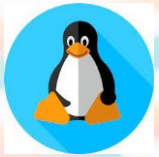
PROGRESS AND SCHEDULING

Van Tham Nguyen

Nội dung

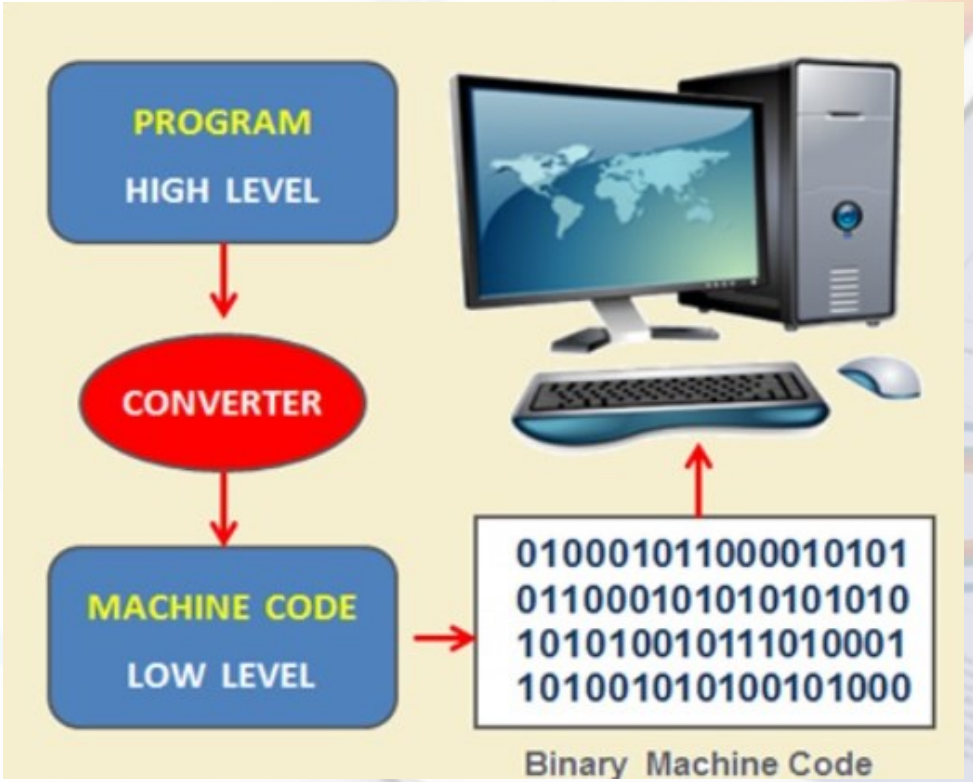
1. Quản lý tiến trình trên linux
   - Các khái niệm liên quan
   - Các loại tiến trình
   - Một số lệnh thông dụng làm việc với tiến trình
   - Điều khiển tác vụ
2. Lập lịch cho các hoạt động thường xuyên
   - Hoạt động thường xuyên
   - Lập lịch bằng crontab
   - Ví dụ



