Báo cáo thực hành 6 lập trình mạng

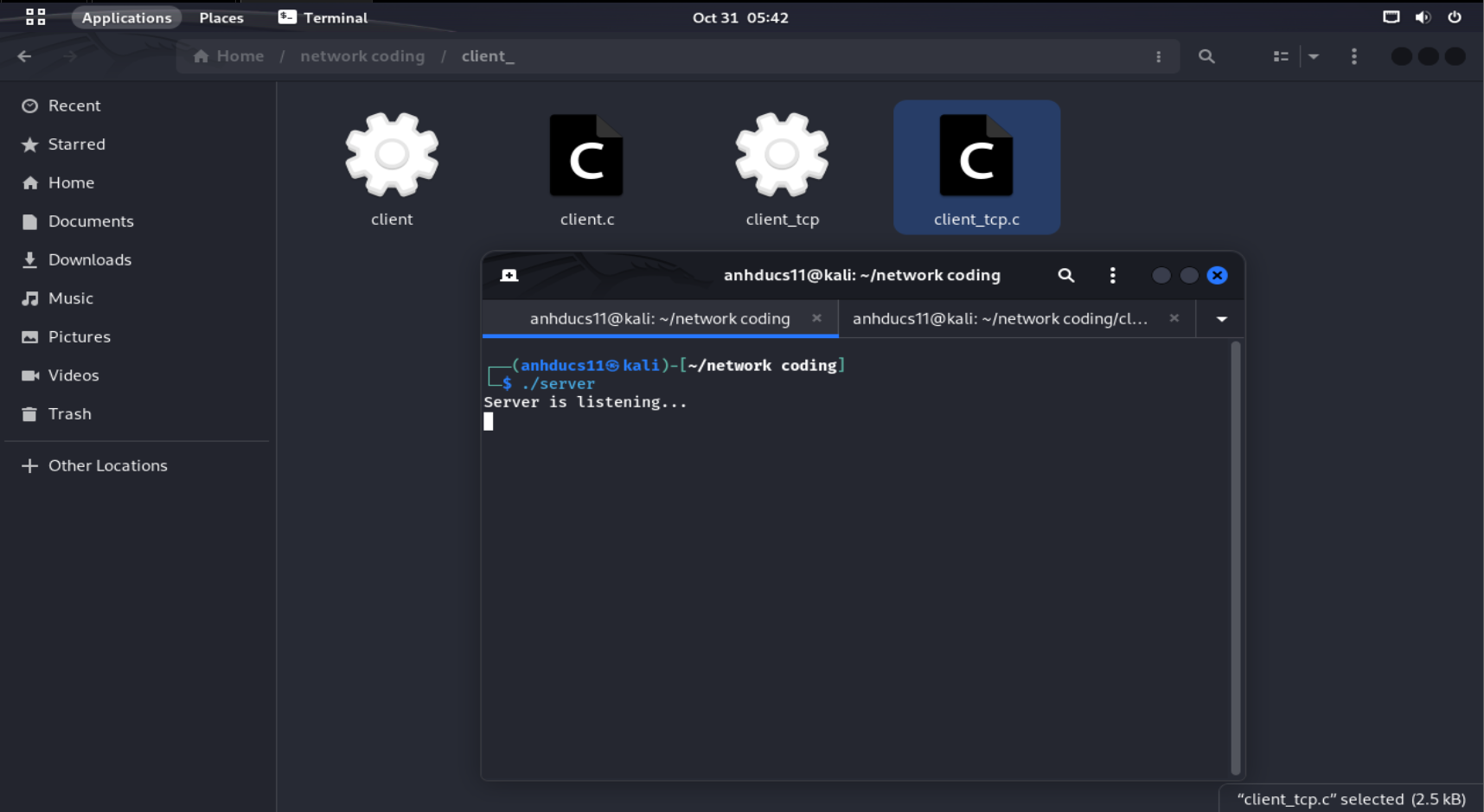
**Bài 1:**

**Chương Trình Server:**

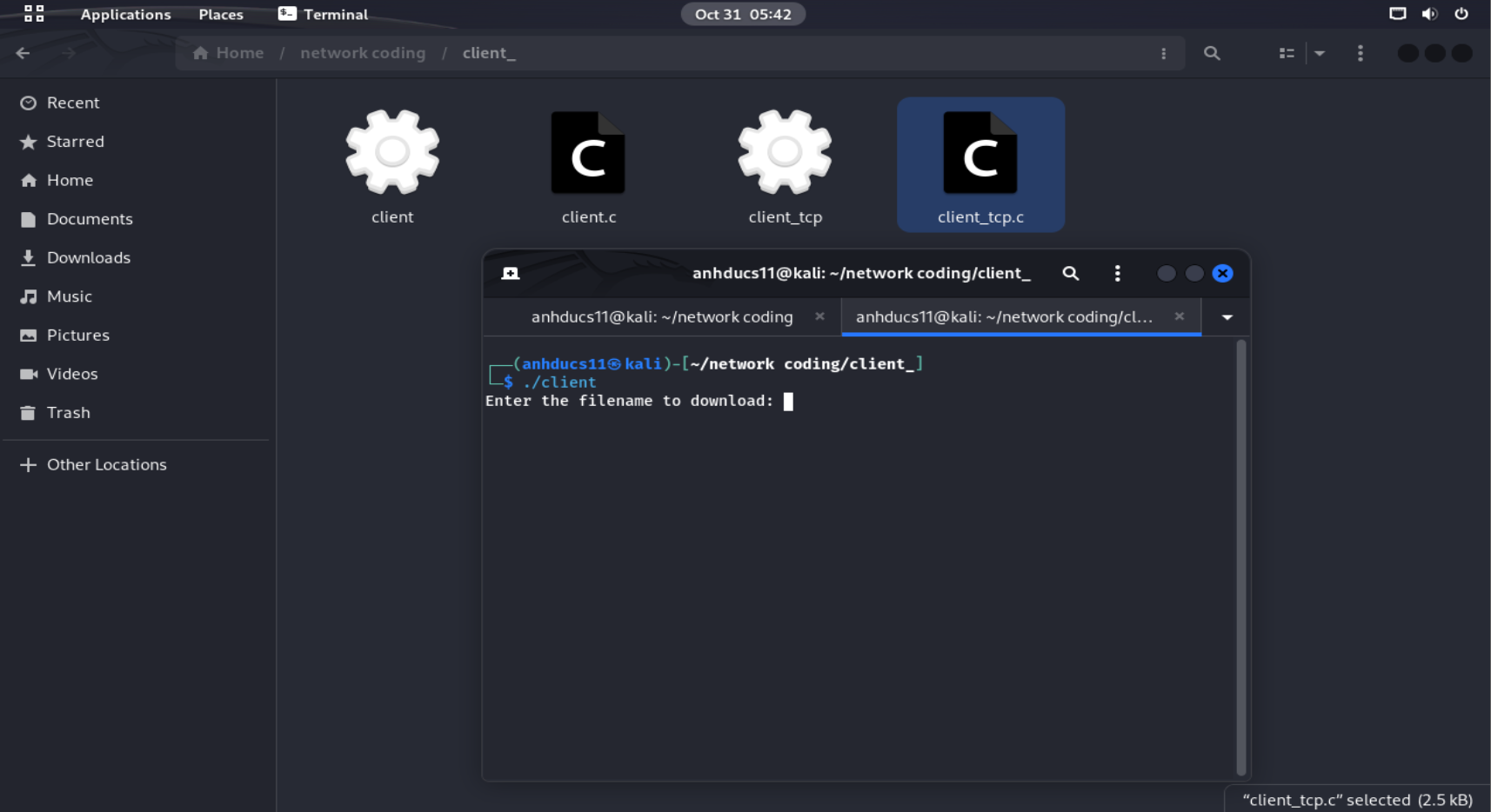
1. Chương trình máy chủ định nghĩa một máy chủ lắng nghe trên cổng 12345 và chấp nhận các kết nối từ máy khách.
2. Khi một máy khách kết nối đến, máy chủ tạo một luồng xử lý riêng (sử dụng POSIX threads) để xử lý kết nối với máy khách này. Luồng này được thực hiện trong hàm **handle\_connection**.
3. Trong hàm **handle\_connection**, máy chủ đọc lệnh từ máy khách (ví dụ: "DOWNLOAD") và sau đó đọc tên file mà máy khách muốn tải về.
4. Máy chủ kiểm tra xem file có tồn tại trong thư mục của máy chủ hay không. Nếu không tồn tại, nó gửi một thông báo lỗi "File not found" cho máy khách.
5. Nếu file tồn tại, máy chủ gửi "OK" đồng thời gửi kích thước của file và dữ liệu của file cho máy khách.
6. Sau khi máy khách đã nhận hết dữ liệu của file, nó gửi một thông báo "BYE" để kết thúc phiên làm việc.
7. Máy chủ đóng kết nối với máy khách và giải phóng tài nguyên sau khi hoàn thành xử lý.

