**Báo cáo thực hành 7 lập trình mạng**

Bài 1:

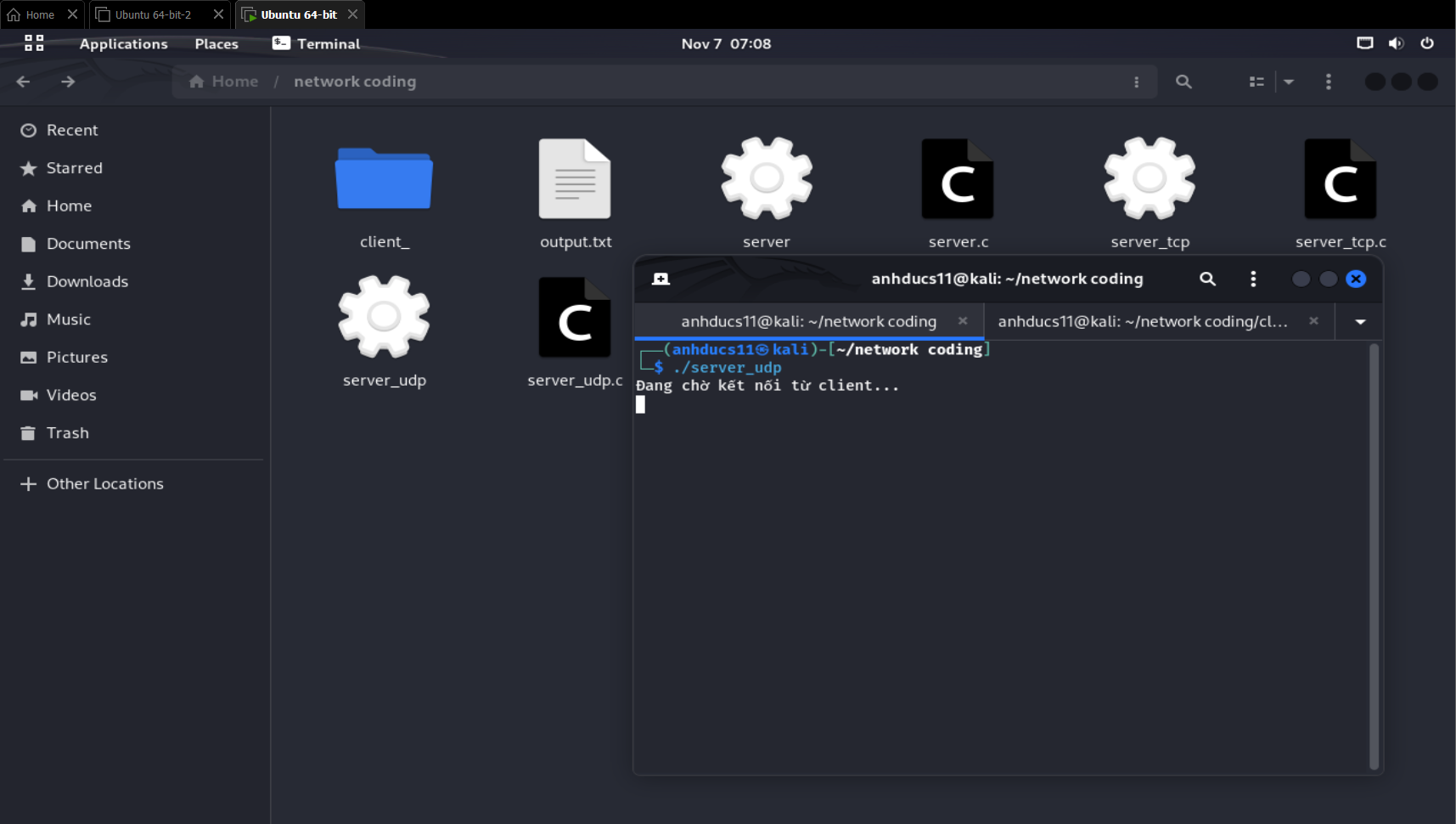
**Client**

1. Thư viện và hằng số:
   * Chương trình sử dụng các thư viện C chuẩn như **stdio.h**, **stdlib.h**, **string.h**, **sys/socket.h**, **netinet/in.h**, **arpa/inet.h**, và **unistd.h**.
   * Được định nghĩa một hằng số **MAX\_BUFFER** với giá trị là 1024, được sử dụng để đọc/gửi dữ liệu trong các bước sau.
2. Hàm **main**:
   * Hàm **main** là hàm chính của chương trình, và nó chứa toàn bộ logic của ứng dụng client.
3. Kiểm tra đầu vào:
   * Chương trình kiểm tra số lượng tham số dòng lệnh. Nếu không đủ 3 tham số, nó sẽ in hướng dẫn sử dụng và kết thúc chương trình với mã lỗi 1.
4. Tạo socket:
   * Sử dụng hàm **socket** để tạo một socket UDP (SOCK\_DGRAM). Nếu việc tạo socket không thành công, chương trình sẽ in ra thông báo lỗi và kết thúc với mã lỗi 1.
5. Định nghĩa địa chỉ server:
   * Chương trình xác định địa chỉ IP và cổng của máy chủ và lưu chúng trong biến **server\_addr**. Địa chỉ IP được lấy từ tham số dòng lệnh và chuyển đổi từ chuỗi sang địa chỉ IP bằng **inet\_aton**. Nếu địa chỉ IP không hợp lệ, chương trình sẽ in ra thông báo lỗi và kết thúc với mã lỗi 1.
6. Gửi tên file cho server:
   * Sử dụng hàm **sendto**, chương trình gửi tên file (tham số thứ hai từ dòng lệnh) đến máy chủ qua socket đã tạo. Nếu việc gửi tên file không thành công, chương trình sẽ in ra thông báo lỗi và kết thúc với mã lỗi 1.
7. Nhận và lưu file:
   * Chương trình mở một tệp tin (file) để ghi dữ liệu nhận được từ máy chủ. Nếu việc mở tệp tin không thành công, chương trình sẽ in ra thông báo lỗi và kết thúc với mã lỗi 1.
   * Sử dụng một vòng lặp, chương trình nhận dữ liệu từ máy chủ bằng hàm **recvfrom** và ghi dữ liệu nhận được vào tệp tin mở trước đó. Nếu có lỗi trong quá trình nhận hoặc ghi dữ liệu, chương trình sẽ in ra thông báo lỗi và kết thúc.
   * Vòng lặp tiếp tục cho đến khi không còn dữ liệu để nhận từ máy chủ.
8. Kết thúc chương trình:
   * Sau khi hoàn thành việc nhận dữ liệu và ghi vào tệp tin, chương trình đóng tệp tin và in ra thông báo thành công.
   * Cuối cùng, chương trình đóng socket và kết thúc với mã lỗi 0, chỉ ra rằng nó đã thực hiện xong mà không có lỗi nào xảy ra.

**Server**

1. Thư viện và hằng số:
   * Chương trình sử dụng các thư viện C chuẩn như **stdio.h**, **stdlib.h**, **string.h**, **sys/socket.h**, **netinet/in.h**, **arpa/inet.h**, và **unistd.h**.
   * Được định nghĩa một hằng số **MAX\_BUFFER** với giá trị là 1024, được sử dụng để đọc/gửi dữ liệu trong các bước sau.
2. Hàm **main**:
   * Hàm **main** là hàm chính của chương trình, và nó chứa toàn bộ logic của ứng dụng máy chủ.
3. Tạo socket:
   * Sử dụng hàm **socket** để tạo một socket UDP (SOCK\_DGRAM). Nếu việc tạo socket không thành công, chương trình sẽ in ra thông báo lỗi và kết thúc với mã lỗi 1.
4. Định nghĩa địa chỉ server:
   * Chương trình xác định địa chỉ IP và cổng mà máy chủ lắng nghe. Địa chỉ IP được đặt là **INADDR\_ANY**, cho phép máy chủ lắng nghe kết nối từ bất kỳ địa chỉ IP nào trên máy chủ.
   * Địa chỉ cổng được đặt là 5000.
5. Gán địa chỉ cho socket:
   * Sử dụng hàm **bind**, chương trình gán địa chỉ server cho socket đã tạo. Nếu việc gán địa chỉ không thành công, chương trình sẽ in ra thông báo lỗi và kết thúc với mã lỗi 1.
6. Vòng lặp chính:
   * Chương trình bắt đầu một vòng lặp vô hạn để liên tục lắng nghe kết nối từ các client.
   * Khi có kết nối từ client, chương trình nhận tên file yêu cầu từ client bằng cách sử dụng hàm **recvfrom**. Sau đó, nó in ra tên file này.
   * Nếu không thể mở tệp tin (file) với tên file yêu cầu, chương trình in ra thông báo lỗi và quay lại vòng lặp chính để chờ yêu cầu khác từ các client.
7. Gửi file cho client:
   * Nếu tệp tin mở thành công, chương trình bắt đầu gửi dữ liệu từ tệp tin cho client bằng cách sử dụng hàm **sendto**. Dữ liệu được gửi theo khung dữ liệu kích thước giới hạn bởi **MAX\_BUFFER**.
   * Chương trình tiếp tục gửi dữ liệu cho client cho đến khi tệp tin đã được gửi hoàn toàn hoặc có lỗi xảy ra trong quá trình gửi.
8. Kết thúc chương trình:
   * Sau khi gửi dữ liệu từ tệp tin hoặc nếu có lỗi xảy ra trong quá trình gửi, chương trình đóng tệp tin và in ra thông báo về việc gửi file thành công hoặc thông báo lỗi.
   * Chương trình tiếp tục lắng nghe kết nối từ các client trong vòng lặp chính.

Chạy Server



Chạy client

