**BÁO CÁO PROJECT**

Ứng viên: Phạm Lê Quốc Khánh – <https://github.com/khanh892002/simple-data-transfering-program>

1. **Cấu trúc project:**
   1. **ReceiveData:**

Bên server có các bước xử lý chính gồm:

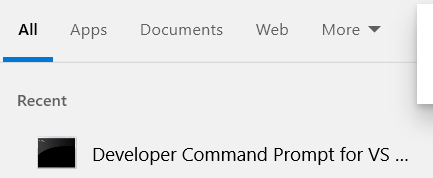
* Xử lý kiểm tra đầu vào hợp lệ khi vừa khởi chạy file. Kiểm tra thư mục lưu trữ có tồn tại hay không.
* Tạo đối tượng lưu thông tin địa chỉ máy và gắn vào một socket là socket chính của server.
* Lắng nghe (listen) và nhận (accept) các kết nối đến.
* Client khi muốn thực hiện một yêu cầu nào sẽ tự động gửi đến trước một chuỗi 4B là FILE hoặc TEXT.
* Chương trình chỉ kiểm tra xem chuỗi được gửi đến có phải là FILE hay không để tiếp tục xử lý. Nếu là chuỗi FILE thì sẽ đợi nhận tên file từ client gửi qua, nếu gặp lỗi trong quá trình tạo file thì chương trình sẽ gửi lại cho client chuỗi “FAIL” để báo lưu file thất bại. Ngược lại, sẽ gửi chuỗi “SCSS” (viết tắt của SUCCESS) và bắt đầu nhận nội dung file từ client.
* Ngược lại nếu chuỗi nhận được trước khi xử lý là TEXT, không phải FILE, thì chương trình sẽ nhận nội dung tin nhắn mà client gửi qua, tin nhắn sẽ chỉ được giới hạn 511 ký tự, nếu dư sẽ không gửi hết.
  1. **SendData:**

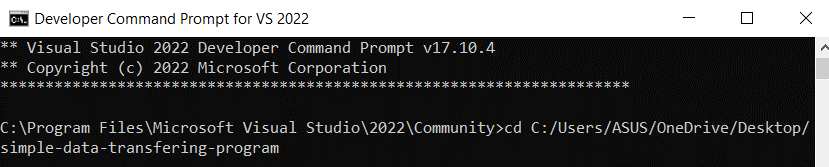
Các bước xử lý chính trên client:

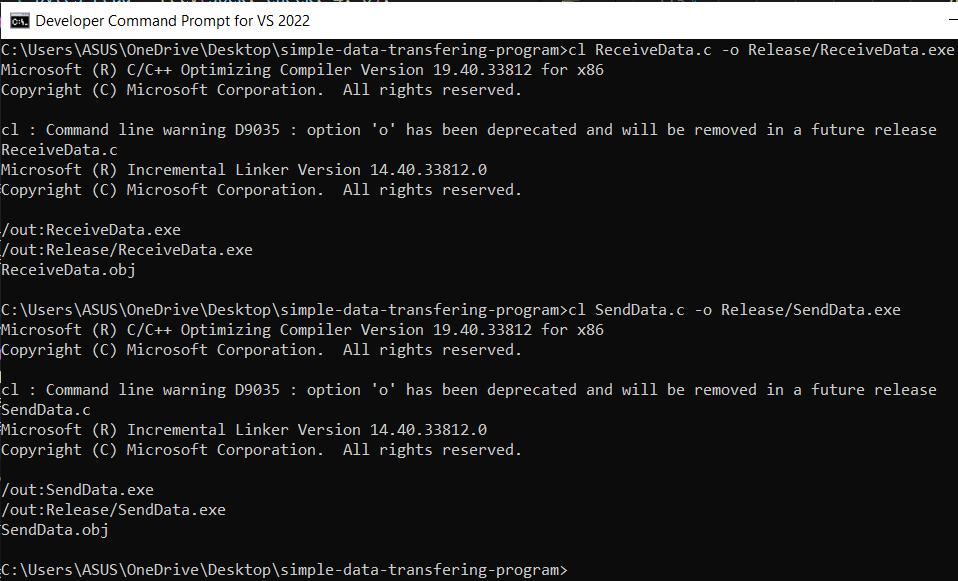
* Xử lý kiểm tra đầu vào hợp lệ khi vừa khởi chạy file.
* Nhận tham số địa chỉ IP của máy server từ đầu vào, tạo một đối tượng lưu thông tin địa chỉ IP, nếu gặp lỗi trong quá trình lưu trữ thì báo lỗi và thoát chương trình. Ngược lại, tạo một socket kết nối đến server lấy thông tin địa chỉ từ đối tượng chứa địa chỉ. Nếu kết nối thất bại thì báo lỗi và thoát chương trình.
* Sau đó, yêu cầu người dùng nhập đúng cú pháp các lệnh, nếu nhập sai sẽ báo lỗi.
* Tạo một mảng ký tự lưu thông tin các lệnh, sẽ chỉ có 3 lệnh chính:
  + SendText: gửi chuỗi TEXT cho server rồi sau đó mới gửi nội dung tin nhắn đến server.
  + SendFile: kiểm tra thông tin đường dẫn file và buffer\_size, kiểm tra cấp phát vùng nhớ buffer để gửi nội dung file, kiểm tra mở file. Nếu thỏa hết điều kiện trên thì gửi chuỗi FILE đến server, xác định và gửi tên file cho server. Đợi nhận chuỗi xác nhận server có thể lưu trữ được file hay không (SCSS/FAIL), nếu được thì gửi nội dung file và thông báo sau khi hoàn thành, ngược lại báo lỗi từ server.
  + exit: thoát vòng lặp, kết thúc chương trình.
  + Ngoài ra: báo lỗi không hợp lệ và yêu cầu người dùng nhập lại đúng cú pháp.

