

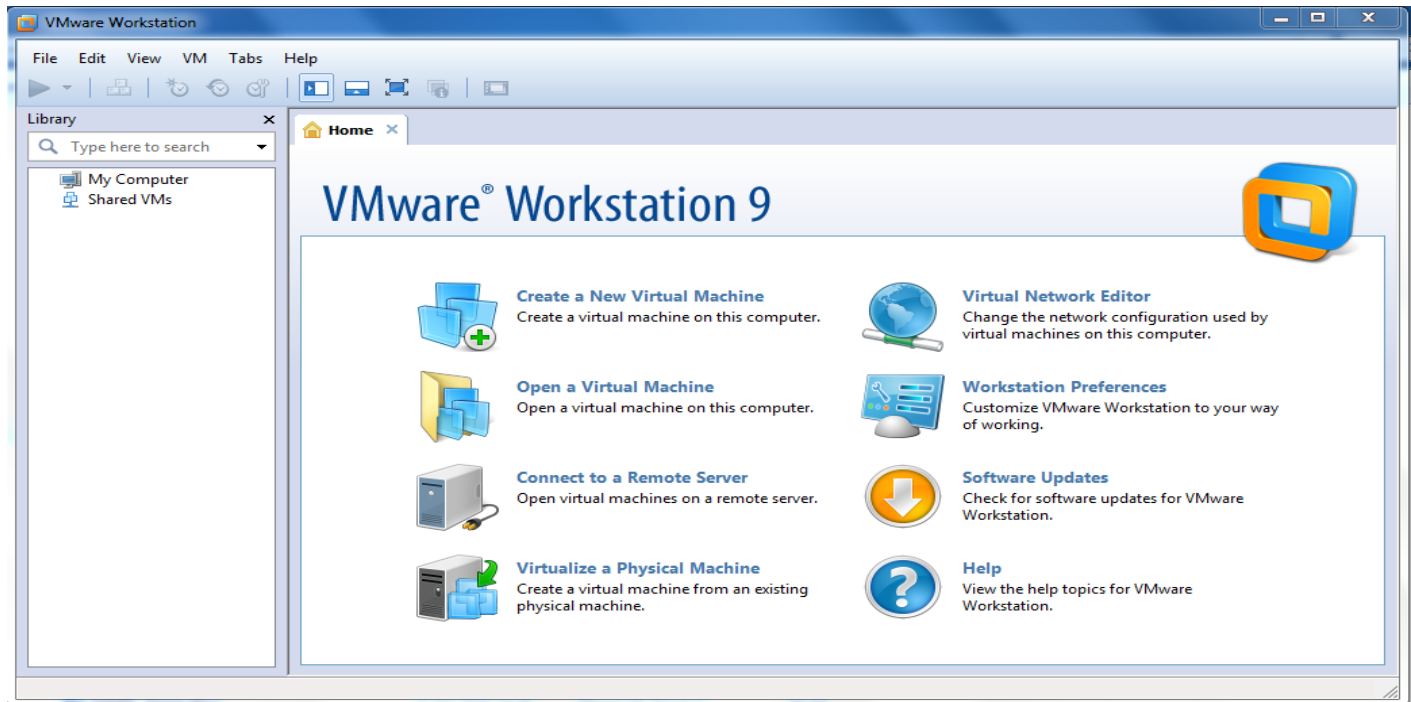
BÀI 3. QUẢN TRỊ TÀI NGUYÊN VÀ DỊCH VỤ

LƯU Ý: TẤT CẢ SINH VIÊN THỰC HIỆN TRÊN FILE IMAGE CỦA HỆ ĐIỀU HÀNH CENTOS 7 ĐÃ CÀI TỪ CÁC BUỔI THỰC HÀNH TRƯỚC TẠI THƯ MỤC ĐÃ TẠO TẠI Ổ SINHVIEN

