Quản lý process

Giảng Viên: Lưu An Phú

Agenda

- Process identifier
- Process được sinh ra và kết thúc như thế nào
- Thay đổi user id vào group id
- Các file thông dịch
- Run command line trong C
- Tracking trạng thái kết thúc của process

Process identifier

- A unique number
- Used to identify process
- Some numbers is reserved for special process
  - Swapd process
  - Init process

Process được sinh ra và kết thúc như thế nào

fork function

- pid_t fork(void)
- Tạo ra 1 bản sao cho bộ nhớ hiện tại của process
  - Copy on write
- PID thật sẽ được trả cho process con
- PID 0 sẽ trả về cho process cha
- Con có thể chạy và hoàn thành công việc trước cha

fork function

- Thực hành:
  - Viết 1 chương trình C fork ra cha và con, trong process cha in ra pid và "I am parent", trong process con làm tương tự. So sánh PID in ra và PID sử dụng lệnh PS
  - Thử mở 1 file trước khi fork, sau đó cả cha và con cùng ghi vào file fd trước đó và check kết quả
  - Thử với biến thông thường và xem kết quả

exec function

- int execl(const char *pathname, const char *arg0, ... /* (char *)0 */)
- Thay the toan bo khong gian bo nhớ của process cha thanh process con
- Có thể khởi tạo 1 chương trình mới
- Sau khi gọi hàm exec, toàn bộ source code phía sau của chương trình sẽ không được thực hiện nữa

exit function

- Cho dù process chủ động hoặc bị động kết thúc, 1 giá trị trạng thái kết thúc luôn được trả về cho process cha.
- Sau khi gọi hàm exit và gửi trạng thái về cho process cha, process con chờ đợi tín hiệu xác nhận từ cha.
- Zombie process.

wait function

- pid_t wait(int *wstatus)
- Khi 1 process con kết thúc, ngoài exit status, kernel còn gửi SIGCHILD signal cho process cha
- Block process cho đến khi 1 trong các process con kết thúc

wait function

- Thực hành: Viết 2 chương trình cha và con. Chương trình cha gọi con với tham số truyền vào là đường dẫn đến file. Chương trình con ghi hello world vào file và trả về trạng thái kết quả cho cha. Cha phải biết kết quả thực hiện của chương trình con.

User id and group id

Thay đổi user id vào group id

- int setuid(uid_t uid)
- int setgid(gid_t gid)
- Khi user-id và group-id thay đổi thì quyền của process cũng thay đổi tương ứng
- Sẽ set theo xu hướng giảm quyền của process

Thay đổi user id vào group id

- Thực hành: Dúng user A để call chương trình C, trong chương trình C chuyển sang user B, sau đó chương trình sẽ tạo file mới và file đó phải thuộc quyền sở hữu là user B.

Các file thông dịch

- Thông dịch
- Biên dịch
- Header của file
  - #! pathname [optional-argument]
  - #! /bin/sh
    - exec header
- Set quyền execute cho file thông dịch

Các file thông dịch

- Thực hành:
  - Copy script trong slide và chạy thử.
  - Viết 1 chương trình C có tên là HocjLinux_sh và dùng nó thông dịch cho script trên

```
1 #!/bin/my_bash
2
3 echo "Hello world I am $1"
```

Run command line trong C

- int system(const char *command)
- Cho phép thực hiện 1 câu lệnh trong C
- Fork chương trình hiện tại và dùng chương trình con để exec chương trình

Tracking trạng thái kết thúc của process

- Đôi khi trong hệ thống có process bị terminate mà không biết lý do.
- Enable tính năng này khi build kernel sẽ cho phép tracking trạng thái kết thúc của process
- Keyword: Process Accounting

Bài tập

- Viết 1 chương trình C sử dụng system() để up/down và set ip cho port mạng.
- Viết 1 chương trình sử dụng hàm fork để tạo ra chương trình con, trong chương trình con run câu lệnh ls và ghi out vào 1 file. Chương trình cha check trạng thái trả về nếu là success thì đọc từ file và in ra màn hình kết quả.

Thank you