ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

Tài liệu hướng dẫn thực hành

HỆ ĐIỀU HÀNH

Biên soạn: ThS Phan Đình Duy

ThS Nguyễn Thanh Thiện

KS Trần Đại Dương

ThS Trần Hoàng Lộc

KS Thân Thế Tùng

MỤC LỤC

NỘI QUY THỰC HÀNH

1. Sinh viên tham dự đầy đủ các buổi thực hành theo quy định của giảng viên hướng dẫn (GVHD) (6 buổi với lớp thực hành cách tuần hoặc 10 buổi với lớp thực hành liên tục).
2. Sinh viên phải chuẩn bị các nội dung trong phần "Sinh viên viên chuẩn bị" trước khi đến lớp. GVHD sẽ kiểm tra bài chuẩn bị của sinh viên trong 15 phút đầu của buổi học (nếu không có bài chuẩn bị thì sinh viên bị tính vắng buổi thực hành đó).
3. Sinh viên làm các bài tập ôn tập để được cộng điểm thực hành, bài tập ôn tập sẽ được GVHD kiểm tra khi sinh viên có yêu cầu trong buổi học liền sau bài thực hành đó. Điểm cộng tối đa không quá 2 điểm cho mỗi bài thực hành.

Bài 6. BÀI THỰC HÀNH TỔNG HỢP

6.1 Mục tiêu

- Sử dụng các kiến thức ở các bài thực hành trước để hoàn thành các yêu cầu thực hành có tính chất tổng hợp, vận dụng.

6.2 Nội dung thực hành

- Viết chương trình C tạo ra một giao diện shell có các chức năng như:
  - Thực thi command trong tiến trình con.
  - Tạo tính năng sử dụng lại các câu lệnh trong quá khứ.
  - Chuyển hướng vào ra.
  - Giao tiếp sử dụng cơ chế đường ống.
  - Kết thúc lệnh đang thực thi.

6.3 Sinh viên chuẩn bị

Để thực hiện bài thực hành này, sinh viên phải đảm bảo những điều sau:

- Đã cài đặt C compiler cho hệ điều hành Linux.
- Biết cách viết, build và chạy một chương trình trên hệ điều hành Linux.

- Biết cách viết chương trình sử dụng cơ chế duplicate và pipeline.

6.3.1 Câu hỏi chuẩn bị

Sinh viên chuẩn bị câu trả lời cho câu hỏi sau trước khi bắt đầu phần thực hành:

- Hãy nêu các định nghĩa, chức năng và ví dụ sử dụng của các hàm sau đây: exec(), dup2(), pipe().

6.4 Sinh viên thực hành

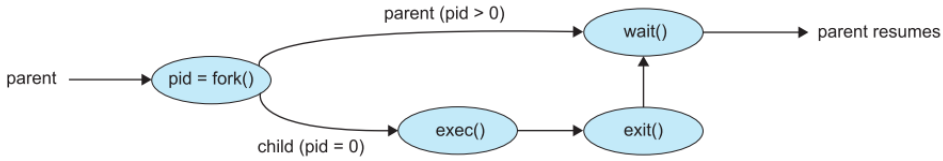
Hãy thiết kế một chương trình C để tạo ra một giao diện shell. Giao diện này cho phép người dùng nhập các lệnh và sau đó thực thi từng lệnh trong một quy trình riêng biệt. Điểm đặc biệt là chương trình này sẽ hỗ trợ việc chuyển hướng đầu vào và đầu ra, cũng như sử dụng pipes như một cách để truyền thông tin giữa các cặp lệnh.

