Linux và Phần mềm Mã nguồn mở

Bài 7: tiến trình và lập lịch

Nhắc lại và chú ý

- Hệ thống file của linux và một số khái niệm quan trọng (superblock, inode, storageblock)
- Gắn kết (mount) và tháo gỡ thiết bị lưu trữ
- Cấu hình tự động gắn kết bằng file “/etc/fstab”
- Cắt và ghép tập tin
- Soạn thảo tập tin văn bản bằng “vi”
- Dẫn hướng vào ra dữ liệu
- Ống (pipe)

Nội dung

1. Tiến trình trên linux
   - Các khái niệm liên quan
   - Các loại tiến trình
   - Một số lệnh thông dụng làm việc với tiến trình
   - Điều khiển tác vụ
2. Lập lịch cho các hoạt động thường xuyên
   - Hoạt động thường xuyên
   - Lập lịch bằng crontab
   - Ví dụ

Phần 1

Tiến trình trên linux

Các khái niệm liên quan

- Chương trình (program) là một file thực thi trong hệ thống, ví dụ: /sbin/shutdown
- Tiến trình (process) là một chương trình đã được nạp vào bộ nhớ và được cấp CPU để hoạt động
  - Ta mở nhiều cửa sổ terminal để thử nghiệm các lệnh, mỗi cửa sổ là một tiến trình
  - Tiến trình đôi khi được gọi là tác vụ (task)
- Khi khởi chạy, mỗi tiến trình được cấp một chỉ số PID (process id) duy nhất. Hệ thống dùng PID để quản lý tiến trình

Các khái niệm liên quan

- Tiến trình cũng có phân quyền sở hữu và truy cập (như với tập tin)
- Linux cho phép nhiều tiến trình chạy cùng lúc
  - Nhân linux có một module riêng lập lịch phân phối CPU cho từng tiến trình để đảm bảo các tiến trình đều được hoạt động hợp lý
  - Mỗi tiến trình có một chỉ số ưu tiên (priority) tương ứng
  - Chỉ số ưu tiên càng thấp thì hệ thống càng ưu tiên phân phối nhiều thời gian sử dụng CPU cho tiến trình đó
  - Có thể chỉnh lại chỉ số ưu tiên này bằng lệnh nice hoặc renice

Các loại tiến trình

- Một tiến trình có thể yêu cầu hệ thống khởi chạy một tiến trình khác (ví dụ: trình duyệt có thể tạo một process riêng khi người dùng mở một link)
  - Khi đó tiến trình được tạo ra gọi là child process
  - Tiến trình ban đầu được gọi là parent process
- Một tiến trình con đang chạy nhưng tiến trình cha của nó đã kết thúc thì được gọi là orphan process
- Một tiến trình đã hoàn tất nhưng vì một lý do gì đó vẫn được giữ trong bộ nhớ thì được gọi là zombie process hoặc defunct process

Các loại tiến trình

- Các tiến trình nhận tương tác từ người dùng thì hoạt động ở chế độ mặt trước (foreground)
- Các tiến trình không nhận tương tác thì hoạt động ở chế độ nền (background)
- Các tiến trình thường chuyển qua chuyển lại giữa hai trạng thái này trong quá trình hoạt động, việc chuyển trạng thái có thể thực hiện do người dùng, do lệnh từ shell hoặc do lập trình
- Tiến trình ở chế độ mặt trước thường nhận được nhiều CPU hơn một chút so với chế độ nền

Các loại tiến trình

- Hệ thống linux có một số các tiến trình đặc biệt gọi là các daemon process
  - Thường cung cấp các chức năng quan trọng của hệ thống, đặc biệt là các dịch vụ mạng
  - Thường thuộc về quyền root
  - Thường không gắn với shell cụ thể nào, không truy xuất vào/ra bàn phím, màn hình
  - Khi sử dụng câu lệnh liệt kê tiến trình sẽ thấy kí hiệu ? ở trường TTY
  - Đa số daemon process không chiếm CPU, chúng chỉ hoạt động khi có yêu cầu

Liệt kê các tiến trình: "ps"

- Cú pháp: ps [options]
- Một số tùy chọn:
  - a tất cả proc của các user khác
  - x các proc không gắn với terminal (daemon)
  - u user-format
  - l long-format
  - w wide output
  - -U user xem proc của một user cụ thể

```
ps aux
ps aux | grep httpd
```