PHẦN 1: MỞ FILE IMAGE CỦA HỆ ĐIỀU HÀNH CENTOS 7

❖ Bước 1: Khởi động phần mềm máy ảo VMware

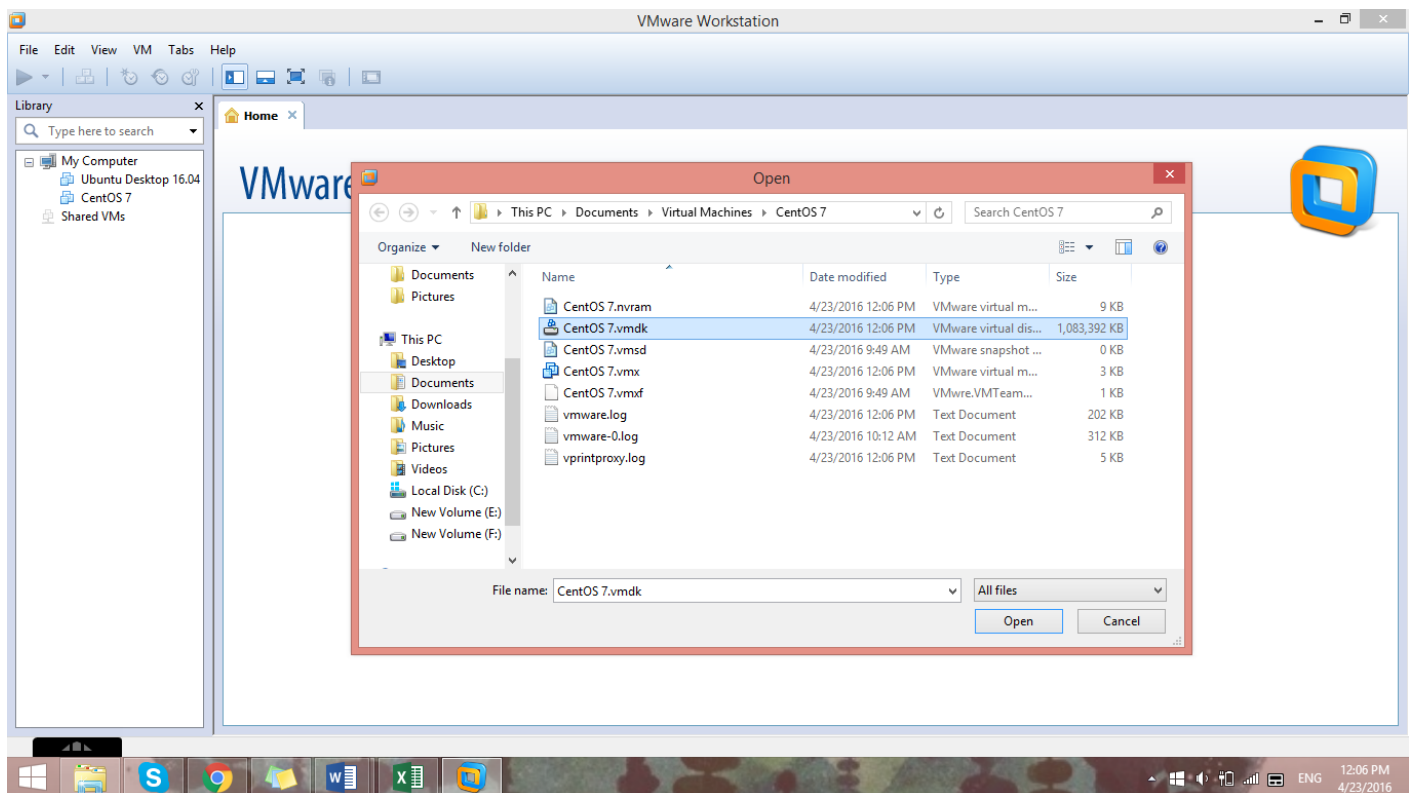
- Sau khi khởi động VMware như hình sau



❖ Bước 2: Mở file image của hệ điều hành CentOS 7 đã cài

Trên màn hình chính VMware chọn File ->Open...

Chọn đến file image VMX của CentOS 7



❖ Bước 3: Khởi động vào máy ảo CentOS 7

Chuột phải vào máy ảo chọn Power -> Power On

```
CentOS Linux 7 (AltArch)
Kernel 3.10.0-327.el7.i686 on an i686
localhost login: _
```

PHẦN 2: CÔNG CỤ QUẢN LÝ TÀI NGUYÊN – PHÂN TÍCH HOẠT ĐỘNG

Quản lý tiến trình:

Linux là hệ điều hành đa nhiệm, do đó việc quản lý tiến trình rất quan trọng.