PART 1
PROCESS MANAGEMENT IN LINUX

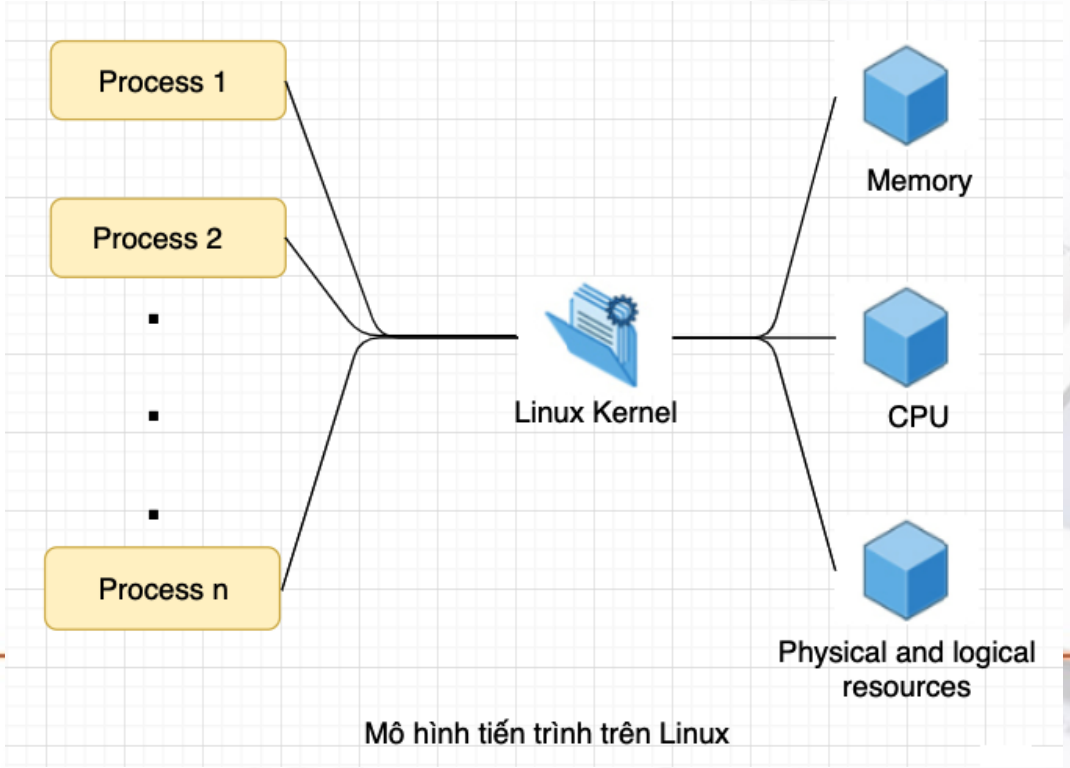
Các khái niệm liên quan

- Chương trình (program) là một file thực thi trong hệ thống, ví dụ: /sbin/shutdown



