

LAB 4: QUẢN LÝ TIỀN TRÌNH TRONG LINUX

Thời lượng: 3 tiết

Nội dung:

- Xem thông tin về tiến trình (process) bằng GUI và CMD
- Liên lạc giữa các tiến trình
- Độ ưu tiên của các tiến trình
- Tham khảo thêm các câu lệnh console trong lab 2

Quản lý tiến trình (Process Management)

1. Sử dụng lệnh top để xem thông tin về các tiến trình, và thông tin chung của hệ thống

```
top - 15:38:31 up 1 min, 2 users, load average: 0,56, 0,30, 0,11
Tasks: 213 total, 2 running, 211 sleeping, 0 stopped, 0 zombie
%Cpu(s): 8,5 us, 6,1 sy, 0,0 ni, 84,6 id, 0,0 wa, 0,0 hi, 0,8 si, 0,0 st
KiB Mem: 2063840 total, 613848 used, 1449992 free, 36796 buffers
KiB Swap: 1046524 total, 0 used, 1046524 free. 236104 cached Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1078	root	20	0	177288	46824	11944	S	19,6	2,3	0:17.46	Xorg
2036	ubuntu	20	0	316884	91836	34148	S	11,3	4,4	0:07.26	compiz
2056	ubuntu	20	0	217752	24836	19388	R	2,0	1,2	0:02.72	nautilus
1	root	20	0	4452	2504	1424	S	0,0	0,1	0:01.02	init
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	0:00.04	ksoftirqd/0
4	root	20	0	0	0	0	S	0,0	0,0	0:00.04	kworker/0:0
5	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	kworker/0:+
6	root	20	0	0	0	0	S	0,0	0,0	0:00.30	kworker/u1+
7	root	20	0	0	0	0	S	0,0	0,0	0:00.39	rcu_sched
8	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_bh
9	root	rt	0	0	0	0	S	0,0	0,0	0:01.26	migration/0
10	root	rt	0	0	0	0	S	0,0	0,0	0:00.00	watchdog/0
11	root	rt	0	0	0	0	S	0,0	0,0	0:00.00	watchdog/1
12	root	rt	0	0	0	0	S	0,0	0,0	0:01.00	migration/1
13	root	20	0	0	0	0	S	0,0	0,0	0:00.03	ksoftirqd/1
14	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kworker/1:0

2. Sử dụng lệnh ps để xem thông tin về các tiến trình

a. Xem tất cả các process : (dạng cây)

ps tree -a

```
ubuntu@ubuntu1:~$ ps tree -a
init
├── ModemManager
│   └── 2*[ModemManager]
├── NetworkManager
│   ├── dhclient -d -sf /usr/lib/NetworkManager/nm-dhcp-client.action -pf...
│   ├── dnsmasq --no-resolv --keep-in-foreground --no-hosts ...
│   └── 3*[NetworkManager]
├── accounts-daemon
│   └── 2*[accounts-daemon]
├── acpid -c /etc/acpi/events -s /var/run/acpid.socket
├── anacron -s
├── avahi-daemon
│   └── avahi-daemon
├── bluetoothd
├── colord
│   └── 2*[colord]
├── cron
├── cups-browsed
├── Amazon
├── dbus-daemon --system --fork
├── dbus-daemon --fork --print-pid 5 --print-address 7 --session
├── dbus-launch --autolaunch=d4f0a5f13cc619f5190cdc325a6ed249 ...
└── getty -8 38400 tty4
```

- b. Xem tiến trình cha (tổ tiên) của lệnh ps vừa thực thi : xem cột PID, PPID

ps -eljH |more