Terminal của Linux cung cấp một số lượng các câu lệnh khá hữu ích hỗ trợ việc quản lý các tiến trình, kill chúng, hoặc nâng độ ưu tiên cho chúng.

- Để liệt kê các tiến trình đang chạy trên hệ điều hành dùng lệnh: ps
 - e: hiển thị thông tin về mỗi tiến trình.
 - l: hiển thị thông tin đầy đủ tiến trình.
 - f: hiển thị thông tin về tiến trình cha.
 - A: hiển thị tất cả các tiến trình.

```

[root@localhost ~]# ps -l
F S  UID  PID  PPID  C  PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S   0  1942  600  0  80   0 -  1658 wait  tty1        00:00:00 bash
4 S   0  1986  1942  0  80   0 -  2232 wait  tty1        00:00:00 su
4 S   0  2098  1987  0  80   0 -  2206 wait  tty1        00:00:00 su
4 S   0  2102  2098  0  80   0 -  1657 wait  tty1        00:00:00 bash
0 R   0  2144  2102  0  80   0 -  1917 -      tty1        00:00:00 ps
[root@localhost ~]#

```

Dưới đây là sự miêu tả của tất cả các file được hiển thị bởi lệnh ps -f.

Cột	Miêu tả
UID	ID người sử dụng mà tiến trình này thuộc sở hữu (người chạy nó).
PID	Process ID.
PPID	Process ID gốc (ID của tiến trình mà bắt đầu nó).
C	CPU sử dụng của tiến trình.
STIME	Thời gian bắt đầu tiến trình.
TTY	Kiểu terminal liên kết với tiến trình.
TIME	Thời gian CPU bị sử dụng bởi tiến trình.
CMD	Lệnh mà bắt đầu tiến trình này.

- Câu lệnh **top** là một phương thức để xem việc chiếm dụng tài nguyên hệ thống, và xem những tiến trình nào chiếm dụng tài nguyên nhiều nhất. Những tiến trình chiếm dụng CPU nhiều nhất sẽ được liệt kê đầu tiên.

```
top - 01:50:53 up 2:45, 1 user, load average: 0.00, 0.01, 0.05
Tasks: 84 total, 2 running, 82 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 1026532 total, 874012 free, 53732 used, 98788 buff/cache
KiB Swap: 2097148 total, 2097148 free, 0 used. 936344 avail Mem
```

PID	USER	PR	NI	UIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2069	root	20	0	0	0	0	S	1.7	0.0	0:04.00	kworker/0:1
2072	nvdong	20	0	8092	1712	1216	R	0.3	0.2	0:00.10	top
1	root	20	0	7908	5180	2464	S	0.0	0.5	0:02.91	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/0
7	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
9	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcuob/0
10	root	20	0	0	0	0	S	0.0	0.0	0:00.67	rcu_sched
11	root	20	0	0	0	0	R	0.0	0.0	0:00.55	rcuos/0
12	root	rt	0	0	0	0	S	0.0	0.0	0:00.45	watchdog/0
13	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	khelper
14	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
15	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	netns
16	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	perf
17	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	writeback
18	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kintegrityd
19	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	bioaset
20	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kblockd
21	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	md
24	root	20	0	0	0	0	S	0.0	0.0	0:00.00	khungtaskd
25	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kswapd0
26	root	25	5	0	0	0	S	0.0	0.0	0:00.00	ksmd
27	root	20	0	0	0	0	S	0.0	0.0	0:00.00	fsnotify_mark
28	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	crypto
36	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kthrotld
37	root	20	0	0	0	0	S	0.0	0.0	0:00.44	kworker/u2:1

- Để thoát khỏi **top**, ấn tổ hợp phím tắt **Ctrl + C**. Tổ hợp phím này cũng dùng để kill các ứng dụng đang chạy từ Terminal.
- Khi liệt kê danh sách tiến trình đang chạy. Nếu cảm thấy danh sách có vẻ quá dài, có thể dùng lệnh sau đây để hiển thị một danh sách ngắn hơn:

```
ps -A | less
```