Các khái niệm liên quan

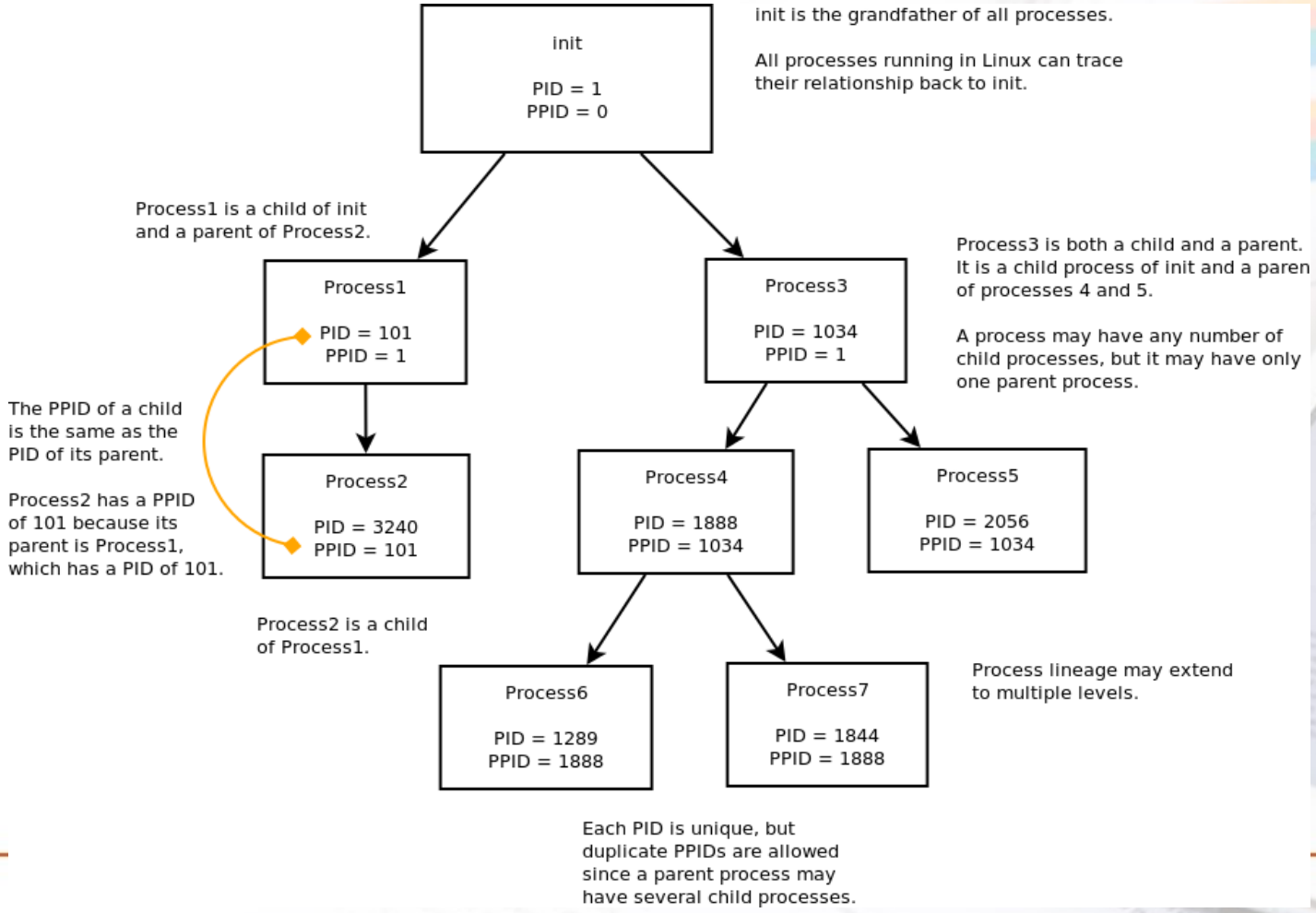
- Tiến trình (process) là một chương trình đã được nạp vào bộ nhớ và được cấp CPU để hoạt động
  - Ta mở nhiều cửa sổ terminal để thử nghiệm các lệnh, mỗi cửa sổ là một tiến trình
  - Tiến trình đôi khi được gọi là tác vụ (task)



Mô hình tiến trình trên Linux

Các khái niệm liên quan

- 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.
  - Khi đó tiến trình được tạo ra gọi là child process
  - Tiến trình ban đầu được gọi là parent process

Ví dụ: Trình duyệt có thể tạo một process riêng khi người dùng mở một link

- Một tiến trình con đang chạy nhưng tiến trình cha của nó đã kết thúc thì được gọi là orphan process (tiến trình mồ côi)
- 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 (tiến trình không tồn tại)

Các loại tiến trình

- Tiến trình hệ thống:

+ Do hệ thống khởi tạo

+ Thường để hệ thống điều khiển và quản lý

- Tiến trình người dùng:

+ Do người dùng khởi tạo

+ Quản trị viên hệ thống điều khiển và quản lý tiến trình của người dùng

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

- 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`: Xem toàn bộ tiến trình trên hệ thống

Liệt kê các tiến trình

```
[root@localhost ~]# ps
    PID TTY          TIME CMD
   1423 tty1     00:00:00 bash
   1456 tty1     00:00:00 ps
[root@localhost ~]# ps -a
    PID TTY          TIME CMD
   1457 tty1     00:00:00 ps
[root@localhost ~]# ps -u
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root        1423  0.0  0.2  26244  3908 tty1     Ss   15:01   0:00 -bash
root        1458  0.0  0.2  58728  3924 tty1     R+   15:06   0:00 ps -u
[root@localhost ~]#
```