Liệt kê các tiến trình: "ps"

```
$ ps
PID   TTY     TIME       CMD
728   pts/3   00:00:00   bash
1010  pts/3   00:00:00   ps

$ ps -auw
USER    PID %CPU %MEM    VSZ    RSS   TTY     STAT   START   TIME   COMMAND
root    728  0.0  0.6   3528   1604   pts/3   S      21:08   0:00   /bin/bash
root   1161  0.0  0.3   3548    860   pts/3   R      22:29   0:00   ps auw
```

- Trạng thái (cột STAT):
  - R Đang thi hành
  - S Đang bị đóng
  - Z Ngừng thi hành
  - W Không đủ bộ nhớ cho tiến trình thi hành

Thông tin chi tiết: "top"

- Liên tục hiển thị thông tin về các tiến trình
- Cú pháp: `top [options]`
- Một số tùy chọn:
  - -d delay Khoảng thời gian trễ giữa hai lần cập nhật
  - -p [pid] Chỉ theo dõi tiến trình có mã là pid
- Một số phím lệnh trong sử dụng trong top:
  - q Thoát khỏi lệnh top
  - space Cập nhật thông tin tiến trình ngay lập tức
  - k Ngừng một tiến trình
  - f Lựa chọn thông tin tiến trình

Thông tin chi tiết: "top"

```
top - 23:08:08 up 12 min,  2 users,  load average: 0.75, 0.68, 0.42
Tasks: 107 total,   2 running, 105 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.3%us,  1.3%sy,  0.0%ni, 98.0%id,  0.3%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:    255392k total,   249748k used,     5644k free,    13176k buffers
Swap:   650624k total,        0k used,   650624k free,   139668k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2486	root	15	0	31768	7380	3276	S	1.3	2.9	0:11.22	Xorg
1843	root	15	0	2380	860	712	S	0.3	0.3	0:01.72	vmware-guestd
2328	root	15	0	3140	868	752	S	0.3	0.3	0:00.15	hald-addon-stor
2731	root	15	0	8224	2528	1924	S	0.3	1.0	0:01.57	vmware-user
1	root	15	0	2140	628	540	S	0.0	0.2	0:01.06	init
2	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
3	root	34	19	0	0	0	S	0.0	0.0	0:00.99	ksoftirqd/0
4	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/0
5	root	10	-5	0	0	0	S	0.0	0.0	0:00.07	events/0
6	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	khelper
7	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	kthread
46	root	10	-5	0	0	0	S	0.0	0.0	0:00.03	kblockd/0
47	root	20	-5	0	0	0	S	0.0	0.0	0:00.00	kacpid
111	root	20	-5	0	0	0	S	0.0	0.0	0:00.00	cqueue/0
112	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	ksuspend_usbd
115	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	khubd
117	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	kseriod

Thông tin chi tiết: "top"

- Đây là công cụ mà quản trị hệ thống linux nào cũng cần biết và sử dụng thành thạo
- Cung cấp thông tin về tiến trình, có thể thực hiện luôn các thao tác với tiến trình (ví dụ: kết thúc tiến trình, thay đổi mức độ ưu tiên,…)
- Cung cấp các chỉ số quan trọng của hệ thống:
  - Thời gian hiện tại, thời gian từ lần khởi động mới nhất
  - Mức độ tải CPU trung bình trong 1, 5, 15 phút gần đây
  - Mức độ chiếm dụng CPU hiện tại
  - Các chỉ số quan trọng của từng tiến trình

Ngừng tiến trình: "kill"

- Cú pháp: kill [-s signal] pid
  kill -l [signal]
- Một số signal:

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

Ngừng tiến trình với tên: "killall"

- Cú pháp: killall [-s signal] name
- Quyền hủy tiến trình (cả kill và killall) thuộc về người sở hữu tiến trình hoặc quyền root
- Lệnh killall kết thúc mọi tiến trình cùng tên, vì thế cẩn thận khi sử dụng lệnh này
  - Tiến trình xử lý web bị lỗi, nếu hủy bằng killall có thể dẫn đến hủy mọi giao dịch web đang thực hiện
- Ví dụ:

  killal -HUP syslogd

  killall -9 man

Điều khiển tác vụ

- Một tác vụ (job) là một tiến trình đang thực thi
- Một số cách điều khiển tác vụ:
  - ^C thoát ngang
  - ^Z chuyển sang chế độ nền
  - “jobs” liệt kê các tác vụ đang thực thi
  - & thực hiện tác vụ ở chế độ nền
- Tiến trình bị tạm ngừng bởi ctrl-Z có thể được tiếp tục bằng lệnh fg hoặc bg
  - fg %x tiếp tục tác vụ x ở foreground
  - bg %x tiếp tục tác vụ x ở background

Thi hành lệnh ở background

- Để tiến trình chạy ở chế độ background, chúng ta thêm dấu & vào sau lệnh thực hiện chương trình
- Ví dụ :

  ```
  find / -name pro -print > results.txt &
  ```

- Để kiểm tra, ta có thể dùng lệnh:
  - ps –aux | grep find
  - "jobs" để xem các tiến trình đang có ở background

Theo dõi hệ thống

- w: xem các user còn đang login đang làm gì
- free: hiển thị thông tin bộ nhớ.
- uptime: thời gian sống của hệ thống
- pstree: hiển thị cây tiến trình
- pgrep, pkill: tìm hoặc gửi signal đến tiến trình dựa theo tên và các thuộc tính khác
- nice, renice, snice: thay đổi priority của tiến trình

Sinh viên chủ động tìm hiểu các lệnh trên!

Phần 2

Lập lịch cho các hoạt động thường xuyên

Hoạt động thường xuyên là gì?

- Những công việc phải làm lặp lại hàng giờ, hàng ngày, hàng tuần, hàng tháng,… nói chung là lặp lại theo định kỳ
  - Kiểm tra email mỗi 10 phút
  - Yêu cầu thay đổi mật khẩu đăng nhập mỗi 2 tháng
  - Kiểm tra và cập nhật hệ thống vào 3 giờ sáng hàng ngày
- Đối với hệ thống thông tin, những việc được thực hiện khi điều kiện đặc biệt nào đó xảy ra cũng được coi là hoạt động thường xuyên
  - Khóa tài khoản 10 phút nếu nhập sai mật khẩu 3 lần
  - Phát cảnh báo nếu nhiệt độ CPU quá 90^0C

Lập lịch bằng crontab

- Những việc lặp đi lặp lại gây nhàm chán cho người dùng vì thế linux có cơ chế cho phép tự động hóa những hoạt động này bằng tiện ích cron
- Về cơ bản cron là một tiến trình daemon của linux
- Người dùng thiết lập các việc cần làm trong các file văn bản đặc biệt của cron (gọi là crontab file)

"`crontab -e`": tạo hoặc chỉnh file crontab, giống "vi"

"`crontab -l`": hiển thị file crontab

"`crontab -r`": xóa file crontab

Lập lịch bằng crontab

- Mỗi người dùng có crontab của riêng họ đặt trong thư mục "/var/spool/cron/" (mỗi người 1 file)
- Các crontab chung của hệ thống đặt ở một số nơi khác, hệ thống sẽ quét các file này và xử lý định kỳ
  - /etc/crontab
  - /etc/cron.d/
  - /etc/cron.hourly/
  - /etc/cron.daily/
  - /etc/cron.weekly/
  - /etc/cron.monthly/

Lập lịch bằng crontab

- Soạn nội dung của 1 lệnh cron: 6 tham số

  <phút> <giờ> <ngày> <tháng> <thứ> <câu lệnh>

- <lệnh> viết như lệnh shell thông thường
- Các phần liên quan đến thời gian quy ước như sau:
  - Cách nhau bởi dấu space hoặc tab
  - Xảy ra với mọi giá trị: dùng dấu *
  - Giá trị liên tiếp: dùng gạch nối, ví dụ 2-6
  - Giá trị rời rạc: liệt kê nối với nhau bằng dấu phẩy
  - Lặp lại dùng dấu /
  - <thứ>: số 0 là chủ nhật, số 1 là thứ hai,…

Lập lịch bằng crontab

```
// mỗi phút thực hiện clear một lần
* * * * * clear
// 5 giờ sáng thứ 2 hàng tuần: backup dữ liệu
0 5 * * 1 tar -zcf /var/bks/home.tgz /home/
// tắt máy vào 5 rưỡi chiều hàng ngày
30 17 * * * shutdown
// chạy khi khởi động lại máy
@reboot echo “hay nghi ngoi mot chut”
```

Các bước thực hiện

- Tạo crontab mới: crontab -e
- Soạn như soạn thảo "vi":
  - Bấm insert để bắt đầu soạn thảo
  - Gõ: * * * * * date >> /tmp/times.log
  - Bấm ESC để thoát chế độ soạn thảo
  - Gõ lệnh lưu và thoát - :wq
- Khởi động lại dịch vụ cron để cập nhật lệnh mới: service crond restart
- Theo dõi sự cập nhật của file times.log xem dịch vụ chạy đúng không: tail -f /tmp/times.log