```
ubuntu@ubuntu1:~$ ps -eljH |more
```

F	S	UID	PID	PPID	PGID	SID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CM
1	S	0	2	0	0	0	0	80	0	-	0	kthrea	?	00:00:00	kt
1	S	0	3	2	0	0	0	80	0	-	0	smpboo	?	00:00:00	
1	S	0	4	2	0	0	0	80	0	-	0	worker	?	00:00:00	
1	S	0	5	2	0	0	0	60	-20	-	0	worker	?	00:00:00	
1	S	0	6	2	0	0	0	80	0	-	0	worker	?	00:00:00	
1	S	0	7	2	0	0	0	80	0	-	0	rcu_gp	?	00:00:00	
1	S	0	8	2	0	0	0	80	0	-	0	rcu_gp	?	00:00:00	
1	S	0	9	2	0	0	0	-40	-	-	0	smpboo	?	00:00:01	
5	S	0	10	2	0	0	0	-40	-	-	0	smpboo	?	00:00:00	
5	S	0	11	2	0	0	0	-40	-	-	0	smpboo	?	00:00:00	

- c. Kiểm tra một số dịch vụ đã chạy chưa ? (như cron, atd , sshd,...)

ps -e |grep cron

```
ubuntu@ubuntu1:~$ ps -e |grep cron
  978 ?        00:00:00 anacron
  979 ?        00:00:00 cron
```

- d. Xem trạng thái của process (running, sleeping,...) : xem cột S

ps -eljH |more

```
ubuntu@ubuntu1:~$ ps -eljH |more
```

F	S	UID	PID	PPID	PGID	SID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CM
1	S	0	2	0	0	0	0	80	0	-	0	kthrea	?	00:00:00	kt
1	S	0	3	2	0	0	0	80	0	-	0	smpboo	?	00:00:00	
1	S	0	4	2	0	0	0	80	0	-	0	worker	?	00:00:00	
1	S	0	5	2	0	0	0	60	-20	-	0	worker	?	00:00:00	
1	S	0	6	2	0	0	0	80	0	-	0	worker	?	00:00:00	
1	S	0	7	2	0	0	0	80	0	-	0	rcu_gp	?	00:00:00	
1	S	0	8	2	0	0	0	80	0	-	0	rcu_gp	?	00:00:00	
1	S	0	9	2	0	0	0	-40	-	-	0	smpboo	?	00:00:01	
5	S	0	10	2	0	0	0	-40	-	-	0	smpboo	?	00:00:00	
5	S	0	11	2	0	0	0	-40	-	-	0	smpboo	?	00:00:00	

- e. Xem các process của user hiện hành

ps hoặc vips -l

```
ubuntu@ubuntu1:~$ ps
  PID TTY          TIME CMD
 2620 pts/0    00:00:00 bash
 2667 pts/0    00:00:00 ps
ubuntu@ubuntu1:~$ ps -l
F S  UID      PID  PPID  C PRI  NI ADDR SZ  WCHAN  TTY          TIME CMD
0 S  1000    2620  2610  0  80   0 - 2092 wait pts/0    00:00:00 bash
0 R  1000    2668  2620  0  80   0 - 1615 - pts/0    00:00:00 ps
```

- f. Mở một terminal mới (tty2), login vào bằng tài khoản user1. Chạy vi. Mở một terminal mới (tty3), login vào bằng tài khoản user2. Chạy top.

```

VIM - Vi IMproved
      version 7.4.52
      by Bram Moolenaar et al.
Modified by pkg-vim-maintainers@lists.alioth.debian.org
Vim is open source and freely distributable

  Become a registered Vim user!
type  :help register<Enter>    for information
type  :q<Enter>                to exit
type  :help<Enter> or <F1>     for on-line help
type  :help version7<Enter>   for version info

  Running in Vi compatible mode
type  :set nocp<Enter>        for Vim defaults
type  :help cp-default<Enter> for info on this
  
```

```

top - 15:53:10 up 16 min, 7 users, load average: 0,18, 0,36, 0,28
Tasks: 260 total, 2 running, 258 sleeping, 0 stopped, 0 zombie
%Cpu(s): 6,6 us, 6,4 sy, 0,0 ni, 86,5 id, 0,0 wa, 0,0 hi, 0,5 si, 0,0 st
KiB Mem: 2063840 total, 1026720 used, 1037120 free, 40624 buffers
KiB Swap: 1046524 total, 0 used, 1046524 free. 275672 cached Mem
  
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1078	root	20	0	192500	52464	12720	R	18,6	2,5	2:55.61	Xorg
2036	ubuntu	20	0	326608	93552	35604	S	7,0	4,5	0:56.36	compiz
2056	ubuntu	20	0	217752	24836	19388	S	2,7	1,2	0:21.40	nautilus
7	root	20	0	0	0	0	S	0,3	0,0	0:01.51	rcu_sched
1	root	20	0	4452	2504	1424	S	0,0	0,1	0:01.03	init
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	0:00.44	ksoftirqd/0
4	root	20	0	0	0	0	S	0,0	0,0	0:00.04	kworker/0:0
5	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	kworker/0:+
8	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_bh
9	root	rt	0	0	0	0	S	0,0	0,0	0:01.27	migration/0
10	root	rt	0	0	0	0	S	0,0	0,0	0:00.00	watchdog/0
11	root	rt	0	0	0	0	S	0,0	0,0	0:00.00	watchdog/1
12	root	rt	0	0	0	0	S	0,0	0,0	0:01.01	migration/1
13	root	20	0	0	0	0	S	0,0	0,0	0:00.28	ksoftirqd/1
14	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kworker/1:0
15	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	kworker/1:+

Trên terminal của root, dùng lệnh:

`ps -a` //hiển thị các process đang chạy trên các terminal hay

`ps -af` //hiển thị các process đang chạy trên các terminal và chủ nhân

```

root@ubuntu1:~# ps -a
  PID TTY          TIME CMD
 4446 pts/2        00:00:00 vi
 4584 pts/0        00:00:00 top
 4654 pts/4        00:00:00 sudo
 4655 pts/4        00:00:00 bash
 4669 pts/4        00:00:00 ps
root@ubuntu1:~# ps -af
UID          PID  PPID  C STIME TTY          TIME CMD
ubuntu      4446   2676  0 15:48 pts/2        00:00:00 vi
ubuntu      4584   4520  0 15:50 pts/0        00:00:00 top
root         4654   4592  0 15:50 pts/4        00:00:00 sudo -i
root         4655   4654  0 15:50 pts/4        00:00:00 -bash
root         4670   4655  0 15:50 pts/4        00:00:00 ps -af
root@ubuntu1:~#
  
```

3. Sử dụng lệnh kill và killall

a. Xem danh sách các SIGNAL của lệnh kill

kill -l

```
ubuntu@ubuntu1:~$ kill -l
1) SIGHUP      2) SIGINT      3) SIGQUIT     4) SIGILL      5) SIGTRAP
6) SIGABRT     7) SIGBUS     8) SIGFPE      9) SIGKILL     10) SIGUSR1
11) SIGSEGV    12) SIGUSR2    13) SIGPIPE    14) SIGALRM     15) SIGTERM
16) SIGSTKFLT  17) SIGCHLD   18) SIGCONT    19) SIGSTOP     20) SIGTSTP
21) SIGTTIN    22) SIGTTOU   23) SIGURG     24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM  27) SIGPROF   28) SIGWINCH   29) SIGIO        30) SIGPWR
31) SIGSYS     34) SIGRTMIN  35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9 56) SIGRTMAX-8 57) SIGRTMAX-7
58) SIGRTMAX-6 59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2
63) SIGRTMAX-1 64) SIGRTMAX
```

b. Xem ý nghĩa của 4 SIGNAL thường dùng (đọc manual)

Tên signal	Số hiệu signal	Miêu tả
SIGHUP	1	Trì hoãn việc kiểm tra trên quản lý terminal hoặc sự dừng của quản lý tiến trình.
SIGINT	2	Được thông báo nếu người sử dụng gửi một tín hiệu ngắt (Ctrl+C).
SIGQUIT	3	Được thông báo nếu người sử dụng gửi một tín hiệu bỏ (Ctrl+D).
SIGFPE	8	Được thông báo nếu một hoạt động thuộc về toán không hợp pháp được thử chạy.

c. Chạy lệnh sau và thực hiện theo yêu cầu

find / -name "cron*"

