**Quản lý tiến trình Linux**

* Giới thiệu về tiến trình
* Là một chương trình chạy trên hệ thống
* Được định danh bởi 1 PID (tiến trình ID) phục vụ cho việc giám sát, điều khiển, giám sát hệ thống
* Có 3 loại tiến trình chính trên Linux:
* Tiến trình với đối thoại (Interactive processes) : là tiến trình khởi động và quản lý bởi shell, kể cả tiến trình forthground hay background.
* Tiến trình batch (Batch processes) : Tiến trình không gắn liền đến bàn điều khiển (terminal) và được nằm trong hàng đợi để lần lượt thực hiện.
* Tiến trình ẩn trên bộ nhớ (Daemon processes) : Là các tiến trình chạy dưới nền (background). Các tiến trình này thường được khởi động từ đầu. Đa số các chương trình server cho các dịch vụ chạy theo cách này.
* Tiến trình cha và tiến trình con
* Khi một tiến trình sinh ra một tiến trình khác thì tiến trình ban đầu được gọi là tiến trình cha và được định danh bởi PPID (Parent PID) và tiến trình mới khi được sinh ra là tiến trình con
* Sự tương tác giữa tiến trình con và tiến trình cha
* Khi tiến trình con đang chạy thì tiến trình cha sẽ chờ
* Khi tiến trình con hoàn thành thì tiến trình cha sẽ tiếp tục thực thi và tiến trình con sẽ được kết thúc
* Các trạng thái của tiến trình

|  |  |
| --- | --- |
| Trạng thái | Mô tả |
| Run | Tiến trình đang chạy |
| Sleep | Tiến trình đang ở trạng thái chờ hoạt động |
| Zombie | Tiến trình mà tiến trình cha đã kết thúc |
| Stop | Tiến trình sẽ dừng hoạt động |

VD: Khi chúng ta vào siêu thị đi để shopping thì tiến trình chúng ta shopping là trạng thái Run, còn trong khi chúng ta lựa chọn đồ để shopping thì tiến trình chúng ta lựa chọn đó sẽ là trạng thái Sleep vì nó làm cho tiến trình shopping của chúng ta phải chờ.

Còn sau khi shopping xong thì chúng ta sẽ phải đi thanh toán hóa đơn ở quầy lễ tân thì lúc này tiến trình chúng ta đợi quầy lễ tân thanh toán sẽ là trạng thái Zombie vì tiến trình shopping của chúng ta đã xong nhưng vẫn phải đợi tiến trình thanh toán vẫn chưa xong. Cuối cùng sau khi thanh toán hóa đơn xong để đi về thì lúc này sẽ là trạng thái Stop hay hiểu là tiến trình shopping của chúng ta đã kết thúc và đã dừng lại

* Quản lý tiến trình

Vì có những tiến trình chạy do code kém nên tiến trình đó sẽ chạy CPU rất nhiều hoặc chạy memory rất nhiều. Ví dụ chạy CPU rất nhiều có thể do khả năng tính toán rất nhiều, hoặc chạy memory rất nhiều là do người code note quá nhiều khi chạy chương trình, thì trong những trường hợp đó, chúng ta phải kiểm soát để tiến trình đó chạy theo đúng lịch trình và không làm tốn quá nhiều tài nguyên

* Việc quản lý tiến trình nhằm
* Xác định những tiến trình đang chạy trên hệ thống
* Trạng thái của các tiến trình
* Tài nguyên mà các tiến trình đang sử dụng
* Kiểm soát tiến trình
* Kết thúc các tiến trình không mong muốn
* Thực thi các nghiệp vụ theo lịch
* Các lệnh quản lý tiến trình
* ps – ef: liệt kê các tiến trình đang hoạt động trên hệ thống

|  |  |
| --- | --- |
| Thông số | Ý nghĩa |
| UID | User quản lý tiến trình |
| PID | ID tiến trình |
| PPID | ID tiến trình cha |
| STIME | Start time |
| C | CPU đang thực thi |
| TIME | Thời gian thực thi của tiến trình |
| CMD | Câu lệnh thực thi |

* pgrep: Hiển thị các tiến trình một chương trình (Chúng ta sẽ lấy ví dụ về java về các nền tảng của Linux or Unix đều chạy bằng java là chính)
* top: Hiển thị trạng thái các tiến trình đang hoạt động trên hệ thống

- Câu lệnh này có thể nói là rất là hiệu năng, được dùng rất nhiều khi chúng ta quản trị hệ điều hành Linux vì nó sẽ hiển thị hết các thông số của CPU, Memory…

|  |  |
| --- | --- |
| Thông số | Ý nghĩa |
| PID | Tiến trình ID của các tiến trình |
| User | User thực thi tiến trình |
| %CPU | %CPU mà tiến trình đang sử dụng |
| %Mem | Dung lượng RAM mà tiến trình đang sử dụng |
| Time+ | Thời gian hoạt động của tiến trình từ khi bật |
| Command | Chương trình chạy |

* kill: Giả sử tiến trình đang bị treo, không thể stop theo lệnh được hay tiến trình chiếm quá nhiều tài nguyên của hệ thống thì chúng ta phải sử dụng lệnh kill để tránh làm ảnh hưởng đến KPI của hệ thống.

- Kill: Gửi tín hiệu tới một tiến trình theo PID (kill – 9 vs kill – 15)

* kill – 9: Kết thúc tiến trình ngay lập tức
* kill – 15: gửi tín hiệu kết thúc đến tiến trình, chờ tiến trình thực hiện cleanup và kết thúc (Trong một số trường hợp k – 15 không thể tắt được tiến trình, buộc phải sử dụng kill – 9 để dừng tiến trình)
* pkill: Kết thúc một tiến trình hoặc nhiều tiến trình theo tên hoặc thuộc tiến của tiến trình. Ngoài ra khi chúng ta sử dụng lệnh này thì phải rất cẩn trọng, bởi vì nó sẽ kill tất cả các tiến trình của chúng ta.
* Một số lệnh khác:

1. Lệnh date/ cal:

- date: hiển thị cho bạn xem chính xác thời gian hiện tai trong máy.

- cal: hiển thị ngày tháng năm dưới dạng lịch dọc.

2. Lệnh mkdir: tạo thư mục mới

3. Lệnh kiểm tra thư mục: ls

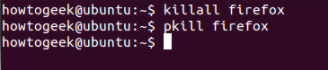
4. Lệnh xem thông tin cấu hình máy: cat /proc/cpuinfo

5. pstree: Liệt kê theo một danh sách các tiến trình theo hình cây

6. pgrep vs kill: Lệnh này cho phép bạn tìm PID của một tiến trình trên hệ thống



7. pkillall vs pkill



* Đặt lịch tiến trình chạy tự động: Có một cái rất là tuyệt vời trong hệ điều hành Linux/Unix đó là chúng ta có thể đặt lịch cho tiến trình chạy tự động. Giúp chúng ta giảm thiểu thời gian chúng ta tác động vào hệ thống. Nhất là khi dù chúng ta không thể online nhưng vẫn có thể chạy chương trình hệ thống khi hệ thống bắt chúng ta online.
* Mục đích
* Thiết lập lịch cho một tác vụ được thực hiện vào một thời điểm xác định
* Giúp tự động hóa các công việc mang tính lặp lại
* Hỗ trợ cho việc vận hành, giám sát hệ thống
* Có 2 kiểu đặt để chạy tự động
* At
* Đặt lịch cho 1 lệnh hoặc 1 tiến trình thực thi 1 lần duy nhất vào 1 thời điểm xác định
* Cú pháp:
* At time [date]
* Lệnh kiểm tra
* Atq (Sau khi chúng ta đặt lịch cho nó xong thì chúng ta có thể kiểm tra xem nó có lịch như thế nào)
* At – I (Dùng để liệt kê chi tiết hơn)
* Hủy 1 job đã lập bằng lệnh at -r (Để hủy 1 câu lệnh nào đấy mà chúng ta đặt nhầm hoặc đặt sai)
* Thiết lập quyền sử dụng at trong file /etc/at.deny,/etc/at.allow (Cho phép 1 ai đó có quyền được phép chạy at và không được chạy at bằng at.deny và at.allow)
* Crontab
* Đặt lịch cho 1 lệnh hoặc 1 tiến trình thực thi lặp lại nhiều lần
* Nguyên lý hoạt động (Khi mà khởi động thì chúng ta sẽ có 1 cái aimon là cron chạy và nó sẽ thực thi các dòng trong file crontab
* Sử dụng daemon cron
* Lịch thực thi được lưu trong file crontab
* Daemon cron đọc file trong crontab để thực thi theo lịch và câu lệnh cấu hình trong crontab
* Định dạng file Crontab (Khi chúng ta khai báo file crontab thì chúng ta phải khai báo theo định dạng)
* Các lệnh quản lý
* Crontab – I: Liệt kê các crontab của user chạy lệnh (Sau khi đặt lịch xong rồi thì chúng ta có thể kiểm tra)
* Crontab – e: Sửa crontab
* Thiết lập quyền sử dụng Crontab (Ngoài ra chúng ta có thể cấp quyền chạy cho 1 ai đó giống như kiểu at)
* /etc/cron.allow: Khai báo các user được phép đặt allow
* /etc/cron.deny: Khai báo các user không được phép đặt allow
* Lưu ý: Trong Crontab cần đặt đường dẫn đầy đủ cho câu lệnh