

Hướng dẫn Lab 5.2 – Đường ống được đặt tên

Đường ống bình thường (Ordinary pipe) trong Linux và Đường ống không tên (Unnamed pipe) trong Windows là cơ chế liên lạc giữa các tiến trình cha con. Trong các vấn đề có tầm vực rộng hơn, ví dụ liên lạc giữa các tiến trình không có mối quan hệ cha con, đường ống được đặt tên (Named pipe) có thể được sử dụng.

Mục tiêu	Lý thuyết liên quan	Tài nguyên
Liên lạc trong shell	Ch3: Process Ch3.3.7.4 Pipes	https://github.com/Trantin84/LAB_IntroOS (mã nguồn ví dụ). Sử dụng image Ubuntu 16 / 18 Sử dụng Windows 10
Đường ống được đặt tên		
Chương trình chat đơn giản		

Yêu cầu sinh viên: Hiểu lý thuyết về liên lạc giữa các tiến trình bằng đường ống có tên. Hiểu và thực thi các đoạn mã đã cung cấp. So sánh đường ống bình thường/không tên với đường ống có tên. Phân tích và áp dụng cho các bài toán liên lạc giữa các tiến trình.

Đánh giá sinh viên: Hỏi đáp 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 Yêu cầu trong buổi thực hành và Bài tập cuối hướng dẫn trong thời gian cho phép của giảng viên.

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] StackOverflow, [2021], Example of using named pipes in Linux Bash,

Access <https://stackoverflow.com/questions/4113986/example-of-using-named-pipes-in-linux-bash>

[3] Geeksforgeeks, [2021], Named Pipe or FIFO with example C program,

Access <https://www.geeksforgeeks.org/named-pipe-fifo-example-c-program/>

Liên lạc thông qua CLI.

Yêu cầu 1: Tạo một đường ống có tên và liên lạc giữa 2 tiến trình qua đường ống đó.

Hướng dẫn: Mở 2 Terminal riêng biệt.

Gõ lệnh trên Terminal 1

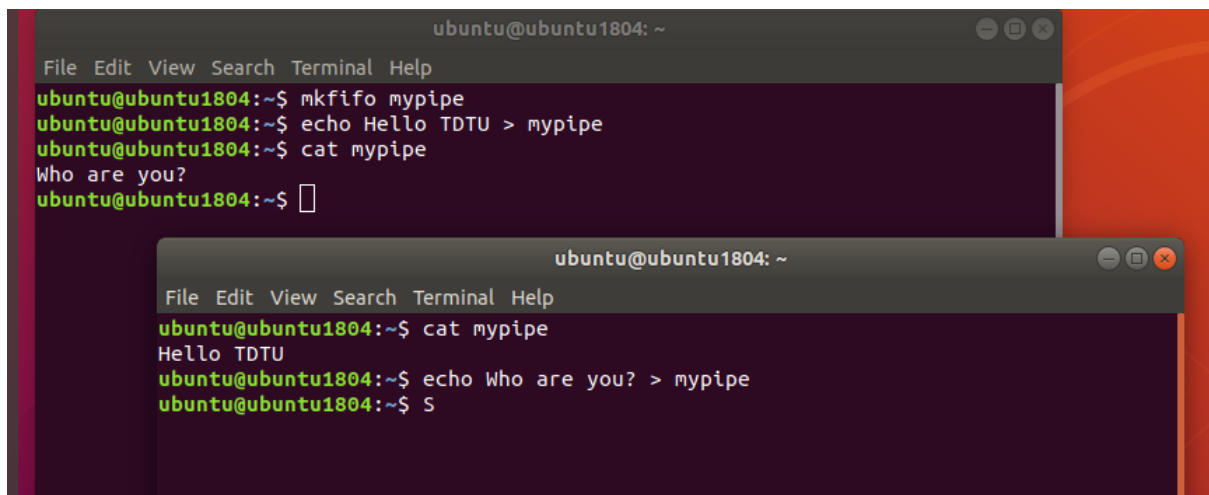
```
>mkfifo mypipe  
>echo Hello TDTU > mypipe
```

Chuyển sang Terminal 2, gõ lệnh

```
>cat mypipe
```

Dấu nhắc sẽ in ra “Hello TDTU”. Có thể tiếp tục liên lạc với cách thức gửi vào là “String > mypipe” và đọc ra là “cat mypipe”. Sau khi kết thúc liên lạc, huỷ đường ống bằng lệnh:

```
>unlink mypipe
```



Hình 1. Liên lạc giữa 2 tiến trình bash độc lập.

Đường ống được đặt tên (Ordinary pipe)

Yêu cầu 2: Viết chương trình chat giữa 2 tiến trình độc lập.