**Chương Trình Client:**

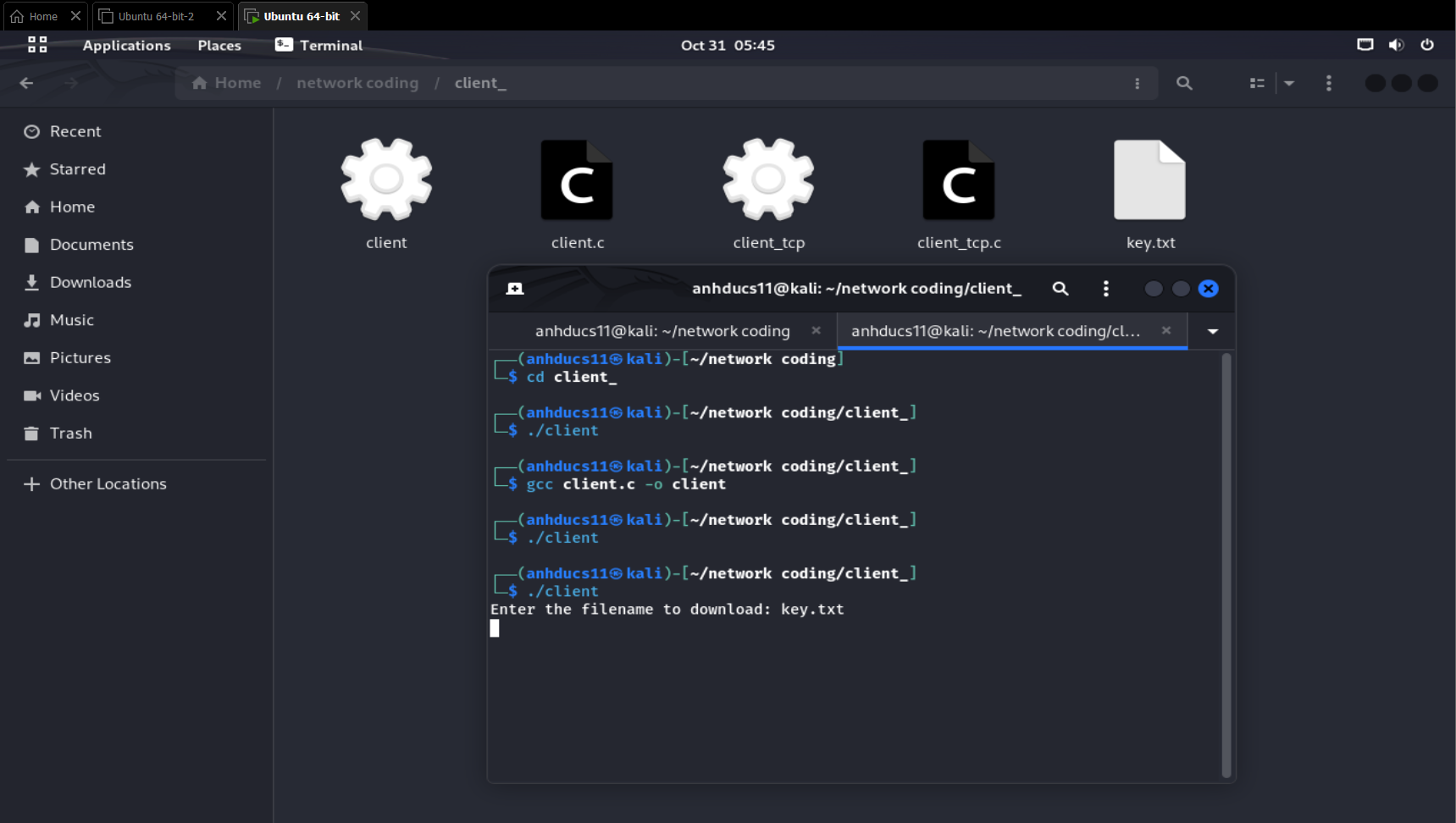
1. Chương trình máy khách tạo một socket và kết nối đến máy chủ qua địa chỉ IP "127.0.0.1" và cổng 12345 (máy chủ phải hoạt động trên cùng một máy tính).
2. Máy khách gửi lệnh "DOWNLOAD" đến máy chủ để bắt đầu phiên làm việc.
3. Sau đó, máy khách yêu cầu người dùng nhập tên file mà họ muốn tải về.
4. Máy khách gửi tên file cho máy chủ và sau đó chờ phản hồi từ máy chủ.
5. Nếu máy chủ trả về "File not found," máy khách in ra thông báo "File not found on the server."
6. Nếu máy chủ trả về "OK," máy khách tiếp tục nhận kích thước của file và dữ liệu file từ máy chủ, và lưu trữ file tải về trên máy khách.
7. Sau khi máy khách đã nhận hết dữ liệu của file, nó gửi một thông báo "BYE" để kết thúc phiên làm việc và đóng kết nối với máy chủ.



