

Hướng dẫn Lab 3.2 – Tạo tiến trình con

Mục tiêu	Lý thuyết liên quan	Tài nguyên
Tạo tiến trình	Ch3: Process Creation	Sử dụng image Ubuntu 14 / 16
Hủy tiến trình	Ch3: Process Creation	

Cập nhật

2021.Jun.2	Thêm các code ví dụ trong sách lý thuyết.
------------	-------------------------------------------

Yêu cầu sinh viên: Hiểu lý thuyết liên quan. Biết biên dịch các đoạn code mẫu và thực thi. Biết cách sử dụng lời gọi fork() tạo ra tiến trình con. Áp dụng lập trình đa nhiệm bằng cách tạo ra các tiến trình con. Áp dụng tạo ra cây tiến trình và quá trình kết thúc đồng bộ.

Đánh giá sinh viên: Trả lời các vấn đề lý thuyết. Kỹ năng thực hành. Bài tập.

Yêu cầu nộp bài: các tập tin mã nguồn .c và tập tin khả thực thi .out của các “ví dụ” và bài tập cuối hướng dẫn.

Preferences

[1] Abraham Silberschatz, Peter B. Galvin, Greg Gagne, [2018], Operating System Concepts, 10th edition, John Wiley & Sons, New Jersey.

Programming Problems of Chapter 3.

[2] Greg Gagne , [2019], GitHub OS-BOOK OSC10e, Westminster College, United States

Access <https://github.com/greggagne/osc10e> in September 2019.

[3] Linux manual page, [2021], on fork(2)

Access <https://man7.org/linux/man-pages/man2/fork.2.html>

Yêu cầu 1: Sử dụng lời gọi `fork()` trong thư viện “`unistd.h`” để tạo ra một tiến trình con, tiến trình con thực thi lời gọi `ls`.

Hướng dẫn: Tải về tập tin *LAB_IntroOS/CHAPTER 3/Fig3_08_Fork.c*, sau đó biên dịch và thực thi rồi quan sát.

?	<ul style="list-style-type: none">- Lệnh <code>ls</code> có chức năng gì?- Lời gọi <code>fork()</code> được sử dụng để làm gì?- Tiến trình con thừa kế những gì từ tiến trình cha sinh ra nó?- Làm sao tách biệt đoạn code để tiến trình cha và con thực hiện các khối lệnh riêng biệt?	LAB 1 Ch3 p.151	Slide Ch3.
----------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------	------------

- Dòng 11 là khai báo biến `pid` nhận kết quả trả về khi gọi `fork()` ở dòng 13.
- Dòng 14 xảy ra khi `fork()` không thành công.
- Dòng 18 chỉ ra khối lệnh phía sau sẽ sao chép cho tiến trình con.
- Dòng 21 chỉ ra phần lệnh của tiến trình cha (tiến trình hiện tại).
- Dòng 20 là lời gọi thực thi một lệnh (xem thêm LAB 3.3).
- Dòng 24 là sự chờ đợi của tiến trình cha, chỉ khi nào tiến trình con hoàn tất thì tiến trình cha mới vượt lệnh này.

```
11    pid_t pid;
13    pid = fork();
14        if (pid < 0) {
18        } else if (pid == 0) {
20            execlp("/bin/ls", "ls", NULL);
21        } else {
24            wait(NULL);
```

Hàm tạo tiến trình `pid_t fork(void)` được sử dụng với lời gọi `pid_t pid=fork()`:

Nếu thành công và `pid=0`: đang trong thân process con.

Nếu thành công và `pid>0`: xử lý trong thân process cha.

Nếu thất bại: `pid=-1` kèm lý do, ví dụ như `ENOMEM`: không đủ bộ nhớ, hay là `EAGAIN`: số tiến trình vượt quá giới hạn cho phép.

Yêu cầu 2: Hoàn thiện đoạn code để một tiến trình cha sinh ra một tiến trình con, mỗi tiến trình in ra màn hình ID của chúng, đổi tên tiến trình trên hệ thống và đồng bộ kết thúc giữa cha và con.

Hướng dẫn: Tải về tập tin *LAB_IntroOS/LAB_3/task2_2.c*, sau đó biên dịch và thực thi và tiếp tục hoàn thiện.

- sleep(10) trong tiến trình con sẽ làm cho tiến trình con kết thúc sau, trong khi tiến trình cha sẽ kết thúc trước. Tiến trình con khi đó được gọi là tiến trình mồ côi. Hãy đặt lệnh wait() vào vị trí thích hợp trong tiến trình cha để cha chờ đợi và kết thúc sau đó.

