

LAB TEST - 01

52100674@student.tdtu.edu.vn [Switch account](#)



LAB TEST - 03

Dòng số 15 sẽ in ra màn hình nội dung gì, và tại sao?

1 point

```
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 }
```

- ☐ PARENT: value = 20 bởi vì tiến trình con đã thêm 15 vào giá trị 5 ban đầu của biến value
- ☐ PARENT: value = 5 bởi vì biến số value của tiến trình cha không bị thay đổi
- ☐ PARENT: value = 5 bởi vì có thể lệnh cộng ở dòng 10 xảy ra sau lệnh in ở dòng 15
- ☐ PARENT: value = 0 vì sau khi sinh tiến trình con, biến số value được khởi tạo.



Để bắt buộc tiến trình có pid=1234 dừng ngay lập tức, lệnh nào cần được sử dụng? 1 point

- ☐ kill -2 1234
- ☐ kill -a 1234
- ☐ kill force 1234
- ☐ kill -9 1234

Tiến trình con tiếp nhận những gì từ tiến trình cha? Chọn mọi đáp án đúng. 2 points

- ☐ Sao chép toàn bộ mã nguồn kể từ lời gọi fork()
- ☐ Tạo mới vùng nhớ và sao chép toàn bộ biến số
- ☐ Có bộ đếm chương trình riêng và chạy độc lập
- ☐ Chia sẻ và dùng chung các biến số toàn cục
- ☐ Chiếm hữu các biến số khai báo sau lời gọi fork()

Lệnh "getppid();" trả về thông tin gì khi thực thi nó trong một terminal? 1 point

- ☐ id của bash đã gọi chạy
- ☐ id của tiến trình chứa câu lệnh
- ☐ id của tiến trình quản lý các tiến trình
- ☐ id của lệnh ppid()



Nếu tiến trình cha kết thúc trước tiến trình con, đó là tình huống gì và diễn tiến tiếp theo là gì? 1 point

- ☐ Tiến trình con trở thành mồ côi và tiếp tục chạy.
- ☐ Tiến trình con trở thành mồ côi và bị buộc kết thúc ngay sau đó.
- ☐ Tiến trình con trở thành xác sống và gây lỗi cho hệ thống
- ☐ Tiến trình cha không bao giờ kết thúc trước khi con nó hoàn tất.

Lệnh wait() ở dòng 14 diễn tiến ra sao? 1 point

```
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 }
```

- ☐ Lệnh trì hoãn cho đến khi hàm main() kết thúc
- ☐ Lệnh trì hoãn cho đến khi lệnh cuối cùng của tiến trình con kết thúc
- ☐ Lệnh chờ với tham số thời gian được truyền vào
- ☐ Lệnh chờ cho đến khi tiến trình con được sinh ra



Nếu tiến trình con kết thúc trước tiến trình cha mà cha không biết về việc kết thúc này, đó là tình huống gì và diễn tiến tiếp theo là gì? 1 point

- ☐ Tiến trình con trở thành mồ côi và tiếp tục chạy.
- ☐ Tiến trình con trở thành xác sống và gây lỗi cho hệ thống.
- ☐ Tiến trình con trở thành xác sống và được HĐH quản lý.
- ☐ Tiến trình cha không bao giờ kết thúc trước khi con nó hoàn tất.

Lệnh nào in các tiến trình dưới dạng cây để xem cho thuận tiện? 1 point

- ☐ ps --forest
- ☐ ps --tree
- ☐ ps -a
- ☐ ps --list

Lời gọi fork() được sử dụng trong trường hợp nào? 1 point

- ☐ Tạo ra thêm một tiến trình
- ☐ Tạo ra thêm hai tiểu trình
- ☐ Thực thi một tiến trình mới
- ☐ Kết thúc tiến trình hiện tại

Page 3 of 5

Back

Next

Clear form

Never submit passwords through Google Forms.

This form was created inside of Ton Duc Thang University. [Report Abuse](#)



Google Forms