- Ấn “q” để thoát.
- Ngoài ra, có thể chỉ định tiến trình cụ thể bằng lệnh **grep**. Ví dụ sau đây sẽ tìm kiếm tiến trình **ttm_swap** trên hệ thống:

```
ps -A | grep ttm_swap
```

```
[root@localhost ~]# ps -A|grep ttm_swap
285 ?        00:00:00 ttm_swap
[root@localhost ~]#
```

- **kill**

Câu lệnh này dùng để tắt các tiến trình, bằng cách cung cấp PID của chúng. PID này có thể được lấy từ lệnh **ps -A** hoặc **pgrep**

```
kill PID
```

```
[root@localhost ~]# kill 1942
[root@localhost ~]#
```

- Để kill các ứng dụng cứng đầu, dùng lệnh **kill -KILL** hoặc **kill -9**.
- **pgrep**
Lệnh này cho phép bạn tìm PID của một tiến trình trên hệ thống. Ví dụ, muốn tìm PID của `ttm_swap`:

- **pgrep ttm_swap**
- Cũng có thể kết hợp nó với lệnh `kill` ở trên.

```
[root@localhost ~]# pgrep ttm_swap
285
[root@localhost ~]#
```

- **pkill & killall**
- Hai câu lệnh này cho phép kill tiến trình bằng cách cung cấp tên của chúng:

```
pkill tuned
```

```
killall tuned
```

Sau khi kill tiến trình `tuned` xong nó không còn xuất hiện nữa

```
[root@localhost ~]# ps -A | grep tuned
795 ?        00:00:02 tuned
[root@localhost ~]# pkill tuned
[root@localhost ~]# ps -A | grep tuned
[root@localhost ~]#
```

- **renice**
Câu lệnh **renice** được dùng để thay đổi giá trị ưu tiên của tiến trình. Mặc định giá trị ưu tiên là 0, giá trị -19 được ưu tiên rất cao, ngược lại 19 lại có độ ưu tiên thấp nhất.
- Câu lệnh **renice** yêu cầu cung cấp PID của tiến trình. Lệnh sau đây sẽ thay đổi priority của tiến trình xuống mức thấp nhất:

```
renice 19 PID
```

```
[root@localhost ~]# pgrep sshd
798
[root@localhost ~]# renice 19 798
798 (process ID) old priority 0, new priority 19
[root@localhost ~]#
```

- Nếu muốn cấp quyền ưu tiên cao hơn cho một tiến trình, bạn cần dùng quyền của root:

```
sudo renice -19 PID
```

- **xkill**
- **xkill** là một cách dễ dàng để kill các chương trình đồ họa đang chạy. Khi bạn gõ lệnh này, con trỏ chuột sẽ biến thành chữ x, để kill ứng dụng nào đang chạy, click vào ứng dụng đó. Nếu bạn không muốn kill ứng dụng, click chuột phải, lúc này sẽ thoát xkill.
- Cách đơn giản hơn để chạy xkill là ấn tổ hợp phím **Alt + F2** , gõ vào xkill , ấn Enter và click vào ứng dụng cần kill.
- Kiểm tra dung lượng các ổ đĩa dùng lệnh df:

```
[root@localhost ~]# df
Filesystem      1k-blocks    Used Available Use% Mounted on
/dev/mapper/cl-root 18307072 887244 17419828 5% /
devtmpfs         503936      0    503936 0% /dev
tmpfs            513264      0    513264 0% /dev/shm
tmpfs            513264     6948    506316 2% /run
tmpfs            513264      0    513264 0% /sys/fs/cgroup
/dev/sda1        508588    121776    386812 24% /boot
tmpfs            102656      0    102656 0% /run/user/0
```

- Để xem dung lượng bộ nhớ chính hiện tại sử dụng lệnh: free -m

```
[root@localhost ~]# free -m
              total        used         free       shared    buff/cache   available
Mem:           1002          46           854           6         101         919
Swap:          2047           0          2047
```

PHẦN 3: DỊCH VỤ CƠ BẢN CỦA LINUX VÀ PIPE

Cơ chế đường ống Pipe

Cơ chế đường ống giữa hai tiến trình cho phép định hướng lại đầu ra của tiến trình thứ nhất trở thành đầu vào của tiến trình thứ hai