- Thay đổi tên của tiến trình bằng lệnh, sau khi thay đổi hãy quan sát trong System Monitor kết quả thay đổi.

```
#include <sys/prctl.h>
const char * name = "New_named";
if (prctl(PR_SET_NAME, (unsigned long) name) < 0) perror("prctl()");
```

Vấn đề lập trình

3.1 Khi chạy đoạn mã sau, nội dung gì được in ra màn hình bởi dòng lệnh A (dòng 15)?

01	#include <sys/types.h>
02	#include <stdio.h>
03	#include <unistd.h>
04	int value = 5;
05	int main()
06	{
07	pid_t pid;
08	pid = fork();
09	if (pid == 0) { /* child process */
10	value += 15;
11	return 0;
12	}
13	else if (pid > 0) { /* parent process */
14	wait(NULL);
15	printf("PARENT: value = %d",value); /* LINE A */
16	return 0;
17	}
18	}

Gợi ý: sinh viên kết luận gì về dữ liệu của tiến trình cha và tiến trình con? Biến *value* khai báo tại dòng 5 có ý nghĩa “toàn cục” không? Nếu muốn chia sẻ dữ liệu giữa hai tiến trình cha con thì có thực hiện được không?

3.2 Có bao nhiêu tiến trình được tạo ra sau khi chạy đoạn chương trình dưới đây, bao gồm cả tiến trình cha ban đầu?

Gợi ý: khi tiến trình kết thúc, chúng sẽ được xóa khỏi “System Monitor”, để các tiến trình tồn tại lâu hơn, hãy thêm vào lệnh *sleep(x)* trước dòng 11 với tham số *x* phù hợp.

?	<ul style="list-style-type: none"> - Tiến trình con được sinh ra thừa kế những gì từ tiến trình cha? - Tiến trình con và tiến trình cha có thể cùng chia sẻ một biến số được khai báo trước đó không? 	Ch3 p.151	Slide Ch3.
---	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------	------------

01	#include <stdio.h>
02	#include <unistd.h>
03	int main()
04	{
05	/* fork a child process */
06	fork();
07	/* fork another child process */
08	fork();
09	/* and fork another */
10	fork();
11	return 0;
12	}

Bài tập lập trình

- Viết chương trình để truyền đối vào số nguyên dương *n* vào và
 - Tiến trình cha tiếp tục tính rồi xuất ra tổng $S = 1 + 2 + \dots + n$
 - Đồng thời tạo một tiến trình con tính tổng các ước số của *n* và in ra màn hình.

Gợi ý cho bài tập này tại **LAB_IntroOS/LAB_3/ex1_Pseudocode.c**

- Programming Problems 3.21 in book [1]

Phỏng đoán Collatz¹ tin rằng dãy số sinh ra sẽ luôn tiến về 1 với bất kỳ số nguyên dương nào được tạo ra ở bước đầu. Dãy số được tạo ra theo giải thuật sau:

$$n = \begin{cases} \frac{n}{2} & \text{nếu } n \text{ là số chẵn} \\ 3 * n + 1 & \text{nếu } n \text{ là số lẻ} \end{cases}$$

Ví dụ với $n=35$, dãy số sinh ra là 35, 106, 53, 160, 80, 40, 20, 10, 5, 16, 8, 4, 2, 1

Viết một chương trình nhận số nguyên dương n vào thông qua đối số, kiểm tra tính đúng của giá trị này. Tạo ra một tiến trình con để tính và in ra dãy số, trong lúc tiến trình cha cần chờ tiến trình con hoàn thành thông qua lời gọi `wait()`.

```
>./collazt 35
35, 106, 53, 160, 80, 40, 20, 10, 5, 16, 8, 4, 2, 1
Ket thuc tien trinh con
```

Gợi ý cho bài tập này tại **LAB_IntroOS/LAB_3/ex2_Pseudocode.c**

CHECKLIST LAB

- ☐ Khởi động Ubuntu.
- ☐ Chạy Terminal.
- ☐ Biên tập một tập tin văn bản .txt, .c
- ☐ Biên dịch và Liên kết một mã nguồn .c
- ☐ Tạo thư viện liên kết tĩnh.
- ☐ Tạo thư viện liên kết động.
- ☐ Chạy một chương trình đã biên dịch.
- ☐ Truyền đối số qua dòng lệnh.
- ☐ Tạo tiến trình con.
- ☐ Cấu trúc mã nguồn cho tiến trình cha và tiến trình con.
- ☐ Tạo cây tiến trình
- ☐ Đồng bộ kết thúc tiến trình cha/con.
- ☐ Gọi chạy một lệnh từ trong chương trình.

¹ https://vi.wikipedia.org/wiki/Ph%E1%BB%8Fng_%C4%91o%C3%A1n_Collatz