- Khi lệnh đang thực thi, nhấn Ctrl_Z để dừng tiến trình.

```
/etc/pam.d/cron
/etc/cron.hourly
find: '/etc/polkit-1/localauthority': Permission denied
/etc/crontab
find: '/etc/ssl/private': Permission denied
/etc/cron.d
/etc/cron.weekly
/etc/init/cron.conf
/etc/default/cron
/etc/cron.monthly
/etc/init.d/cron
/etc/cron.daily
/usr/bin/crontab
/usr/share/man/man8/cron.8.gz
/usr/share/man/man5/crontab.5.gz
/usr/share/man/man1/crontab.1.gz
/usr/share/bash-completion/completions/crontab
/usr/share/bug/cron
/usr/share/doc/cron
/usr/share/doc/cron/examples/cron-tasks-review.sh
/usr/share/doc/cron/examples/cron-stats.pl
/usr/share/doc/cron/examples/crontab2english.pl
/usr/sbin/cron
ubuntu@ubuntu1:~$
```

- Kiểm tra bằng lệnh `ps -l` xem lệnh còn hoạt động không ? (Tiến trình ứng với lệnh `find` có PID nào? Ở trạng thái gì?)

```
ubuntu@ubuntu1:~$ ps -l
  PID TTY          STAT       TIME COMMAND
    1 ?           Ss          0:01 /sbin/init
ubuntu@ubuntu1:~$ ps -l
 F S    UID     PID   PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
 0 S   1000    4862   4852  0  80   0 -  2075 wait   pts/0      00:00:00 bash
 0 R   1000    5832   4862  0  80   0 -  1615 -      pts/0      00:00:00 ps
ubuntu@ubuntu1:~$
```

- Chấm dứt hoạt động của tiến trình `find` bằng lệnh sau có được không?

- `kill -SIGTERM <PID_find>`
- (kiểm tra lại bằng lệnh `ps`)

```
ubuntu@ubuntu1:~$ kill -SIGTERM 4862
ubuntu@ubuntu1:~$ ps
  PID TTY          TIME CMD
 4862 pts/0      00:00:00 bash
 5840 pts/0      00:00:00 ps
ubuntu@ubuntu1:~$
```

- Chấm dứt hoạt động của tiến trình `find` bằng lệnh sau có được không?

- `kill -SIGINT <PID_find>`
- (kiểm tra lại bằng lệnh `ps`)

```
ubuntu@ubuntu1:~$ kill -SIGINT 4862
ubuntu@ubuntu1:~$ ps
  PID TTY          TIME CMD
 4862 pts/0      00:00:00 bash
 5853 pts/0      00:00:00 ps
ubuntu@ubuntu1:~$
```

- Chấm dứt hoạt động của tiến trình `find` , bằng lệnh sau có được không ?

- `kill -SIGKILL <PID_find>`
- (kiểm tra lại bằng lệnh `ps`)

```
ubuntu@ubuntu1:~$ ps
  PID TTY          TIME CMD
 6137 pts/0      00:00:00 bash
 6200 pts/0      00:00:00 ps
ubuntu@ubuntu1:~$ kill -SIGKILL 6137
```

d. Dùng `kill` hủy terminal hiện tại : Terminal hiện tại tương ứng với tiến trình `bash`, vậy nên hủy terminal hiện tại có nghĩa là hủy tiến trình `bash` => kết quả là sẽ kết thúc `bash`, và quay lại tiến trình `login` (thấy xuất hiện dấu nhắc `login`)

⇒ Hãy viết lệnh hủy tiến trình `bash` bằng lệnh `kill`?

