Họ và tên : Trần Đình Khang

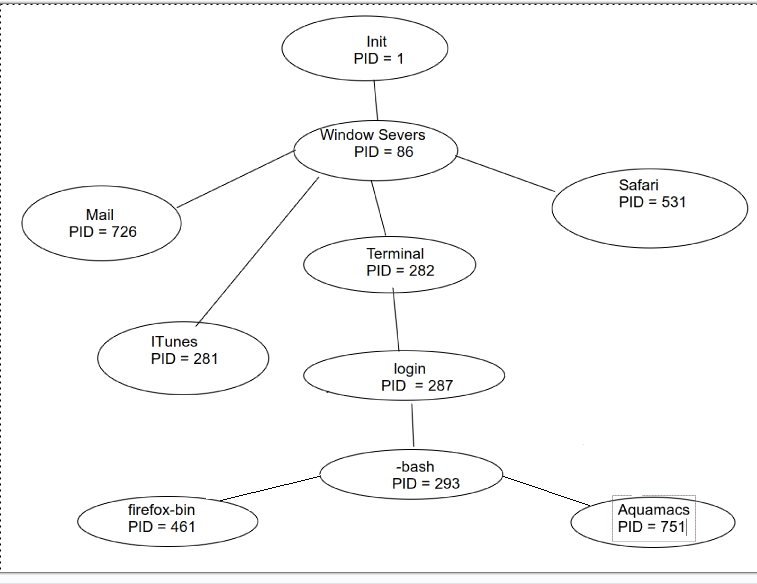
MSSV: 18520072

Mã Lớp: IT007.K21.KHTN

Bài thực hành Lab03

Bài 1:

1. **Vẽ cây quan hệ parent-child cho các tiến trình:**

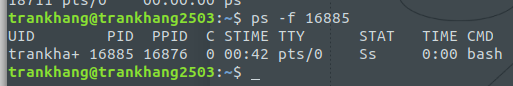


1. **Trình bày cách sử dụng lệnh ps để tìm tiến trình cha của một một tiến trình dựa vào PID của nó.**

**+ Cách 1:**

- Gõ ps -f <PID process>

- Màn hình sẽ hiển thị như sau:



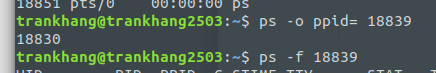
- Số hiển thị dưới PPID chính là PID của process cha.

Ví dụ trong hình PPID = 16876

**+ Cách 2**:

- Gõ ps -o ppid= <PID process>

- Số hiển thị trên màn hình chính là PPID



1. **Tìm hiểu và cài đặt lệnh pstree (nếu chưa được cài đặt), sau đó trình bày cách sử dụng lệnh này để tìm tiến trình cha của một tiến trình dựa vào PID của nó.**

- Gõ lệnh pstree -s -p <PID process>

- Màn hình sẽ hiển thị có dạng 1 chuỗi có số PID.

Screenshot from 2020-05-08 01-37-30

- Số PID của process bên trái của process nhập chính là số PID của process cha.

Bài 2:

- Chương trình sẽ in ra: “I see 17 coconuts!”

- Vì khi gọi lệnh pid = fork(), thì tiến trình cha P0 sinh ra tiến trình con P1 thông qua lời gọi hàm fork(). Khi đó pid P1 = 0.

+ Trong phạm vi lệnh if pid == 0 , thì pid P1 thỏa điều kiện nên gán num\_coconuts = 42. Sau đó exit(0), tiến trình con sẽ vào trạng thái zombie và thật sự kết thúc khi tiến trình cha gọi hàm wait(NULL).

+ Sau khi tiến trình con kết thúc, num\_coconuts được hoàn trả giá trị nên num\_coconuts = 17. Nó sẽ in ra “I see 17 coconuts!” . Lệnh exit(0) kết thúc tiến trình cha P0.

Bài 3:

- Thuộc tính tiểu trình:

|  |  |  |
| --- | --- | --- |
| Thuộc Tính | Giá trị mặc định | Ý nghĩa |
| Guradsize | PAGEIZES | Kích thước đảm bảo cho tiểu trình không dùng quá không gian được cấp phát |
| Scope | PTHREAD\_SCOPE\_PROCE  SS | Dùng tài nguyên trong phạm vi cho phép của tiến trình |
| Detachstate | PTHREAD\_CREATE\_JOIN  ABLE | Tiểu trình được hợp với các tiến trình khác |
| Stackaddr | NULL | Tiểu trình mới có địa chỉ trong system-allocated stack |
| SatckSize | NULL | Tiểu trình tới sẽ có kich thước do system quy định |
| Inheritsched | PTHREAD\_INHERIT\_SCHED | Tiểu trình con sẽ thừa kế lịch độ ưu tiên của tiểu trình cha |
| SchedPolicy | SCHED\_OTHER | Tiểu trình sẽ chạy tuân theo độ ưu tiên của tiểu trình |

Thiết lập thuộc tính cho 1 tiểu trình:

+ Sử dụng lệnh biến đặc biệt ‘attr’ với kiểu ‘pthread\_attr\_t\*’

+ Dùng lệnh pthread\_attr\_init(&attr) để reset lại giá trị mặt định tất cả cùng một thuộc tính mà được chọn.

+ Gọi hàm thuộc tính :

pthread\_attr\_set(detachstate/Inheritsched/SchedPolicy/scope/....)()

+ Dùng pthread\_attr\_destroy(): để hủy thuộc tính không cần thiết.



Đoạn chương trình trên dùng lệnh pthread\_attr\_setguardsize để điều chỉnh guardsize

Bài 4:

