
drwx----- 2 root root 6 Aug 10 2021 speech-dispatcher
-rw----- 1 root root 0 Oct 1 01:56 spooler
-rw----- 1 root root 0 Sep 15 10:06 spooler-20221001
drwxr-x-- 2 sssd sssd 55 Oct 1 01:56 sssd
-rw----- 1 root root 0 Sep 15 10:05 tallylog
drwxr-xr-x 2 root root 23 Sep 19 04:29 tuned
-rw-rw-r-- 1 root utmp 9600 Oct 4 07:47 wtmp
[root@localhost b2012046]#

```

Đọc tập tin sử dụng lệnh `last` hiện tất cả các lần đăng nhập của người dùng, do tập tin dưới dạng nhị phân

```

[root@localhost b2012046]# last
b2012046 tty2      tty2          Tue Oct 4 07:47  still logged in
toan.do tty2      tty2          Tue Oct 4 07:32 - 07:46 (00:14)
reboot system boot 5.14.0-160.el9.x Tue Oct 4 07:31  still running
toan.do tty2      tty2          Sat Oct 1 01:56 - down (03:15)
reboot system boot 5.14.0-160.el9.x Sat Oct 1 01:55 - 05:11 (03:15)
b2012046 tty2      tty2          Mon Sep 19 04:30 - down (03:17)
reboot system boot 5.14.0-160.el9.x Mon Sep 19 04:29 - 07:48 (03:18)
reboot system boot 5.14.0-160.el9.x Sat Sep 17 04:54 - 07:48 (2+02:54)
b2012046 tty2      tty2          Thu Sep 15 10:38 - down (02:46)
reboot system boot 5.14.0-160.el9.x Thu Sep 15 10:37 - 13:24 (02:46)
b2012046 tty2      tty2          Thu Sep 15 10:25 - down (00:00)
b2012046 tty2      tty2          Thu Sep 15 10:16 - 10:24 (00:08)
reboot system boot 5.14.0-160.el9.x Thu Sep 15 10:15 - 10:25 (00:10)

wtmp begins Thu Sep 15 10:15:05 2022
[root@localhost b2012046]#

```

- Hiện thị lần đăng nhập lần cuối cùng sử dụng lệnh `head` với tham số `-n=1`: in người dùng cuối cùng đăng nhập vào hệ thống

```
[root@localhost b2012046]# last | head -n 1
b2012046 tty2          tty2          Tue Oct  4 07:47    still logged in
[root@localhost b2012046]#
```

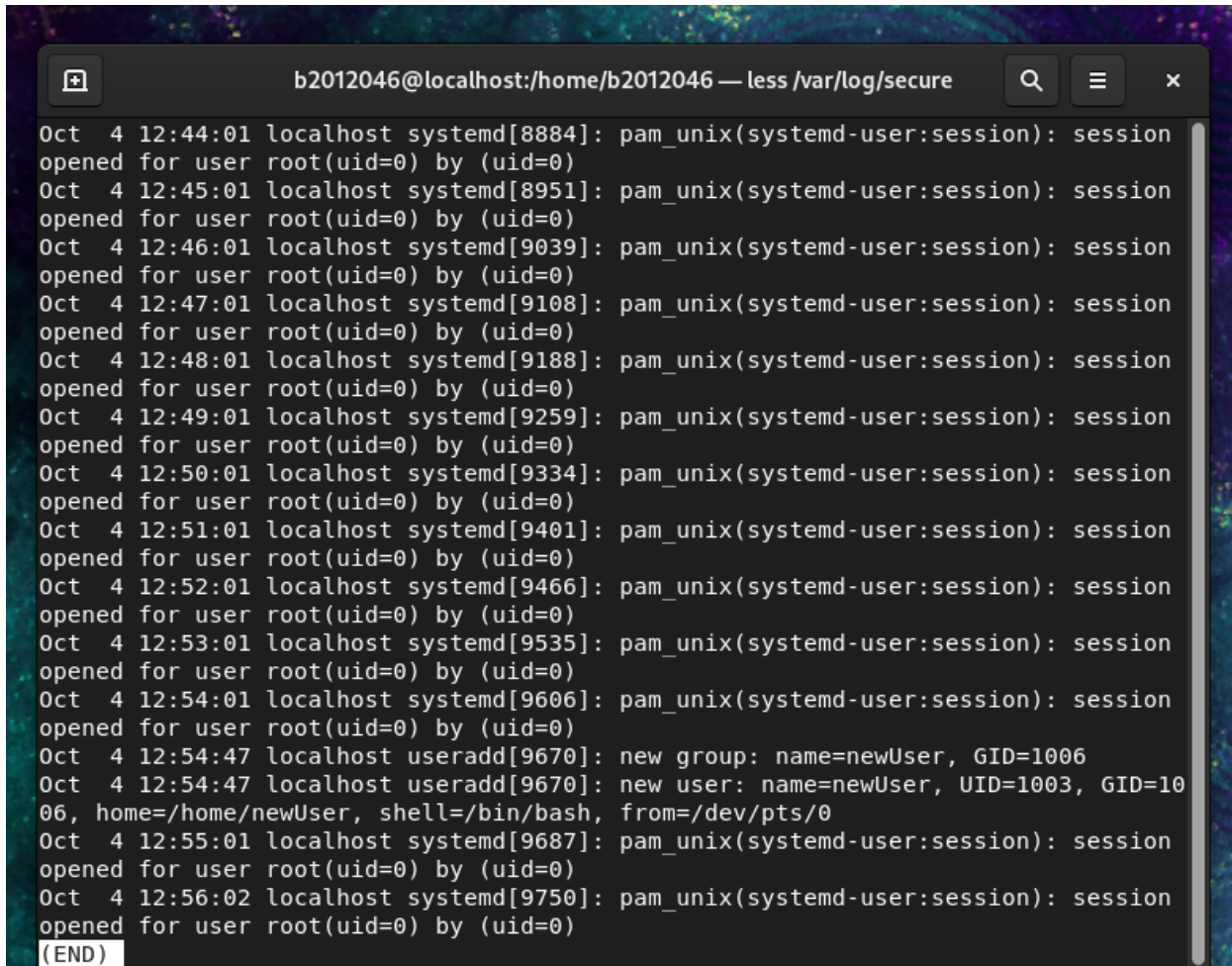
5.2. Tạo một người dùng mới.

- **Tạo người dùng mới bằng lệnh adduser**

```
[root@localhost b2012046]# adduser newUser
[root@localhost b2012046]#
```

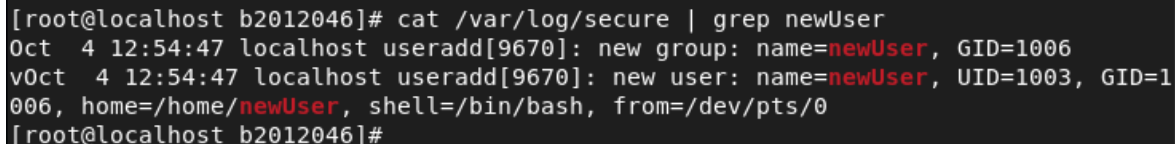
- Trong tập tin log có tập tin chứa các sự kiện liên quan an ninh hệ thống (**secure**)

```
b2012046@localhost:/home/b2012046
[root@localhost b2012046]# adduser newUser
[root@localhost b2012046]# less /var/log/secure
```



```
b2012046@localhost:/home/b2012046 — less /var/log/secure
Oct  4 12:44:01 localhost systemd[8884]: pam_unix(systemd-user:session): session
opened for user root(uid=0) by (uid=0)
Oct  4 12:45:01 localhost systemd[8951]: pam_unix(systemd-user:session): session
opened for user root(uid=0) by (uid=0)
Oct  4 12:46:01 localhost systemd[9039]: pam_unix(systemd-user:session): session
opened for user root(uid=0) by (uid=0)
Oct  4 12:47:01 localhost systemd[9108]: pam_unix(systemd-user:session): session
opened for user root(uid=0) by (uid=0)
Oct  4 12:48:01 localhost systemd[9188]: pam_unix(systemd-user:session): session
opened for user root(uid=0) by (uid=0)
Oct  4 12:49:01 localhost systemd[9259]: pam_unix(systemd-user:session): session
opened for user root(uid=0) by (uid=0)
Oct  4 12:50:01 localhost systemd[9334]: pam_unix(systemd-user:session): session
opened for user root(uid=0) by (uid=0)
Oct  4 12:51:01 localhost systemd[9401]: pam_unix(systemd-user:session): session
opened for user root(uid=0) by (uid=0)
Oct  4 12:52:01 localhost systemd[9466]: pam_unix(systemd-user:session): session
opened for user root(uid=0) by (uid=0)
Oct  4 12:53:01 localhost systemd[9535]: pam_unix(systemd-user:session): session
opened for user root(uid=0) by (uid=0)
Oct  4 12:54:01 localhost systemd[9606]: pam_unix(systemd-user:session): session
opened for user root(uid=0) by (uid=0)
Oct  4 12:54:47 localhost useradd[9670]: new group: name=newUser, GID=1006
Oct  4 12:54:47 localhost useradd[9670]: new user: name=newUser, UID=1003, GID=10
06, home=/home/newUser, shell=/bin/bash, from=/dev/pts/0
Oct  4 12:55:01 localhost systemd[9687]: pam_unix(systemd-user:session): session
opened for user root(uid=0) by (uid=0)
Oct  4 12:56:02 localhost systemd[9750]: pam_unix(systemd-user:session): session
opened for user root(uid=0) by (uid=0)
(END)
```

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



```
[root@localhost b2012046]# cat /var/log/secure | grep newUser
Oct  4 12:54:47 localhost useradd[9670]: new group: name=newUser, GID=1006
vOct  4 12:54:47 localhost useradd[9670]: new user: name=newUser, UID=1003, GID=1
006, home=/home/newUser, shell=/bin/bash, from=/dev/pts/0
[root@localhost b2012046]#
```

--- Hết ---