$Linux#

Chương 5. Quản lý tiến trình

- Process (Tiến trình)
- Tasking (Tác vụ)
- PipeLine (Đường ống)

5.1 Quản lý tiến trình

- **Tiến trình ?**
- **Trạng thái của tiến trình**
- **Các thao tác trên tiến trình**

Tiến trình?

- **Thảo luận (ôn lại Nguyên lý HĐH)**
  - Tiến trình là gì
  - Các thông tin của tiến trình
  - Trạng thái của tiến trình
  - Tác vụ (tiểu trình)
  - Phân biệt Chương trình/ Tiến trình/ Tiểu trình

  #Quản lý danh sách tiến trình

  # Cấp phát tài nguyên cho tiến trình

Tiến trình

- Trạng thái của tiến trình
  - Đang thực thi
  - Tạm dừng/ khôi phục
  - Khóa/ bỏ khóa/ hủy
  - Kết thúc

Các lệnh quản lý tiến trình (tr75)

ps -ux	hiển thị tất cả các tiến trình thực hiện bởi người dùng
ps T	hiển thị các tiến trình dang chạy bởi thiết bị đầu cuối hiện thời của người dùng
ps aux	hiển thị tất cả các tiến trình trên hệ thống

#ps manpage // xem chi tiết các tùy chọn

***ps** accommodates UNIX-style and BSD-style arguments*
usage: ps -[Unix98 options] ps [BSD-style options] ps --[GNU-style long options] ps --help for a command summary

Summary of options
-a show all processes for the current user linked to a tty (*except* the session leader) **-e** or **-A** show all processes **-f** gives the PPID (Parent Process ID) and the STIME (Start Time) **-l** is similar to **-f** and displays a long list **a** show all processes linked to a tty, including other users **x** show all processes without a controlling tty as well

Quản lý tiến trình

#kill / #killall SIGNAL nprocess_NAME

Dừng các tiên trình

Lệnh **kill** sẽ gửi các tín hiệu đến các tiến trình. Có tổng cộng 63 tín hiệu. Tín hiệu mặc định dừng một tiến trình được gọi là SIGTERM với giá trị 15.

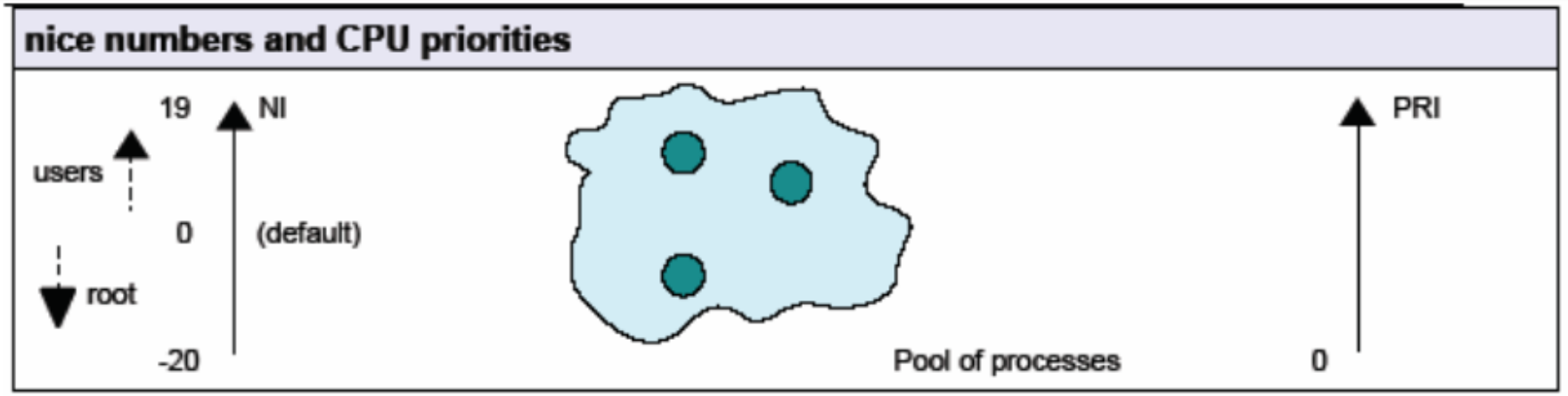
kill

cú pháp

kill SIGNAL process_PID

Mọi tiến trình có thể lựa chọn nhận hay không nhận một tín hiệu ngoại trừ SIGKILL sẽ được thực hiện bằng nhân hệ thống. Các daemon sẽ hiểu SIGUP có nghĩa là "đọc lại file cấu hình"

Quản lý ưu tiên giữa các tiến trình



Sử dụng lệnh **renice** để thay đổi mức độ ưu tiên của một tiến trình. Dùng lệnh nice để thiết lập mức độ ưu tiên của một tiến trình.