- ⇒ **B1: nhập tên tiến trình**
- ⇒ **B2: đọc tên tiến trình**

- ⇒ **B3: sử dụng lệnh (ps aux | grep nameprocess | grep -v grep | awk '{print \$2}'**
- ⇒
- ⇒ **Sử dụng lệnh kill PID_find**
- ⇒ **Vd: kill 4650**
- ⇒ Kill- SIGTERM 4650

e. Dùng killall huỷ tất cả các bash shell hiện tại trong 1 terminal :

- Trên terminal tty1 , login vào hệ thống với user1
- Dùng lệnh su, chuyển sang dùng quyền của user2

```
ubuntu@ubuntu1:~$ su user2
Password:
user2@ubuntu1:/home/ubuntu$
```

- Tiếp tục , dùng lệnh su, chuyển sang dùng quyền của user3

```
user2@ubuntu1:/home/ubuntu$ su user3
Password:
user3@ubuntu1:/home/ubuntu$
```

- Xem toàn bộ các tiến trình phát sinh sau chuỗi lệnh trên
 - *ps -aljH | grep tty1*

```
user3@ubuntu1:/home/ubuntu$ ps
  PID TTY          TIME CMD
 7428 pts/0        00:00:00 bash
 7447 pts/0        00:00:00 ps
user3@ubuntu1:/home/ubuntu$ ps -aljH |grep 7428
4 S   1003   7428   7427   7428   7020  0  80   0 -  2076 wait    pts/0    00:00:00
      bash
0 R   1003   7449   7428   7449   7020  0  80   0 -  1607 -      pts/0    00:00:00
      ps
0 S   1003   7450   7428   7449   7020  0  80   0 -  1538 pipe_w pts/0    00:00:00
      grep
```

⇒ Để kill tất cả các bash shell hiện tại (trong 1 terminal) , dùng lệnh

kill -SIGKILL <PID_của shell gốc trên cùng>

```
user3@ubuntu1:/home/ubuntu$ ps -aljH |grep 7428
4 S   1003   7428   7427   7428   7020  0  80   0 -  2076 wait    pts/0    00:00:00
      bash
0 R   1003   7449   7428   7449   7020  0  80   0 -  1607 -      pts/0    00:00:00
      ps
0 S   1003   7450   7428   7449   7020  0  80   0 -  1538 pipe_w pts/0    00:00:00
      grep
user3@ubuntu1:/home/ubuntu$ kill -SIGKILL 7428
```

⇒ Để kill tất cả các bash shell hiện tại trên tất cả các terminal, dùng

killall -SIGKILL bash

```
user2@ubuntu1:/home/ubuntu$ ps
  PID TTY          TIME CMD
 7414 pts/0    00:00:00 bash
 7452 pts/0    00:00:00 ps
user2@ubuntu1:/home/ubuntu$ killall -SIGKILL bash
```

4. Tìm hiểu dịch vụ lập lịch cron:

a. Lập lịch để mỗi phút ghi một chuỗi vào file test , Thao tác như sau :

- Dùng quyền của root, mở file /etc/crontab bằng vi hoặc nano Nhập vào cuối file dòng sau :

```
* * * * * root echo "hello" >> /test.txt
```

```
GNU nano 2.2.6      File: crontab      Modified
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
#
* * * * * root echo "hello" >>/test.txt

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

- Kiểm tra bằng cách: chờ vài phút trôi qua, rồi mở file test.txt để xem. Hoặc chỉnh lại đồng hồ hệ thống rồi xem lại file.

```
root@ubuntu1:~# nano test.txt
root@ubuntu1:~# cat test.txt
***** root echo "hello" >>/test.txt
```

Bài tập đề nghị

Xem/Thay đổi độ ưu tiên của tiến trình => lệnh Set Priority hoặc sử dụng Task Manager

Sử dụng lệnh findstr tìm chuỗi “Windows” trong các file .txt có trong ổ đĩa C:

Thay đổi độ ưu tiên của tiến trình từ Normal => High.

Đo lường thời gian và so sánh để thấy kết quả của thiết lập này?

```
ubuntu@ubuntu1:~$ ps -eo
error: format specification must follow -o

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.

For more details see ps(1).
ubuntu@ubuntu1:~$
```

Tham khảo về Task Manager và Priority:

<https://technet.microsoft.com/en-us/library/cc938567.aspx> //task manager

<https://technet.microsoft.com/en-us/library/cc938611.aspx> //priority

<http://woshub.com/process-priority-management-in-windows/>