**Bài tập Lập trình mạng tuần 5**

**I/O Multiplexing**

Họ và tên: Nguyễn Tuấn Thành

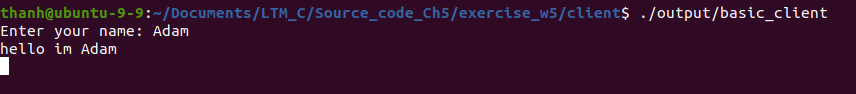
MSSV: 20210800

**Phần 1&2: Lập Trình Socket TCP - Thiết Lập Cơ Bản, Logic Server và Phát Tin Nhắn**

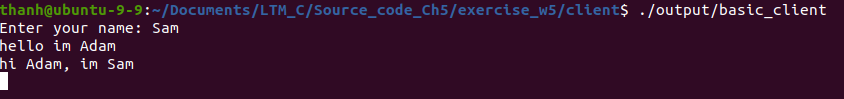
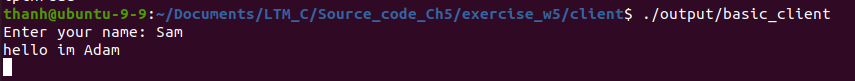
Các file mã nguồn đã được đính kèm trong assigment.

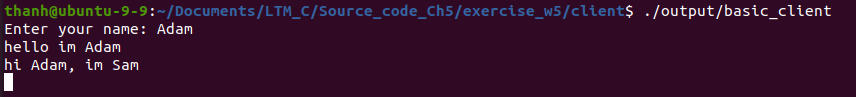
Kết quả chạy chương trình:

+ Với người dùng Adam, nhập 1 message



+ Khi đó, người dùng Sam đã nhận được message từ Adam, tiếp theo Sam nhập thêm 1 message



+ Adam đã nhận được message

**Kết quả:**

**- Server với kết nối TCP và socket cơ bản đã có thể xử lý nhiều client và có thể hiển thị tin nhắn từ 1 client đến các client khác trong thời gian thực**

**Phần 3: I/O Multiplexing để Xử Lý Nhiều Client**

Thực hiện chạy server và khởi tạo 3 người dùng Adam, Eva, Sam. Cho 3 người dùng trao đổi tin nhắn và đóng kết nối của 1 người dùng, kết quả chạy chương trình thu được:

**3.1: Sử Dụng select() để Xử Lý Nhiều Client:**

A computer screen shot of a computer

Description automatically generatedA computer screen shot of a computer

Description automatically generated

**3.2: Sử Dụng pselect() để Thực Hiện I/O Multiplexing An Toàn Với Tín hiệu**

A computer screen shot of a computer screen

Description automatically generated

**3.3: Sử Dụng poll() để Xử Lý Nhiều Client**

A computer screen shot of a computer code

Description automatically generated

Kết quả thu được:

* Server mở connect và thu nhận các lượt connnect đến từ người dùng cũng như các message của họ
* Sau khi một người dùng ngắt kết nối khỏi server, server sẽ ngắt kết nối 1 cách an toàn và thông báo tới các người dùng khác

**3.4: Giải thích cơ chế hoạt động của các hàm xử lý khác nhau**

**3.4.1: select()**

Hoạt động của select server có thể được mô tả như sau:

1. Sử dụng select(): Server sử dụng select() để theo dõi nhiều socket client và socket server.

A blue background with red and blue text

Description automatically generated

2. Xử lý kết nối mới: Nếu có kết nối mới từ client, server sẽ chấp nhận kết nối và thêm socket mới vào danh sách clients.

A computer screen shot of a program code

Description automatically generated

3. Xử lý tin nhắn từ client: Nếu có tin nhắn từ client, server sẽ đọc tin nhắn và phát đến tất cả các client còn lại.

4. Xử lý ngắt kết nối: Nếu một client ngắt kết nối, server sẽ xóa socket của client đó khỏi danh sách clients.

A computer screen shot of text

Description automatically generated

**3.4.2: pselect()**

1. Xử lý tín hiệu: Sử dụng hàm signal() để thiết lập xử lý tín hiệu SIGINT. Khi nhận được tín hiệu này (như khi người dùng nhấn Ctrl+C), server sẽ thực hiện hàm handle\_signal().

2. Hàm handle\_signal(): Hàm này sẽ đóng tất cả các kết nối client trước khi thoát chương trình, đảm bảo rằng mọi client đều được ngắt kết nối một cách đúng đắn.



A computer screen shot of text

Description automatically generated

3. Sử dụng pselect(): Thay vì select(), mã sử dụng pselect() để cho phép xử lý tín hiệu và giám sát socket đồng thời. Tuy nhiên, trong trường hợp này, ta không cần truyền một biến tín hiệu vào pselect(), mà chỉ cần xử lý các tín hiệu trong hàm xử lý.

A blue background with text

Description automatically generated

**3.4.3: poll server**

1. Sử dụng poll(): Server sử dụng poll() để giám sát socket server và các socket client, cho phép xử lý đồng thời nhiều kết nối.

A computer screen shot of a error

Description automatically generated

2. Cập nhật tập hợp socket: Mỗi lần lặp, mã sẽ cập nhật tập hợp pollfd để thêm socket server và các socket client hiện có.

A blue background with red text

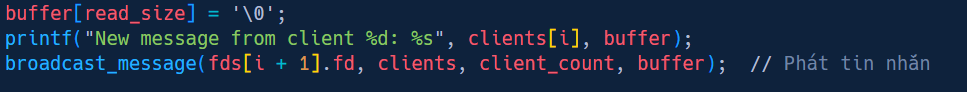
Description automatically generated

3. Xử lý kết nối mới: Nếu có kết nối mới từ client, server sẽ sử dụng accept() để nhận kết nối và thêm socket client vào danh sách.

A screen shot of a computer code

Description automatically generated

4. Xử lý tin nhắn từ client: Khi một client gửi tin nhắn, server đọc tin nhắn từ socket client và phát tới tất cả các client khác, sử dụng hàm broadcast\_message()



Hàm broadcast\_message():

A screen shot of a computer code

Description automatically generated