1. **Cách khởi chạy:**

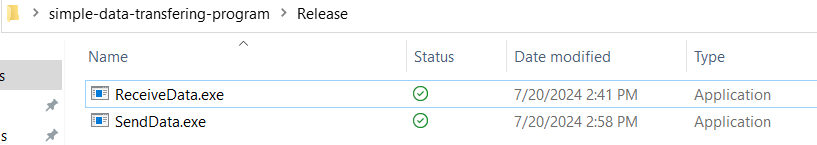
Ta cần biên dịch các file SendData.c và ReceiveData.c để sử dụng trong terminal dưới dạng các lệnh. Để biên dịch, ta sử dụng Developer Command Prompt và chỉ định các file và vị trí lưu kết quả biên dịch:



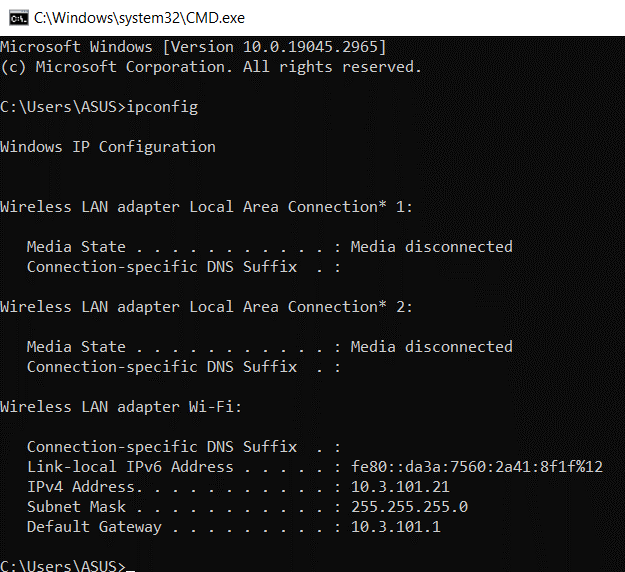


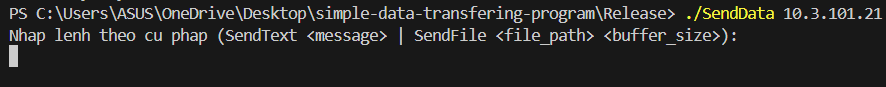


khi này sẽ có 2 file exe được tạo ra trong thư mục Release:

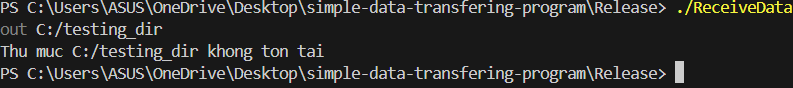


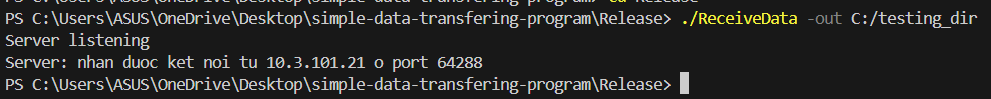
SendData cần chỉ định rõ địa chỉ IP của thiết bị đang chạy file ReceiveData, nếu là cùng một máy thì có thể sử dụng địa chỉ 127.0.0.1:



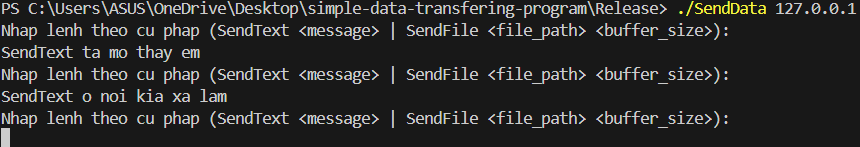


Đối với bên server nếu thư mục được truyền vào không tồn tại thì chương trình sẽ báo lỗi và thoát chương trình:

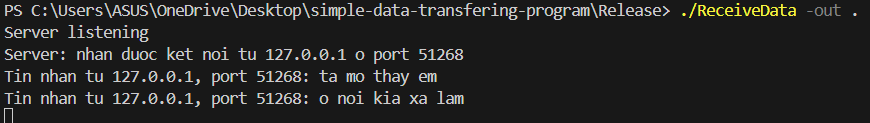


****

1. **Kết quả:**
   1. **Chức năng SendText:**
      1. Client:

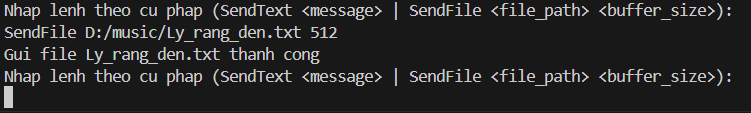
****

* + 1. Server:

****

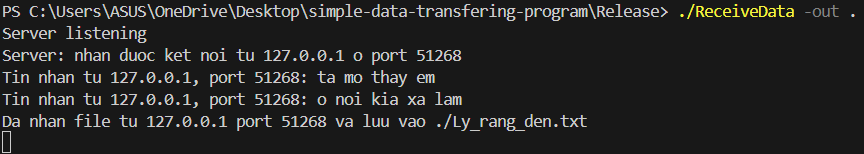
* 1. **Chức năng gửi file SendFile:**
     1. Client:

Sau khi gửi các thông tin của file đến server thì chương trình sẽ nhận phản hồi từ server cho biết file đã được gửi qua hay chưa:

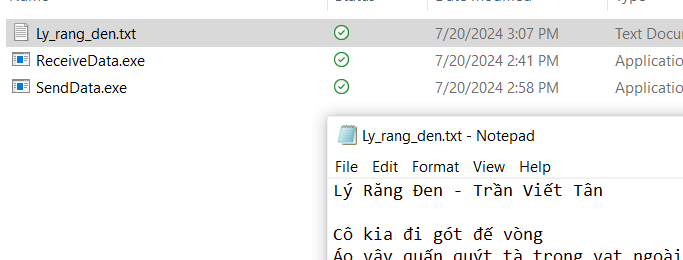
****

* + 1. Server:

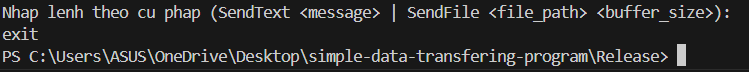
Khi đã lưu thành công file thì chương trình sẽ có thông báo trên server:

****

file sẽ được lưu vào thư mục được gán khi khởi chạy file ReceiveData.exe. (. Là thư mục cha chứa file ReceiveData.exe):

****

* 1. **Thoát (ở SendData):**

****

1. **Những lỗi chưa khắc phục được:**
   1. Lỗ hổng tiềm năng:

Nếu một thiết bị kết nối đến server khi đang chạy, thì có thể gửi đến một chuỗi “FILE”, gửi một tên file và truyền đến nội dung file độc hại vào server.

1. **Tài liệu tham khảo:**

<https://www.geeksforgeeks.org/tcp-server-client-implementation-in-c/>

<https://www.bogotobogo.com/cplusplus/sockets_server_client.php>