Chạy server



Chạy client



Sau khi download thì file key.txt đã có ở thư mục client

**Bài 2:**

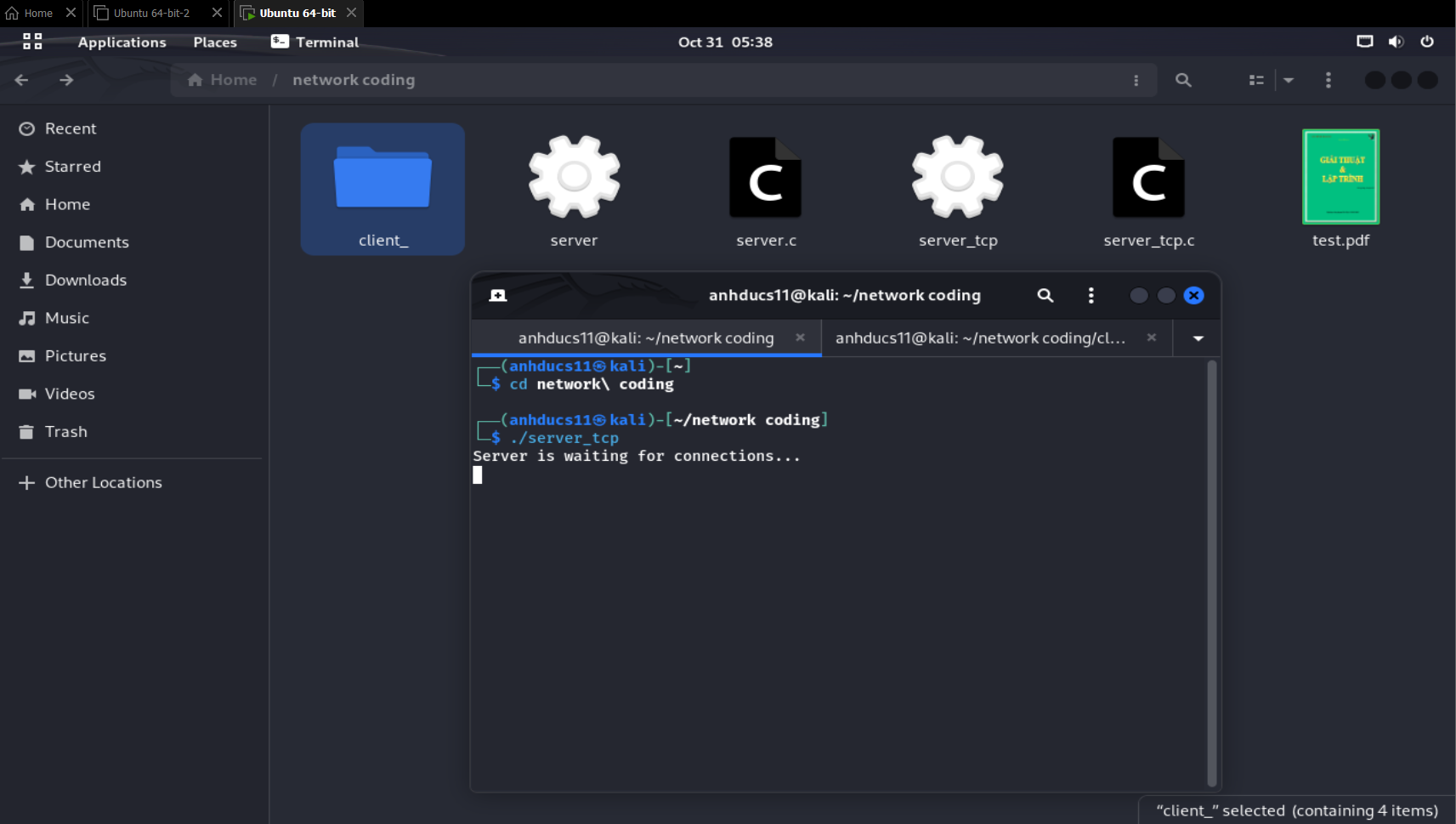
Mô tả chương trình:

**Server (server.c):**

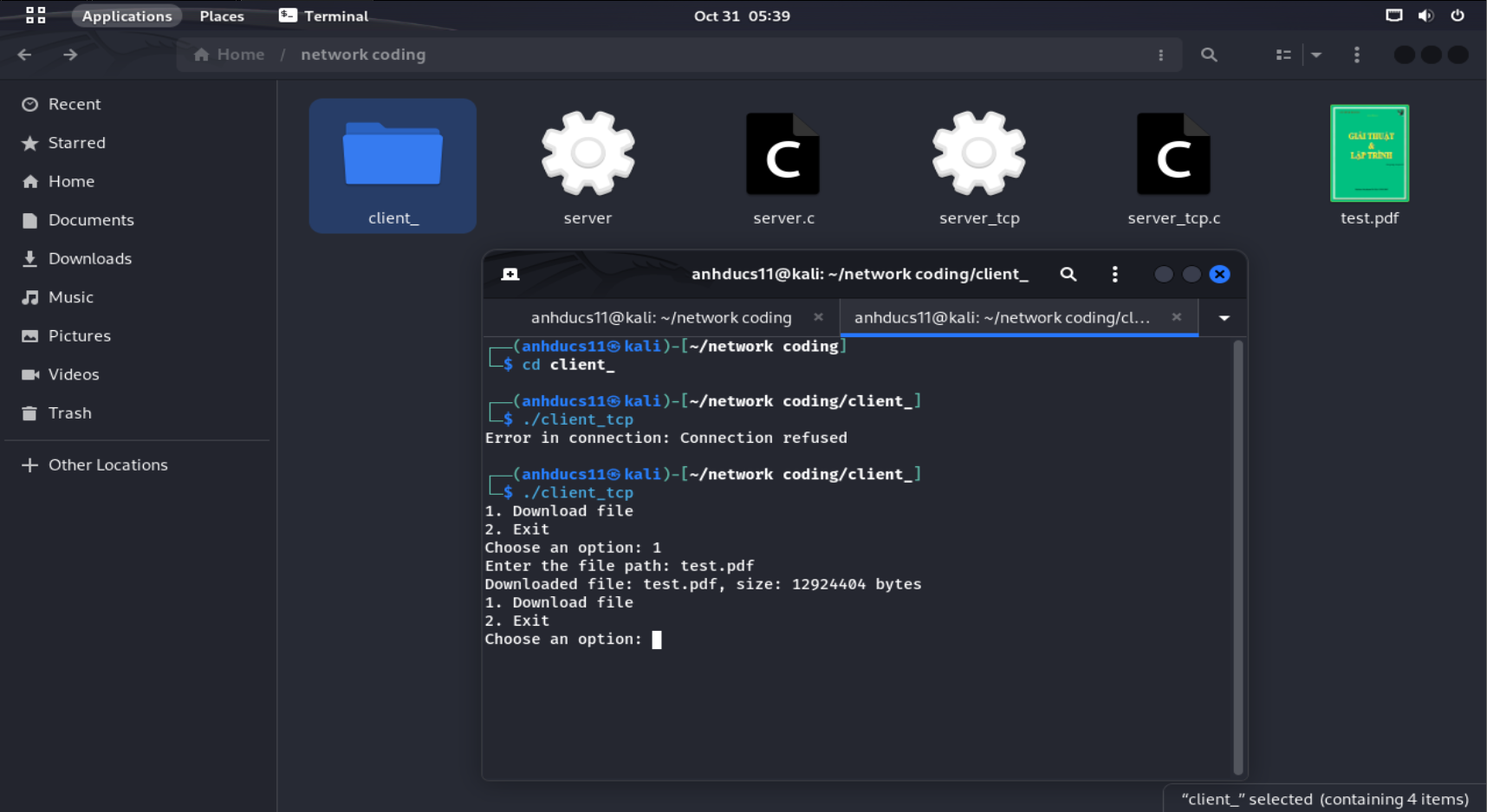
1. Chương trình server.c bắt đầu bằng việc tạo một socket sử dụng hàm **socket()**. Đây là điểm kết nối mà các client có thể kết nối.
2. Sau đó, nó cấu hình địa chỉ và cổng mà server sẽ lắng nghe sử dụng struct **server\_address**. Ở đây, cổng 12345 được sử dụng làm ví dụ.
3. Server gọi **bind()** để gán địa chỉ và cổng cho socket.
4. Sau đó, server sử dụng **listen()** để lắng nghe các kết nối từ client. Server có thể chấp nhận tối đa 5 kết nối đồng thời (được chỉ định trong hàm **listen()**).
5. Một vòng lặp vô hạn bắt đầu để chấp nhận kết nối từ client sử dụng **accept()**. Khi có một client kết nối, server sẽ thực hiện xử lý yêu cầu của client.
6. Server chấp nhận yêu cầu từ client, dựa trên lệnh client gửi. Hiện tại, chỉ có hai lệnh được hỗ trợ: "download" để tải file và "exit" để thoát.
7. Nếu client gửi lệnh "download", server nhận đường dẫn tới file mà client muốn tải và sử dụng hàm **send\_file()** để gửi file đó đến client.
8. Trong hàm **send\_file()**, server gửi tên và kích thước của file trước, sau đó, nó mở file và chia nhỏ nó thành các phần và gửi từng phần tới client.
9. Khi tải file hoàn thành, server đóng kết nối với client.

**Client (client.c):**

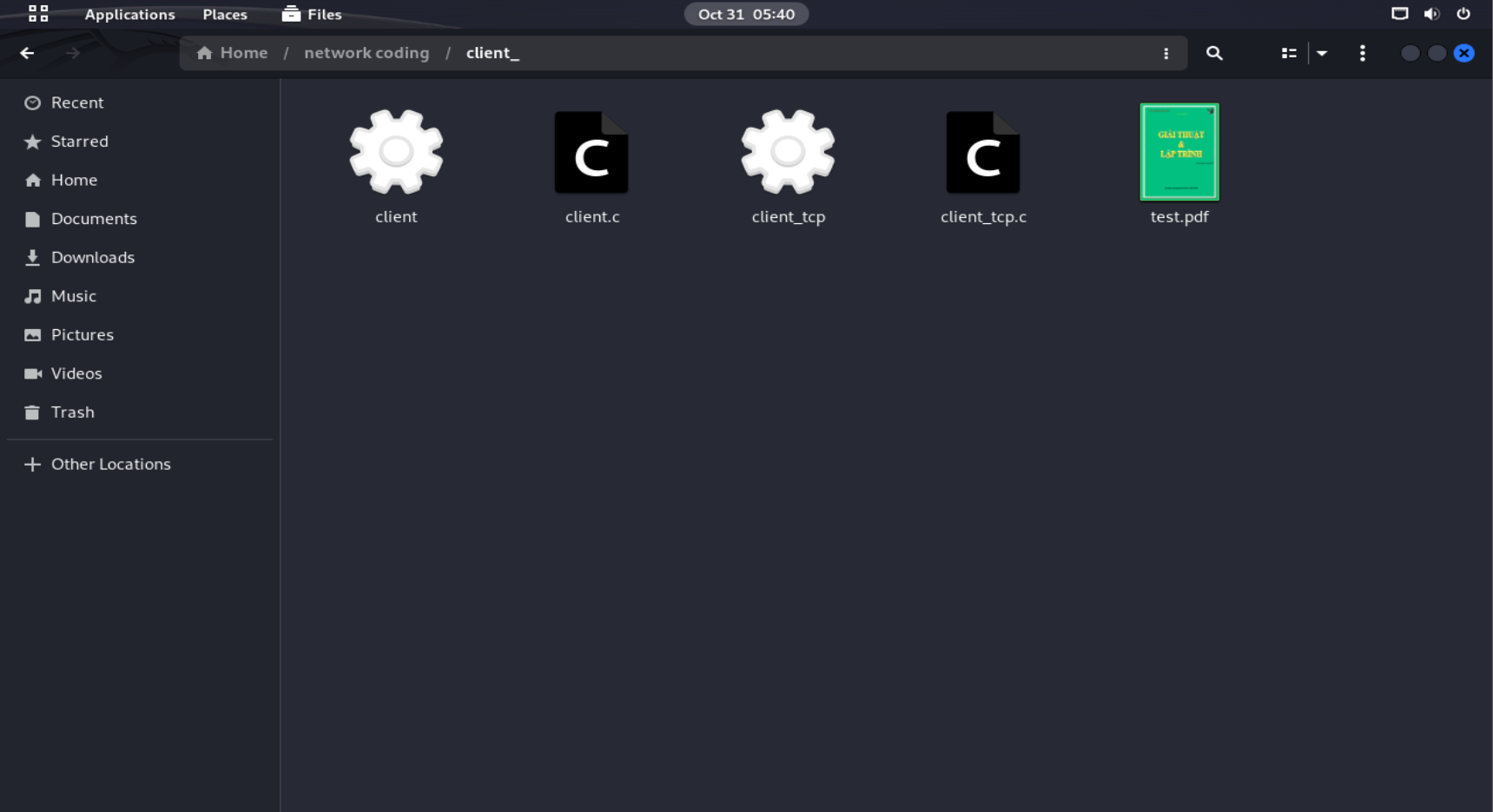
1. Chương trình client.c bắt đầu bằng việc tạo một socket sử dụng hàm **socket()**. Client cố gắng kết nối đến server với địa chỉ IP của server và cổng 12345.
2. Client hiển thị một menu lựa chọn với hai tùy chọn: "Download file" và "Exit".
3. Khi người dùng chọn "Download file," client gửi lệnh "download" đến server và sau đó yêu cầu người dùng nhập đường dẫn tới file mà họ muốn tải.
4. Sau khi nhận đường dẫn file từ người dùng, client gửi nó đến server.
5. Client nhận thông tin về tên và kích thước file từ server.
6. Sau đó, client tạo một file rỗng với tên file và bắt đầu nhận dữ liệu từ server và ghi vào file.
7. Khi tải file hoàn thành, client đóng file và hiển thị thông tin về tên và kích thước file đã tải.
8. Client cũng có khả năng tính toán md5sum của file và so sánh với md5sum mà server gửi. Nếu md5sum trùng khớp, file được tải xuống mà không bị thay đổi. Nếu không, client hiển thị thông báo về sự khác biệt.
9. Client có thể chọn "Exit" để đóng kết nối với server và thoát chương trình.



Chạy server



Chạy client và download file.



Sau khi chạy download thì file đã có ở thư mục client.