Giao diện shell cung cấp cho người dùng lời nhắc, sau đó lệnh được nhập. Mỗi khi lệnh thực hiện xong, shell sẽ hiện lên dấu nhắc để chạy lệnh khác. Ví dụ dưới đây minh họa dấu nhắc it007sh> và lệnh của người dùng: echo abc

```
it007sh> echo abc
```

Một kỹ thuật để triển khai giao diện shell là trước tiên tiến trình cha đọc những gì người dùng nhập vào dòng lệnh (trong trường hợp này, là lệnh "echo abc "), sau đó tạo ra một tiến trình con riêng biệt để thực hiện lệnh đó. Trừ khi được chỉ định khác, tiến trình cha đợi

cho đến khi tiến trình con thoát ra trước khi tiếp tục. Điều này tương tự về chức năng với việc tạo ra một tiến trình mới như được minh họa dưới đây:



Hình 1. Tạo process sử dụng fork()

Đoạn chương trình mẫu tạo ra dấu nhắc it007sh>:

```
#include <stdio.h>
#include <unistd.h>
#define MAX LINE 80 /* The maximum length command */


int main(void) {
char *args[MAX LINE/2 + 1]; /* command line arguments */
int should run = 1; /* flag to determine when to exit
program */
    while (should run) {
        printf("it007sh>");
        fflush(stdout);
        /**
        Do something
        */
    }
return 0;
}
```

Dựa vào đoạn chương trình trên hãy thực hiện thêm các yêu cầu dưới đây:

1. Thực thi lệnh trong tiến trình con

- Ví dụ: khi thực hiện

```
it007sh> echo abc
```

Kết quả sẽ in ra chuỗi abc, kết thúc dòng lệnh sẽ hiển thị dấu nhắc it007sh> để người dùng nhập lệnh tiếp theo. Lưu ý rằng trong khi lệnh `echo abc` đang thực thi, không cho người dùng nhập command mới.

- Gợi ý: Xem hình 1.

2. Tạo tính năng sử dụng lại câu lệnh gần đây

- Cho phép người dùng thực thi lệnh gần đây bằng cách sử dụng các phím lên/xuống (để chọn lệnh) và nhấn Enter.

Ví dụ: Nếu các lệnh đã nhập vào shell (theo thứ tự)

```
echo abc
ls -l
pwd
```

thì khi sử dụng các phím lên (↑) /xuống (↓), shell sẽ lần lượt hiển thị lại các lệnh trên để người dùng lựa chọn. Nhấn phím Enter để thực thi lệnh đang hiển thị.

3. Chuyển hướng vào ra

- Hỗ trợ các toán tử chuyển hướng '>' và '<', trong đó '>' chuyển hướng đầu ra của lệnh sang một tệp và '<' chuyển hướng đầu vào của lệnh từ một tệp. Ví dụ: nếu người dùng nhập

```
it007sh>ls > out.txt
```

thì đầu ra từ lệnh ls sẽ được chuyển hướng đến tệp out.txt. Tương tự, đầu vào cũng có thể được chuyển hướng. Ví dụ, nếu người dùng nhập

```
it007sh>sort < in.txt
```

thì tệp in.txt sẽ đóng vai trò là đầu vào cho lệnh sắp xếp.

- Gợi ý: sử dụng hàm dup2()

4. Giao tiếp sử dụng cơ chế đường ống

- Cho phép đầu ra của một lệnh đóng vai trò là đầu vào cho lệnh khác bằng cách sử dụng một đường ống. Ví dụ: Khi người dùng nhập

```
it007sh>ls -l | less
```

thì đầu ra của lệnh `ls -l` đóng vai trò là đầu vào cho lệnh `less`.

- Gợi ý: sử dụng hàm pipe() và dup2().

5. Kết thúc lệnh đang thực thi bằng tổ hợp phím Ctrl + C

- Ví dụ: Thực hiện lệnh

```
it007sh>top
```

sẽ liên tục hiển thị các tiến trình của hệ thống. Khi đó, nếu sử dụng tổ hợp phím Ctrl + C, lệnh thực thi trên sẽ kết thúc và hiển thị dấu nhắc it007sh> mời người dùng nhập lệnh tiếp theo.