**BÁO CÁO THỰC HÀNH TUẦN 5**

**MÔN: THỰC HÀNH LẬP TRÌNH MẠNG**

**Sinh viên: Đinh Huy Dương**

**MSSV: 20215020**

**Phần 1+2:** Triển khai Server, Client và Logic phát tin

* Mã nguồn được đặt ở file chatroom\_server.c và chatroom\_client.c
* Server sử dụng hàm select() cho cơ chế I/O Multiplexing

A screenshot of a computer program

Description automatically generated

**Phần 3:** Các hàm khác nhau sử dụng cho cơ chế I/O Multiplexing

* Mã nguồn cho chương trình sử dụng hàm pselect() được đặt ở file chat\_room\_server\_pselect.c
* Mã nguồn cho chương trình sử dụng hàm poll() được đặt ở file chat\_room\_server\_poll.c
* Trong chương trình sử dụng pselect ta thử nghiệm đóng server an toàn bằng tín hiệu SIGINT. Hoàn toàn tương tự khi đóng client bằng SIGINTA computer screen shot of a program

  Description automatically generated

**GIẢI THÍCH CƠ CHẾ**

1. **select()**

* select() giám sát một tập hợp các file descriptor (FD) để xem có bất kỳ FD nào sẵn sàng cho các hoạt động I/O (đọc, ghi, hoặc lỗi).
* Tạo và khởi tạo một tập hợp các FD (fd\_set).
* Thêm các FD cần giám sát vào tập hợp này.
* Gọi select() với tập hợp các FD.
* select() sẽ block cho đến khi một hoặc nhiều FD sẵn sàng hoặc thời gian chờ hết hạn.
* Kiểm tra tập hợp các FD để xác định FD nào sẵn sàng.

1. **pselect():**

* Giống select() nhưng xác định 1 tập các signal được chặn trong lúc đợi
* Tạo và khởi tạo một tập hợp các FD (fd\_set).
* Thêm các FD cần giám sát vào tập hợp này.
* Tạo và khởi tạo một tập hợp các tín hiệu (sigset\_t).
* Gọi pselect() với tập hợp các FD, tập hợp các tín hiệu.
* pselect() sẽ chặn cho đến khi một hoặc nhiều FD sẵn sàng hoặc thời gian chờ hết hạn.
* Kiểm tra tập hợp các FD để xác định FD nào sẵn sàng.

1. **poll()**

* Giám sát một mảng các struct pollfd (trong chương trình là fds) để xem có bất kỳ FD nào sẵn sàng cho các hoạt động I/O.
* Tạo và khởi tạo một mảng fds.
* Thêm các FD cần giám sát vào mảng này và chỉ định các sự kiện cần giám sát (POLLIN, POLLOUT, v.v.).
* Gọi poll() với fds.
* poll() sẽ chặn cho đến khi một hoặc nhiều FD sẵn sàng hoặc thời gian chờ hết hạn.
* Kiểm tra fds để xác định FD nào sẵn sàng.

**Phần 4:** Tính năng trải nghiệm người dùng

* Ta sẽ implement những tính năng này ở chatroom\_server.c. Sử dụng hàm select() cho I/O Multiplexing
* Các tính năng:
  + Muốn cho người dùng được đặt tên cho bản thân để người khác nhìn thấy, ta có thể tạo 1 struct Client để chứa thông tin của cả Socket lẫn username.

A black background with colorful text

Description automatically generated

* + - Client sẽ gửi tên cho Server:

A computer code with text on it

Description automatically generated

* + - Lúc này Server sẽ nhận giá trị đầu từ Client là tên của người dùng nhập. Lưu ý phải memset() buffer username này, nếu không người dùng khác có tên dài hơn sẽ ghi đè phần cuối của client có username độ dài ngắn hơn.

A computer code with text

Description automatically generated with medium confidence

* + - Server quảng bá tin nhắn của một người cho cả chatroom, hiện cả tên A computer screen with text

      Description automatically generated

A screen shot of a computer program

Description automatically generated

* + Server có tính năng thông báo cho các client khác về 1 client rời, vào chatroom. Ta sử dụng mảng clients chứa struct Client để lấy username của Client. Tiếp đó sử dụng hàm broadcast\_message() để quảng bá thông điệp cho mọi Client, chú ý tham số của sender\_fd được đặt là -1 để báo cho mọi người.

A computer screen shot of text

Description automatically generated

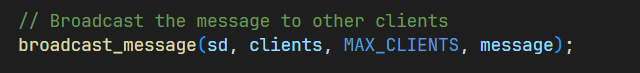
A screen shot of a computer code

Description automatically generated

A screen shot of a computer program

Description automatically generated

* + Cơ chế phát tin cho cả phòng: Sử dụng broadcast\_message, nhưng tham số đầu là sd (sender descriptor) để tránh gửi lại cho người gửi



KHÓ KHĂN:

* Làm như nào để lưu được cả tên người dùng và socket của Client?
* Tạo 1 struct Client
* Ban đầu có lỗi về đuôi tên dài hơn dính vào tên người ngắn hơn
* Clear buffer nhận tên bên Server
* Một vài vấn đề về sprintf có thể không in được vào message so buffer quá nhỏ và thông tin có thể bị truncated (cắt)
* Tăng độ dài buffer
* Có thể không phải khó khăn, nhưng do một vài lý do bảo mật C không cho dùng strcpy để copy xâu, mà thay đó sử dụng strncpy, tương tự với snprintf được dùng thay cho sprintf (Khả năng cao là do lỗi Buffer Overflow bị lợi dụng để hack)A screen shot of a computer

  Description automatically generated