- 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

Liệt kê các tiến trình

- Tạo tiến trình vi Bai.txt

```
[root@localhost ~]# ps -u
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root       1401  0.0  0.2  26244  3908 tty1     Ss   22:44   0:00 -bash
root       1483  0.0  0.2  58728  3976 tty1     R+   23:11   0:00 ps -u
[root@localhost ~]# vi Bai.txt
```

- Dừng tiến trình vi Bai.txt: Ctrl + Z

```
"Bai.txt" [New File]

[1]+  Stopped                 vi Bai.txt
[root@localhost ~]# ps -u
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root       1401  0.0  0.2  26244  3908 tty1     Ss   22:44   0:00 -bash
root       1484  0.0  0.2  37240  4124 tty1     T    23:12   0:00 vi Bai.txt
root       1485  0.0  0.2  58728  3908 tty1     R+   23:12   0:00 ps -u
[root@localhost ~]# _
```

Thông tin tiến trình

- 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 tiến trình

```
[root@localhost ~]# top_
```

```
top - 15:24:49 up 24 min,  1 user,  load average: 0.00, 0.00, 0.00
Tasks:  97 total,   1 running,  96 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.1 us,  0.0 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.1 hi,  0.1 si,  0.0 st
MiB Mem :   1769.3 total,   1372.2 free,    155.8 used,    241.4 buff/cache
MiB Swap:   1024.0 total,   1024.0 free,      0.0 used.   1465.5 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    7 root      20   0       0      0      0 I   0.1   0.0   0:01.21 kworker/0:1-events
 1473 root      20   0   65412   4676   3920 R   0.1   0.3   0:00.22 top
    1 root      20   0  186004  13764   8936 S   0.0   0.8   0:01.03 systemd
    2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
    3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
    6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
```

```
[root@localhost ~]# top -p 814, 1
```

```
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  814 root      20   0  227700  15244  12496 S   0.0   0.8   0:00.13 sssd_be
    1 root      20   0  186004  13764   8936 S   0.0   0.8   0:01.04 systemd
```

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

Ngừng các tiến

- 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

```
[root@localhost ~]# ps -u
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root        1401  0.0  0.2  26244  3908 tty1     Ss   22:44   0:00 -bash
root        1484  0.0  0.2  37240  4124 tty1     T    23:12   0:00 vi Bai.txt
root        1485  0.0  0.2  58728  3908 tty1     R+   23:12   0:00 ps -u
[root@localhost ~]# kill -9 1484
[root@localhost ~]# ps -u
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root        1401  0.0  0.2  26244  3908 tty1     Ss   22:44   0:00 -bash
root        1486  0.0  0.2  58728  3856 tty1     R+   23:14   0:00 ps -u
[1]+  Killed                  vi Bai.txt
[root@localhost ~]#
```

Ngừng tiến trình

Ngừng tiến trình theo tên

- Cú pháp: killall [-s signal] <name of process>
- 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 :

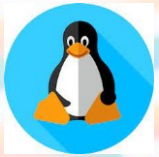
  ```
  find / -name pro –print > results.txt &
  ```

- 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 sau 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

Các công cụ lập lịch

- **Lập lịch bằng Crontab**
- Lập lịch bằng AT, APT, ATRM
- Lập lịch bằng Systeam Timer

```
[root@localhost ~]# cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name  command to be executed

[root@localhost ~]# _
```

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
- **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.
- **Crontab** (CRON TABle) là một file chứa đựng bảng biểu (schedule) của các thực thể được chạy.
- Người dùng thiết lập các việc cần làm trong các file văn bản đặc biệt của cron.