Hướng dẫn: Tải về tập tin *LAB_IntroOS/LAB_5/task2_2parent.c* và */task2_2child.c*, biên dịch và thực thi trên 2 terminal khác nhau, tiến trình parent thực thi trước.

a. Giải thích mã nguồn PARENT

- Dòng 22, 23 và 24 định nghĩa 2 đường ống và quyền truy cập toàn quyền (0666)
- Dòng 26 khai báo đường ống kích thước tối đa 4096 bytes.
- Dòng 29 khai báo các chuỗi s1, s2, buf với kích thước bằng kích thước đường ống.
- Dòng 30 chứa con trỏ quản lý đầu đọc và đầu ghi của đường ống tạo ra.
- Dòng 31 là lời gọi tạo ra đường ống có đường dẫn FIFO1, quyền đọc ghi PM.
- Dòng 36 là mở đầu ghi (phía 1) của đường ống FIFO1.
- Dòng 38 là mở đầu đọc (phía 0) của đường ống FIFO2.
- Dòng 44 là ghi chuỗi s1 và đường ống tạo ra ở dòng 36.
- Dòng 45 là đọc hết (PIPE_BUF là 4096 bytes) đường ống và ghi vào s2.
- Dòng 50 và 51 là đóng đầu đọc và đầu ghi của đường ống.
- Dòng 52 và 54 là huỷ bỏ đường ống (unlink), sau khi đã đóng cả 2 đầu đọc ghi của mỗi đường ống.

22	#define FIFO1 "/tmp/ff.1"
23	#define FIFO2 "/tmp/ff.2"
24	#define PM 0666
26	#define PIPE_BUF 4096
29	char s1[PIPE_BUF], s2[PIPE_BUF], buf[PIPE_BUF];
30	int readfd, writefd;
31	if ((mknod(FIFO1, S_IFIFO PM, 0) < 0) && (errno != EEXIST))
36	if ((writefd = open(FIFO1, 1)) < 0)
38	if ((readfd = open(FIFO2, 0)) < 0)
44	write(writefd, s1, strlen(s1));
45	read(readfd, s2, PIPE_BUF);
50	close(readfd);
51	close(writefd);
52	if (unlink(FIFO1) < 0)
54	if (unlink(FIFO2) < 0)

b. Giải thích mã nguồn CHILD

- Dòng 31 là tạo ra đường ống FIFO2 để gửi dữ liệu chiều ngược lại.
- Các dòng còn lại thực hiện công việc tương tự tiến trình Cha, chú ý thao tác diễn ra trên đường ống nào.

31	if ((mknod(FIFO2, S_IFIFO PM, 0) < 0) && (errno != EEXIST)) {
----	---

36	if ((readfd = open(FIFO1, 0)) < 0)
38	if ((writefd = open(FIFO2, 1)) < 0)
41	read(readfd, s2, PIPE_BUF);
46	write(writefd, s1, strlen(s1));
50	close(readfd);
51	close(writefd);

```

File Edit View Search Terminal Help
ubuntu@ubuntu1804:~/tttin/LAB5$ ./parent.out
(me): Hello student.
Child said: Hi teacher, how are you today?
(me): Thanks. I'm fine. Have you finished your homework?
Child said: Not yet. I feel sick. Might I've got COVID-19.
(me): Oh no no. Stay at home and call 18001119.
Child said: Ok teacher.
eel sick. Might I've got COVID-19.
(me): byebye
ubuntu@ubuntu1804:~/tttin/LAB5$

ubuntu@ubuntu1804: ~/tttin/LAB5
File Edit View Search Terminal Help
ubuntu@ubuntu1804:~/tttin/LAB5$ ./child.out
Parent said: Hello student.
(me): Hi teacher, how are you today?
Parent said: Thanks. I'm fine. Have you finished your homework?
(me): Not yet. I feel sick. Might I've got COVID-19.
Parent said: Oh no no. Stay at home and call 18001119.
omework?
(me): Ok teacher.
byebye
Parent said: Oh no no. Stay at home and call 18001119.
omework?
(me): ubuntu@ubuntu1804:~/tttin/LAB5$

```

Hình 2. Chương trình chat bằng Đường ống được đặt tên.

Bài tập lập trình.

1. Tiến trình cha chuyển đổi số đầu tiên (argv [1]) là một số nguyên lớn hơn 3 cho tiến trình con thông qua đường ống được đặt tên. Tiến trình con nhận, tính giá trị $n! = 1 * 2 * \dots * n$ và ghi nó vào đường ống. Tiến trình cha nhận và xuất dữ liệu ra màn hình. Sử dụng đường ống bình thường (Named Pipe).

```

> ./baitap2A.out 4
4! = 24

```