Cú pháp

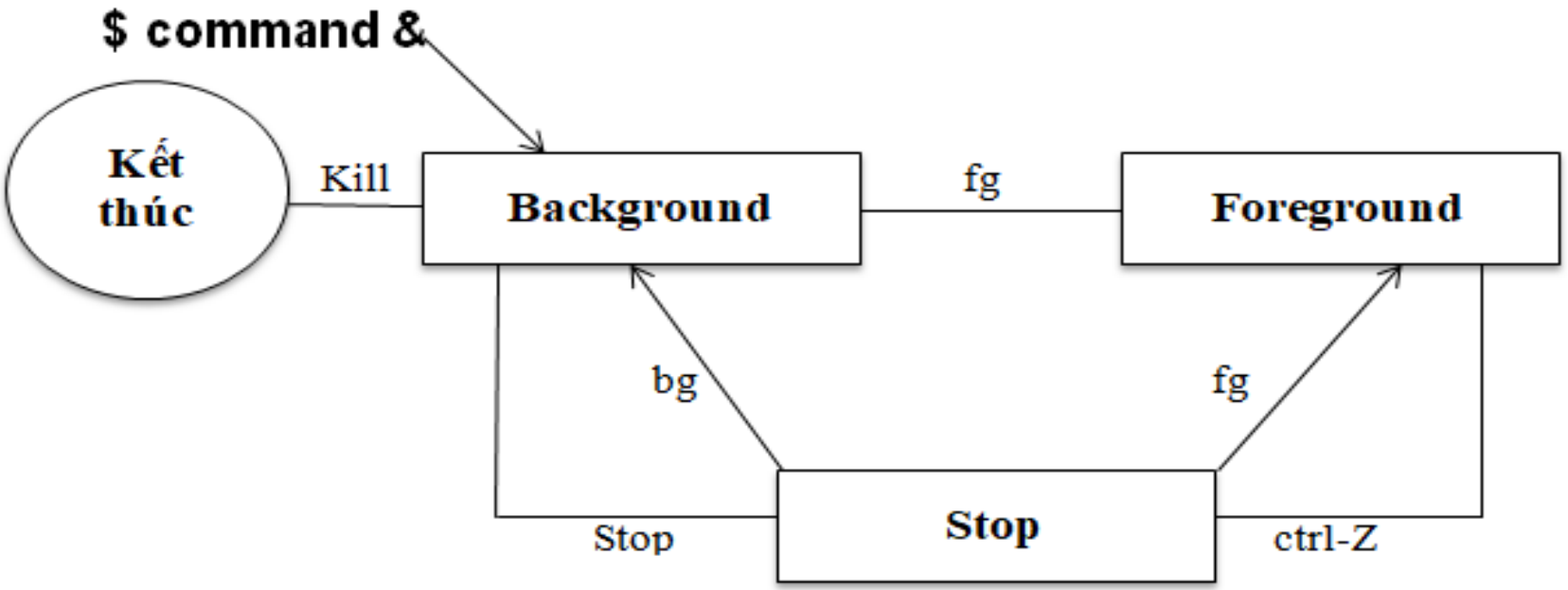
```
Nice -<NI> <process>
renice <+/-NI> -p <PID>
```

Quản lý cây tiến trình (#pstree)

pstree // quản lý cây tiến trình

```
bash(1046)---xinit(1085)-+-X(1086)
                         `-xfwm(1094)-+-xfce(1100)---xterm(1111)---bash(1113)-+-pstree(1180)
                                      |                                       |-soffice.bin(1139)---soffice.bin(1152)-+
                                                                                                                      -soffice.bin(1153)
                                      |                                       |                                       |-soffice.bin(1154)
                                      |                                       |                                       |-soffice.bin(1155)
                                      |                                       |                                       |-soffice.bin(1156)
                                      |                                       |                                       `-soffice.bin(1157)
                                      |                                       `-xclock(1138)
                                      |-xfgnome(1109)
                                      |-xfpager(1108)
                                      |-xfsound(1107)
                                      `-xscreensaver(1098)
```

5.2 Quản lý tác vụ



Thảo luận:

-Tác vụ

-Quản lý tác vụ

Dừng và bắt đầu các công việc (job)

- $xclock // tạm bỏ dấu nhắc lệnh shell
- $bg //khôi phục dấu nhắc lệnh shell
- $hub
- $nohub

Lệnh quản lý tiến trình và tác vụ

- **#ps –A**: Kiểm tra mọi tiến trình trong hệ thống.
- **#ps -U root -u root –N**: Kiểm tra mọi tiến trình ngoại trừ những tiến trình hệ thống.
  #ps -u username: Kiểm tra những tiến trình được thực hiện bởi một người dùng nhất định.
- Hoặc bạn có thể sử dụng lệnh **#top** để xem những tiến trình đang chạy trên hệ thống trong thời gian thực.
- **# isostat** Theo dõi Average CPU Load và Disk Activity
- **# iostat –n #uptime**
- **#w username** // Kiểm tra nhật ký đăng nhập hệ thống và tác vụ thực hiện**#vmstat [-m, a-, n]** // kiểm soát hành vi hệ thống, n = 0,1,2**#kill / #killall/ #nice**

Một số ví dụ về tiến trình

Tạm dừng tiến trình

Sử dụng phím **Ctrl - Z** để đưa một tiến trình đang chạy ở tiền cảnh vào chạy ở hậu cảnh. Khi một tiến trình nhận được tín hiệu **Ctrl - Z** nó sẽ bị hệ thống cho tạm dừng và đưa vào hậu cảnh. Dấu nhắc hệ thống được trả lại cho người dùng. Tuy đưa vào hậu cảnh nhưng tiến trình đang bị tạm dừng, nó chỉ thực sự chạy lại ở hậu cảnh khi bạn cho phép . Ví dụ:

```
$ls -R / >allfiles.txt
^Z
[1]+ Stopped      ls -R / >allfiles.txt
$
```

Muốn xem PID của tiến trình bạn gọi lệnh **ps -af**

Đánh thức tiến trình

Sử dụng lệnh **jobs** để kiểm tra chương trình của ta đang dừng hay đang chạy.

```
$jobs
[1]+ Stopped      ls -R / >allfiles.txt
```

Lệnh **jobs** hiển thị trạng thái của tất cả các tiến trình đang chạy ở hậu cảnh. Như kết quả trên: tác vụ **[1]** đang ở trạng thái dừng. Để yêu cầu tiến trình của ta tiếp tục hoạt động ở hậu cảnh: sử dụng lệnh **bg**.

```
$bg 1
Ls -R / >allfiles.txt
$jobs
[1]+ Running      ls -R />allfiles.txt &
```

Dùng lệnh **fg** để mang tiến trình trở lại hoạt động ở phía tiền cảnh.

```
$fg 1
Ls -R / >allfiles.txt
```

Lệnh hủy tiến trình

Hủy tiến trình

Không phải lúc nào tiến trình cũng hoạt động tốt đẹp. Có thể chúng sẽ bị treo hoặc bước vào vòng lặp vô tận và không bao giờ chấm dứt. Trong trường hợp này, ta cần phải loại bỏ chương trình ra khỏi hệ thống. Lệnh **kill** của Linux thường được dùng cho mục đích này, **kill** yêu cầu cung cấp mã số định danh PID của tiến trình. Lệnh **kill** thường dùng chung với lệnh **ps -af**. Bạn dùng lệnh **ps -af** để xem thông tin về tiến trình đang chạy, sau đó lấy PID của tiến trình cần hủy và gọi lệnh **kill**.

```
$ls -R / >data.txt
^Z
$ps -af
PID     TTY        TIME       CMD
128     tty1     00:00:00    bash
137     pts/9    00:00:00    ls -R
235     pts/0    00:00:00    bash
…
$kill 137

$ps -af
PID     TTY        TIME       CMD
128     tty1     00:00:00    bash
235     pts/0    00:00:00    bash
…
```

Có một số tiến trình có độ ưu tiên cao và không thể loại bỏ theo cách thông thường. Lúc này ta sử dụng kill ở cấp độ -9. Ví dụ:

```
$kill -9 137
```

Các thao tác trên tác vụ

- Ví dụ:
- *$ emacs &*

[1] 756

$ stop 756

 # or $ stop °/ol $ bg 756

 # or $ bg 7,1

$ kill 756

 # or $ kill 7.1

5.3 Cơ chế đường ống

- **Đầu vào và đầu ra chuẩn của tiến trình**

Thav vì sử dụng đầu vào và đầu ra mặc định, có thể thực hiện chuyển hướng đầu vào và đầu ra.

- Chuyển hướng đầu vào chuẩn (<) $ tee < test.txt
- Chuyển hướng đầu ra chuẩn (>, »)

$ ls > /dev/lp $ ls » test.txt

- Chuyển hướng kênh báo lỗi

$ rm prog.c 2> /dev/null

$ gcc prog.c 2 » erreur.txt

Thực hiện song song hai câu lệnh

cmd 1 && cmd2

- *cmdl I cmd2*
- *$cat /etc/passwd I grep trunghq*

Câu lệnh tee Câu lệnh tee cho phép copy đầu ra chuẩn thành 2 đầu ra khác nhau:

- *ls -1 I tee test I more*

Thảo luận & Hỏi đáp

1. Tiến trình là gì; trạng thái tiến trình?
2. Các lệnh quản lý tiến trình
3. Lệnh $kill / $killall
4. Quy trình quản lý tác vụ ($bg/ $ stop)
5. Đầu vào/ra chuẩn (stdin/ stdout)
6. Cơ chế đường ống là gì $xclock