Lập lịch bằng crontab

Các câu lệnh

- Crontab files không cho phép người dùng tạo hoặc chỉnh sửa trực tiếp với bất kỳ trình text editor nào,
- Người dùng muốn làm việc với Crontab files thì sử dụng các lệnh:

- Tạo hoặc chỉnh file crontab (giống vi)

Cú pháp: crontab -e

```
[root@localhost ~]# crontab -e_
```

```
* * * * * clear
15 16 * * * echo "Relax" > /home/Tommy/testfile_Crontab.txt
~
~
```

Lập lịch bằng crontab

Các câu lệnh

- Xóa file crontab

  Cú pháp crontab -r

```
[root@localhost ~]# crontab -r
[root@localhost ~]# crontab -l
no crontab for root
[root@localhost ~]#
```

Lập lịch bằng crontab

Các câu lệnh

- Hiện thị nội dung file crontab

  Cú pháp : crontab -l

```
[root@localhost ~]# crontab -l
@reboot echo "Khoi dong lai !"
* * * * * clear
15 16 * * * echo "Relax" > /home/Tommy/testfile_Crontab.txt
[root@localhost ~]# _
```

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:

  <a b c d e> [user name] </directory/command> [output]

The parts of a cron command are:

1. The first five fields `a b c d e` specify the time/date and recurrence of the job.
2. User name:
3. In the second section, the `/directory/command` specifies the location and script you want to run.
4. The final segment `output` is optional. It defines how the system notifies the user of the job completion.

Lập lịch bằng crontab

1. Cron Time Format

Field	Possible Values	Syntax	Description
a-Minute	0 – 59	**7 * * * ***	The cron job is initiated every time the system clock shows 7 in the minute's position.
b-Hour	0 – 23	**0 7 * * ***	The cron job runs any time the system clock shows 7am (7pm would be coded as 19).
c-Day	0 – 31	**0 0 7 * ***	The day of the month is 7 which means that the job runs every 7th day of the month.
d-Month	0 = none and 12 = December	**0 0 0 7 ***	The numerical month is 7 which determines that the job runs only in July.
e-Day of the Week	0 = Sunday and 7 = Sunday	**0 0 * * 7**	7 in the current position means that the job would only run on Sundays.

Lập lịch bằng crontab

2. Command to Execute

- Specifying the command to execute.
- It represents the exact directory and filename of the script or commands you want cron to complete.

For example:

```
/root/backup.sh
```

=> the command looks at the root directory of the system and runs the ***backup.sh*** script. You may specify any script or command you wish.

Lập lịch bằng crontab

3. Output (Optional)

- By default, cron sends an email to the owner of the crontab file when it runs.
- This is a convenient way to keep track of tasks.
- Regular or minor tasks can fill up your inbox quickly.
- To turn off email output, add the following string, >/dev/null 2>&1, after the timing and command fields.

Example:

* * * * * directory/command >/dev/null 2>&1

Lập lịch bằng crontab

4. Using Operators (Optional)

- **An asterisk (*)**: stands for all values. Use this operator to keep tasks running during all months, or all days of the week.
- **A comma (,):** specifies separate individual values.
- **A dash (–):** indicates a range of values.
- **A forward-slash (/):** is used to divide a value into steps.

Example:

- */2 would be every other value,
- */3 would be every third,
- */10 would be every tenth, etc.

Lập lịch bằng crontab

- Các từ khóa thông dụng:

Parameter	Value of Parameters	Meaning
@reboot		Chạy một lần mỗi khi khởi động lại
@yearly	0 0 1 1 *	Chạy một lần mỗi năm
@annually		Tương tự @yearly
@monthly	0 0 1 * *	Chạy mỗi tháng một lần
@weekly	0 0 * * 0	hạy mỗi tuần một lần
@daily	0 0 * * *	Chạy một lần mỗi ngày
@midnight		Tương tự @daily
@hourly	0 * * * *	Chạy một lần mỗi giờ

Lập lịch bằng crontab

Ví dụ

```
// 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”
```

Setting Up a Cron Job

- Step 1: Create a new crontab: `crontab -e`
- Step 2: Editing is similar to “vi”: