Bài thực hành lab 3

Họ tên: Võ Anh Kiệt

MSSV: 20520605

Mã lớp: IT007.M21.ANTN

Hướng dẫn

GCC

Text

Description automatically generated

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

Make file

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

Trình gỡ lỗi

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text

Description automatically generated

Hướng dẫn thực hành

Tiến trình

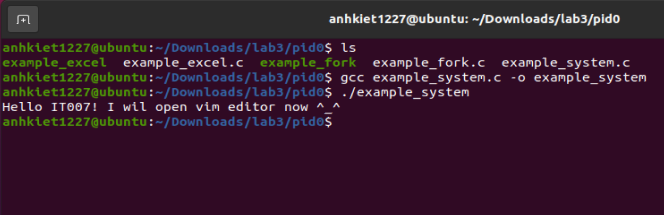
Tạo

Text

Description automatically generated

Text

Description automatically generated



Text

Description automatically generated

Kết thúc

Text

Description automatically generated

Tạo tiểu trình

Text

Description automatically generated

Text

Description automatically generated

Dừng tiểu trình

Graphical user interface, text

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

Hợp gỡ tiểu trình

Text

Description automatically generated

Truyền dữ liệu cho tiểu trình

Text

Description automatically generated

Signal

Text

Description automatically generated

Bài tập ôn tập

Câu 1:

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

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

Lệnh ps dùng để liệt kê chi tiết các tiến trình

Graphical user interface, text

Description automatically generated

Lệnh ps -f là lệnh liệt kê đầy đủ và chi tiết hơn lệnh ps

A screenshot of a computer

Description automatically generated with medium confidence

Các thông số được thể hiện

|  |  |
| --- | --- |
| **Cột** | **Mô tả** |
| UID | ID người sử dụng, mà tiến trình buộc phải sở hữu. |
| PID | ID của tiến trình. |
| PPID | ID của tiến trình cha. |
| C | CPU sử dụng tiến trình. |
| STIME | Thời gian bắt đầu tiến trình. |
| TTY | Kiểu terminal liên kết với tiến trình. |
| CMD | Lệnh bắt đầu tiến trình này. |

Những tiến trình có thể chạy song song với ps

|  |  |
| --- | --- |
| **Tùy chọn** | **Mô tả** |
| -a | Chỉ thông tin về tất cả người dùng. |
| -x | Chỉ thông tin các tiến trình mà không có termial. |
| -u | Chỉ thông tin thêm vào chức năng -f |
| -e | Hiển thị thông tin mở rộng |

Cách tìm tiến trình cha: check cột PID rồi sau đó nhìn qua cột PPID là cột chứa ID của tiến trình của tiến trình ID tiến trình cột PID

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

Lệnh pstree liệt kê tiến trình theo dạng cây

Tiến trình cha là tiến trình ở bên trái của một tiến trình

Tiến trình con là tiến trình nằm bên phải của một tiến trình

Ví dụ:

Systemd là tiến trình cha của ModemManager

ModemManager là tiến trình chả của 2\*[{ModemManager}]

Text

Description automatically generated

Sử dụng pstree –p để hiện thị pid của tiến trình

A picture containing text

Description automatically generated

Sau mỗi tiến trình sẽ có thông số pid trong ngoặc tròn

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



Đặt biến pid với kiểu dữ liệu pid\_t, sau đó gán biến num\_coconuts = 17 kiểu số nguyên. Tạo tiến trình pid = fork();

In ra dòng “I see 17 coconuts!” vì pid > 0 sẽ vào else {wait (NULL);} thì giá trị của coconuts vẫn là 17.

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. (Gợi ý các hàm liên quan đến thuộc tính của pthread đều bắt đầu bởi: pthread\_attr\_\*)

Các hàm thuộc tính

|  |  |  |
| --- | --- | --- |
| 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\_PROCESS | Dùng tài nguyên trong phạm vi cho phép của tiến trình |
| Detachstate | PTHREAD\_CREATE\_JOINABLE | 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 |

Minh hoạ:

pthread\_attr\_setstacksize() cài đặt kích thước stack

pthread\_attr\_getstacksize() lấy kích thước stack

Code:

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

int main()

{

size\_t stksize;

pthread\_attr\_t atr;

pthread\_attr\_getstacksize(&atr, &stksize);

printf("Kich thuoc stack cu: %ld\n", stksize);

pthread\_attr\_setstacksize(&atr, 20520605);

pthread\_attr\_getstacksize(&atr, &stksize);

printf("Kich thuoc stack moi: %ld\n", stksize);

return 0;

}

Kết quả

Text

Description automatically generated

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!”

code:

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <signal.h>

#include <sys/wait.h>

int value = 1;

void turnOnSigint()

{

printf("\nYou are press CTRL+C! Goodbye\n");

value = 0;

}

int main()

{

value = 1;

printf("Welcome to IT007, I am 20520605!\n");

pid\_t pid;

pid = fork();

if (pid == 0)

{

system("vim abcd.txt");

}

else

{

signal(SIGINT, turnOnSigint);

}

while (value)

{

}

return 0;

}



Text

Description automatically generated

Text

Description automatically generated