BÁO CÁO THỰC HÀNH

**Môn học: Hệ điều hành (IT007)**

**Lab 03: TIẾN TRÌNH VÀ TIỂU TRÌNH**

*GVHD: Phạm Quốc Hùng*

1. **THÔNG TIN CHUNG:**

*(Liệt kê tất cả các thành viên trong nhóm)*

Lớp: IT007.N25.1

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Lê Đoàn Trà My | 21521149 | 21521149@gm.uit.edu.vn |

1. **NỘI DUNG THỰC HIỆN:[[1]](#footnote-1)**

|  |  |  |
| --- | --- | --- |
| **STT** | **Công việc** | **Kết quả tự đánh giá** |
| 1 | Câu 1: Mối quan hệ cha-con giữa các tiến trình | Hoàn thành tốt, 100% |
| 2 | Câu 2: | Hoàn thành tốt, 100% |
| 3 | Câu 3: | Hoàn thành tốt, 100% |
| 4 | Câu 4: | Hoàn thành tốt, 100% |
| Điểm tự đánh giá | | 9.5 - 10 |

**Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.**

BÁO CÁO CHI TIẾT

**1. Mối quan hệ cha-con giữa các tiến trình**

***a.*** *Vẽ cây quan hệ parent-child của các tiến trình bên dưới:*

A picture containing text, screenshot, font, number

Description automatically generated

*Hình 1. Quan hệ giữa các tiến trình*

- Trả lời:

A picture containing diagram, text, drawing, circle

Description automatically generated

*Hình 2. Cây quan hệ giữa các tiến trình*

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

- Sử dụng lệnh **ps** để xem các tiền trình đang hoạt động trên hệ thống:

A screenshot of a computer

Description automatically generated with low confidence

*Hình 3. Các tiến trình đang chạy*

- Sử dụng lệnh **ps -f** để xem thông tin các tiến trình cha của các tiến trình đang chạy (PPID: PID tiến trình cha của tiến trình đang xét):

A screenshot of a computer

Description automatically generated with medium confidence

*Hình 4. Các thông tin tiến trình cha của các tiến trình đang chạy*

- Để xem PID của tiến trình 2503: **ps -f 2503**

A picture containing text, screenshot, font

Description automatically generated

*Hình 5. Thông tin tiến trình cha (PPID) của tiến trình có PID = 2503*

Vậy, tiến trình cha của tiến trình có PID = 2503 là tiến trình có PID = 2485.

***c****. 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.*

*- Sử dụng lệnh pstree -s -p <PID tiến trình đang xét>*

**

*Hình 6. Sơ đồ cây các tiến trình, cha của bash (PID = 2503) là gnome-terminal (PID =2485)*

**2. Chương trình bên dưới in ra kết quả gì? Giải thích tại sao?**

A picture containing text, screenshot, document, font

Description automatically generated

*Hình 7. Nội dung chương trình bài 2 lab3*

*- Kết quả in ra:* **I see 17 coconuts!**

**A screenshot of a computer screen

Description automatically generated with medium confidence**

*Hình 8. Nội dung file thực thi và kết quả chạy của bài 2 lab3*

*- Giải thích:*

+ Ban đầu, khởi tạo biến pid và khởi tạo biến num\_coconuts có giá trị 17.

+ Chạy lệnh fork() để tạo một tiến trình mới. (*Tiến trình gọi hàm fork() là tiến trình cha, tiến trình mới được tạo là tiến trình con. Tiến trình cha quay lại việc thực thi và tiến trình con bắt đầu thực thi tại cùng 1 nơi).*

+ Khi đó, tiến trình cha có PID > 0; tiến trình có PID = 0.

+ Tiến trình cha và tiến trình con thực thi đoạn code tiếp theo:

|  |  |
| --- | --- |
| Tiến trình cha | Tiến trình con |
| - Vì PID > 0, thực thi lệnh wait(NULL), nghĩa là đợi cho tiến trình con kết thúc.  - Sau khi tiến trình con kết thúc, tiến trình cha thực hiện lệnh printf(“I see %d coconuts!\n, num\_coconuts”); với num\_coconuts = 17 được gán ban đầu (tiến trình cha và con có bộ nhớ riêng).  - exit(0); kết thúc và thoát tiến trình cha. | - Vì PID = 0, biến num\_coconuts được gán bằng 42.  - exit(0) thoát khỏi tiến trình con mà không thực thi lệnh printf. |

**3. Trong phần thực hành, các ví dụ chỉ sử dụng thuộc tính mặc định của pthread, hãy tìm hiểu POSIX thread và trình bày tất cả các hàm được sử dụng để làm thay đổi thuộc tính của pthread, sau đó viết các chương trình minh họa tác động của các thuộc tính này và chú thích đầy đủ cách sử dụng hàm này trong chương trình.**

*- Các hàm được sử dụng để thay đổi thuộc tính của pthread:*

+ pthread\_attr\_init(pthread\_attr\_t \*attr)

+ pthread\_attr\_destroy(pthread\_attr\_t \*attr);

+ pthread\_attr\_getdetachstate(const pthread\_attr\_t \*attr,int \*detachstate);

+ pthread\_attr\_setdetachstate(pthread\_attr\_t \*attr, intdetachstate);

+ pthread\_attr\_getguardsize(const pthread\_attr\_t \* restrictattr,size\_t \* restrict

guardsize);

+ pthread\_attr\_setguardsize(pthread\_attr\_t \*attr, size\_tguardsize);

+pthread\_attr\_getinheritsched(const pthread\_attr\_t \* restrictattr,int \* restrict

inheritsched);

+ pthread\_attr\_setinheritsched(pthread\_attr\_t \*attr, intinheritsched);

+pthread\_attr\_getschedparam(const pthread\_attr\_t\*restrictattr,struct sched\_param \*

restrict param);

+ pthread\_attr\_setschedparam(pthread\_attr\_t \*attr,const struct sched\_param \*param);

+ pthread\_attr\_getschedpolicy(const pthread\_attr\_t \* restrict attr,int\* restrict policy);

+pthread\_attr\_setschedpolicy(pthread\_attr\_t \*attr, intpolicy);

+ pthread\_attr\_getscope(const pthread\_attr\_t \* restrict attr, int \* restrict

contentionscope);

+ pthread\_attr\_setscope(pthread\_attr\_t \*attr, intcontentionscope);

+ pthread\_attr\_getstack(const pthread\_attr\_t \* restrict attr,void \*\* restrict stackaddr,

size\_t \* restrict stacksize);

+ pthread\_attr\_setstack(pthread\_attr\_t \* restrict attr, void \*stackaddr,size\_t stacksize);

+ pthread\_attr\_getstacksize(const pthread\_attr\_t \* restrict attr,size\_t\* restrict

stacksize);

+ pthread\_attr\_setstacksize(pthread\_attr\_t \*attr, size\_t stacksize);

*- Các chương trình:*

*A screenshot of a computer screen

Description automatically generated with medium confidence*

*Hình 9. Nội dung file thực thi và kết quả chạy của init và destroy*

*A screenshot of a computer screen

Description automatically generated with medium confidence*

*Hình 10. Nội dung file thực thi và kết quả chạy của set và get detach state*

*A screenshot of a computer

Description automatically generated*

*Hình 11. Nội dung file thực thi và kết quả chạy của set và get guard size*

*A screenshot of a computer

Description automatically generated*

*Hình 12. Nội dung file thực thi và kết quả chạy của set và get Inherited Scheduling Policy*

*A screenshot of a computer screen

Description automatically generated with medium confidence*

*Hình 13. Nội dung file thực thi và kết quả chạy của set và get Scheduling parameters*

*A screenshot of a computer

Description automatically generated*

*Hình 14. Nội dung file thực thi và kết quả chạy của set và get Scheduling policy*

*A screenshot of a computer screen

Description automatically generated with medium confidence*

*Hình 15. Nội dung file thực thi và kết quả chạy của set và get scope*

*A screenshot of a computer

Description automatically generated*

*Hình 16. Nội dung file thực thi và kết quả chạy của set và get stack*

*A screenshot of a computer

Description automatically generated*

*Hình 17. Nội dung file thực thi và kết quả chạy của set và get stacksize*

**4. Viết chương trình làm các công việc sau theo thứ tự:**

a. In ra dòng chữ: “Welcome to IT007, I am <your\_Student\_ID>!”

b. Mở tệp abcd.txt bằng vim editor

c.Tắt vim editor khi người dùng nhấn CRTL+C

d. Khi người dùng nhấn CTRL+C thì in ra dòng chữ: “You are pressed CTRL+C! Goodbye!”

*- Nội dung file chương trình bài 4 (Ex4.c) và quá trình thực thi:*

*A screenshot of a computer

Description automatically generated*

*Hình 18. Nội dung file Ex4.c và kết quả thực thi câu a, b*

A screenshot of a computer

Description automatically generated

*Hình 19. Nội dung file Ex4.c và kết quả thực thi câu c*

A screenshot of a computer screen

Description automatically generated with medium confidence

*Hình 20. Nội dung file Ex4.c và kết quả thực thi câu d*

1. Ghi nội dung công việc, các kịch bản trong bài Thực hành [↑](#footnote-ref-1)