Cơ chế đường ống được thiết lập bằng cách sử dụng ký tự: |

Cú pháp có dạng: cmd1 | cmd2

- Lọc thông tin từ file: \$cat tên_file | grep nội_dung_cần_lọc

```
[root@localhost ~]# cat anaconda-ks.cfg | grep root
rootpw --iscrypted $6$u2ayb59JrTS1PdS1$Xpw5gd12KZKE7vST8q0MTu/kovu4jCs3PaDhrWzxcqzKX8t7kSzNf5onNxpU1aPNO4YU1ZDRKqXHUP4B4U1Se/
[root@localhost ~]#
```

- Hiển thị nội dung n dòng đầu tiên trên file : \$head -n tên_file

```
[root@localhost ~]# head -10 anaconda-ks.cfg
#version=DEVEL
# System authorization information
auth --enableshadow --passalgo=sha512
# Use CDROM installation media
cdrom
# Use graphical install
graphical
# Run the Setup Agent on first boot
firstboot --enable
# Keyboard layouts
```

- Hiển thị n dòng cuối cùng của 1 file: \$tail -n tên_file

```
[root@localhost ~]# tail -10 anaconda-ks.cfg
zend
%addon org_fedora_oscap
  content-type = scap-security-guide
  profile = default
zend
%addon com_redhat_kdump --enable --reserve-mb='auto'
zend
[root@localhost ~]#
```


- Đếm số dòng trên file: `$wc tên_file`

```
[root@localhost ~]# wc -l anaconda-ks.cfg
49 anaconda-ks.cfg
```

Các tham số của lệnh `wc`:

-l: Hiển thị số dòng của file

-c: Hiển thị số ký tự của file

-w: Hiển thị số từ của file

- Tìm kiếm trong file hoặc trong đầu vào chuẩn các dòng có chứa các ký tự hoặc xâu thỏa mãn một số điều kiện nào đó. Sử dụng lệnh `grep`

Một số ký tự đặc biệt n `grep` sử dụng một số ký tự đặc biệt trong câu lệnh: . Biểu diễn 1 ký tự bất kỳ * Lặp lại ký tự ở vị trí trước ^ bắt đầu 1 dòng \$ kết thúc 1 dòng [...] xác định danh sách hoặc một khoảng các ký tự cần tìm kiếm [^..] các ký tự không tìm kiếm " Chú ý: để tránh nhầm lẫn, nên đặt các ký tự biểu diễn điều kiện trong ngoặc kép.

- `$grep [^t] /etc/passwd`: Tìm trong file `/etc/passwd` các dòng có ký tự “t”

```
[root@localhost ~]# grep [^t] /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
avahi-autoipd:x:170:170:Avahi IPv4LL Stack:/var/lib/avahi-autoipd:/sbin/nologin
systemd-bus-proxy:x:999:998:systemd Bus Proxy:/:/sbin/nologin
systemd-network:x:998:997:systemd Network Management:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:997:996:User for polkitd:/:/sbin/nologin
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
postfix:x:89:89:/:/var/spool/postfix:/sbin/nologin
nvdong:x:1000:1000:nvdong:/home/nvdong:/bin/bash
hackathon:x:1001:1001:/:/home/hackathon:/bin/bash
thuchanh:x:1002:1002:/:/home/thuchanh:/bin/bash
th2:x:1003:1003:/:/home/th2:/bin/bash
```

- Tìm trong file /etc/passwd các dòng bắt đầu bằng ký tự “t”

```
[root@localhost ~]# grep "^t" /etc/passwd
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin
thuchanh:x:1002:1002:~/home/thuchanh:/bin/bash
th2:x:1003:1003:~/home/th2:/bin/bash
[root@localhost ~]#
```

- So sánh file1 và file2 : diff file1 file2 ”

Tìm sự khác nhau giữa file1 và file2 (các file dưới dạng văn bản)

Kết quả hiển thị dưới dạng các dòng

```
[root@localhost ~]# ls -lrt
total 12
drwxr-xr-x. 2 root root 39 Apr 23 09:58 openscap_data
-rw-----. 1 root root 1310 Apr 23 09:59 anaconda-ks.cfg
-rwxr-xr-x. 1 root root 20 Apr 23 10:05 run.sh
-rwxr-xr-x. 1 root root 27 Apr 24 02:41 run1.sh
[root@localhost ~]# diff run.sh run1.sh
1a2
> Change
[root@localhost ~]#
```

- Lọc thông tin theo cột

Khi thông tin cần lọc có đầu vào dạng bảng và chỉ muốn hiển thị 1 cột thông tin nào đấy sử dụng lệnh awk như sau

grep [thông tin cần lọc] [nguồn đầu vào] | awk {'print \$[số thứ tự cột]'} }

ví dụ lệnh dk hiển thị thông tin dạng bảng như sau

```
[root@localhost ~]# df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/mapper/cl-root 18307072 887244 17419828 5% /
devtmpfs         503936      0      503936 0% /dev
tmpfs            513264      0      513264 0% /dev/shm
tmpfs            513264     6948     506316 2% /run
tmpfs            513264      0      513264 0% /sys/fs/cgroup
/dev/sda1        508588    121776     386812 24% /boot
tmpfs            102656      0      102656 0% /run/user/0
```

Muốn hiển thị chỉ thông tin cột 1 dùng lệnh sau:

dk | awk {'print \$1}'

THỰC HÀNH:

Bài 1: Tìm tất cả các dòng không bắt đầu bằng dấu # trong file /etc/sudoers

Bài 2: Sử dụng lệnh `df -k` và `awk` để hiển thị cột `use%`

MỘT SỐ DỊCH VỤ CƠ BẢN CỦA LINUX (CENTOS)

- Để xem các dịch vụ đang chạy sử dụng lệnh: `/sbin/service crond status`

```
[root@localhost ~]# /sbin/service crond status
Redirecting to /bin/systemctl status crond.service
■ crond.service - Command Scheduler
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2016-04-23 23:05:25 ICT; 3h 48min ago
   Main PID: 597 (crond)
   CGroup: /system.slice/crond.service
           └─597 /usr/sbin/crond -n

Apr 23 23:05:25 localhost.localdomain systemd[1]: Started Command Scheduler.
Apr 23 23:05:25 localhost.localdomain systemd[1]: Starting Command Scheduler...
Apr 23 23:05:26 localhost.localdomain crond[597]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 1% if used.)
Apr 23 23:05:27 localhost.localdomain crond[597]: (CRON) INFO (running with inotify support)
[root@localhost ~]#
```

Service (dịch vụ) thường được gọi là **daemon** là một chương trình xác định chạy ở nền của hệ thống và thường là không tương tác được (*non-interactive*). Các chương trình đó được sử dụng cho nhiều mục đích khác nhau bao gồm: quản lý phần cứng (*hardware*), truy cập mạng (*network access*), theo dõi (*monitoring*), ghi log (*logging*). Tất cả các hệ điều hành đều có một tập hợp các dịch vụ để tự động thực thi nhiều hoạt động.

Runlevel: trong hệ thống linux, người dùng có thể đặt một dịch vụ hoặc một nhóm các dịch vụ chạy ở một chế độ xác định để thực hiện một vài mục đích nào đó hoặc giới hạn lại mục đích sử dụng của một / nhóm các dịch vụ đó. Đối với hệ điều hành Fedora hoặc các hệ điều hành Linux dựa trên nền RedHat thì runlevel có các giá trị chính sau:

Runlevel-1: chế độ chạy đơn người dùng.(single-user mode)

Runlevel-2: chế độ chạy đa người dùng (multi-user mode)

Bài thực hành 3: Quản trị tài nguyên và dịch vụ

Runlevel-3: chế độ đa người người, hỗ trợ mạng (multi-user and networking mode)

Runlevel-5: X11 (runlevel 3 + X Windows System).

Thông thường, các dịch vụ chạy ở chế độ đồ họa (dựa trên X-Server như startx) thì thì runlevel ở mức 5 và các dịch vụ không chạy ở chế độ đồ họa thì runlevel ở mức 3. Bình thường, không có dịch vụ nào chạy ở runlevel 1.

Để xác định runlevel mà bạn đang sử dụng thì bạn sử dụng lệnh sau:

Code:

```
# /sbin/runlevel
```

Để xác định runlevel mà hệ thống của bạn sẽ chạy ở lần khởi động kế tiếp:

Code:

```
# cat /etc/inittab | grep :initdefault: id:5:initdefault:
```

PHẦN 4: TỰ ĐỘNG HÓA CÁC TÁC VỤ TRONG LINUX

- **Cron là gì:**

Xét về mặt kỹ thuật, **cron** là một tiện ích cho phép thực hiện các tác vụ một cách tự động theo định kỳ, ở chế độ nền của hệ thống hay còn gọi là **daemon**. **Crontab (CRON TABLE)** là một file chứa đựng thời khóa biểu (**schedule**) của các **entries** được chạy. Nó thường dùng để nén định kỳ các file ghi nhật ký, đồng bộ hóa hay backup dữ liệu.

- **Cron làm việc thế nào?**

cron chạy trên nền hệ thống và thường được khởi chạy lúc khởi động. Nó định kỳ kiểm tra các tác vụ cần làm mỗi một phút. Các tác vụ này được dựa trên các file crontab. Các file **crontab** thường được đặt tại thư mục **/var/spool/cron/crontabs**.

Các tệp tin trong thư mục này có tên trùng với tên người dùng, user.

Như vậy mỗi user đều có thể có crontab riêng.

Các file cấu hình và các thư mục liên quan:

File cấu hình chính: **/etc/crontab**

Thư mục lưu thông tin cho từng user: **/var/spool/cron/crontabs**

Xem thêm "**\$ man cron**".

• Cấu trúc của crontab

Một **crontab** file schedule đơn giản là một text file. Nó gồm có năm trường xác định thời gian, và trường cuối cùng là lệnh sẽ được chạy định kỳ, cấu trúc như sau:

minutes (0-59)

hour (0-23)

day of the month (1-31)

month of the year (1-12)

day of the week (0-6, 0=Sunday)

Các quy tắc chung:

- Các cột cách nhau bằng dấu cách space hay tab.
- 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.
- Giá trị liên tiếp kề nhau được định nghĩa bằng dấu trừ. Ví dụ: 1-5 có nghĩa là giá trị nằm từ 1 đến 5. Nếu nó ở cột "day of week" thì tác vụ này sẽ thực hiện từ thứ hai đến thứ sáu.
- Các giá trị không kề nhau thì cách nhau bằng dấu phẩy. Ví dụ 2,4,6 ở trường "ngày" sẽ là thứ ba, năm, bảy.
- Các khoảng thời gian lặp lại sẽ sử dụng "/". Ví dụ nếu trường thứ nhất là phút định nghĩa là */10 thì cứ 10 phút sẽ thực hiện một lần.

• Thao tác với crontab

Các tệp tin cấu hình của cron ở định dạng text nhưng chúng ta không nên thao tác trực tiếp bằng tay. Hệ điều hành thường đưa kèm cho bạn công cụ để bạn sửa các tệp tin này đó chính là "crontab".

\$ crontab -e: Chỉnh sửa crontab

\$ crontab -l: Liệt kê bảng crontab cho user hiện tại

Giả sử muốn ghi log mỗi phút. Vậy thì ta chạy lệnh sau:

Thực hiện: Crontab -e để mở crontab chỉnh sửa: Nhấn Esc + i

Gõ dòng sau vào

```
* * * * * date >> home/logs/log.txt
```

Sau đó lưu crontab bằng Esc+:wq!

```
* * * * * date >> home/logs/log.txt
```

```
[root@localhost ~]# crontab -l
* * * * * date >> /home/logs/log.txt
[root@localhost ~]#
```

THỰC HÀNH:

Bài 1: Đăng nhập với user root. Tạo 1 file tên **checkdate.sh** với nội dung như sau:

```
echo date >> log.txt
```

và lưu lại ở thư mục **/root/logs**

Thêm quyền thực thi cho file **checkdate.sh** này

Sau đó thêm 1 crontab tự động hóa chạy hằng giờ vào phút thứ 50 như sau:

```
50 * * * * /logs/checkdate.sh
```

BÀI TẬP VỀ NHÀ:

Bài 1: Thực hiện tạo 1 tác vụ tự động (crontab) để tạo ra 1 file log mỗi 20 phút hàng ngày ghi lại số lượng dòng của file log: /tmp/yum.log