Cse3027: Comuter Networks
Project 1: Socket Programming

myserver.c 코드 개요

이 프로그램은 클라이언트로부터 HTTP 요청을 받아들이고, 요청된 파일을 읽어 클라이언트에게 응답하는 기능을 수행합니다. 다중 클라이언트 요청을 처리하기 위해 프로세스를 fork하는 방식으로 구현되었습니다.

함수 설명
main 함수
프로그램의 진입점입니다. 사용자로부터 포트 번호를 입력받고, 서버 소켓을 생성하여 클라이언트의 연결을 대기합니다. 클라이언트의 요청이 들어오면 handle_request 함수를 호출하여 요청을 처리합니다.

handle_request 함수
클라이언트의 요청을 처리하는 함수입니다. 클라이언트와의 연결이 성공하면 새로운 프로세스를 fork하여 요청을 처리합니다. 클라이언트의 요청을 파싱하고, 요청된 파일을 찾아 send_response 함수를 호출하여 응답을 보냅니다.
send_response 함수
클라이언트에게 HTTP 응답을 보내는 함수입니다. 요청된 파일을 읽어서 클라이언트에게 전송합니다. 요청된 파일의 확장자에 따라 적절한 Content-Type 헤더를 설정하여 응답합니다.

get_file_size 함수
파일의 크기를 구하는 함수입니다. 파일 끝으로 이동한 후 현재 파일 위치를 가져와서 파일의 크기를 계산합니다.

send_file 함수
파일을 클라이언트에게 전송하는 함수입니다. 파일을 읽어서 BUFFER_SIZE 단위로 클라이언트에게 전송합니다.

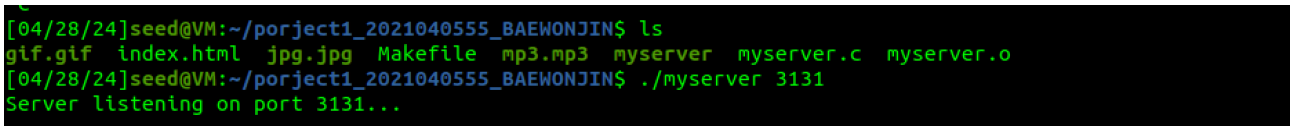
어려웠던 부분

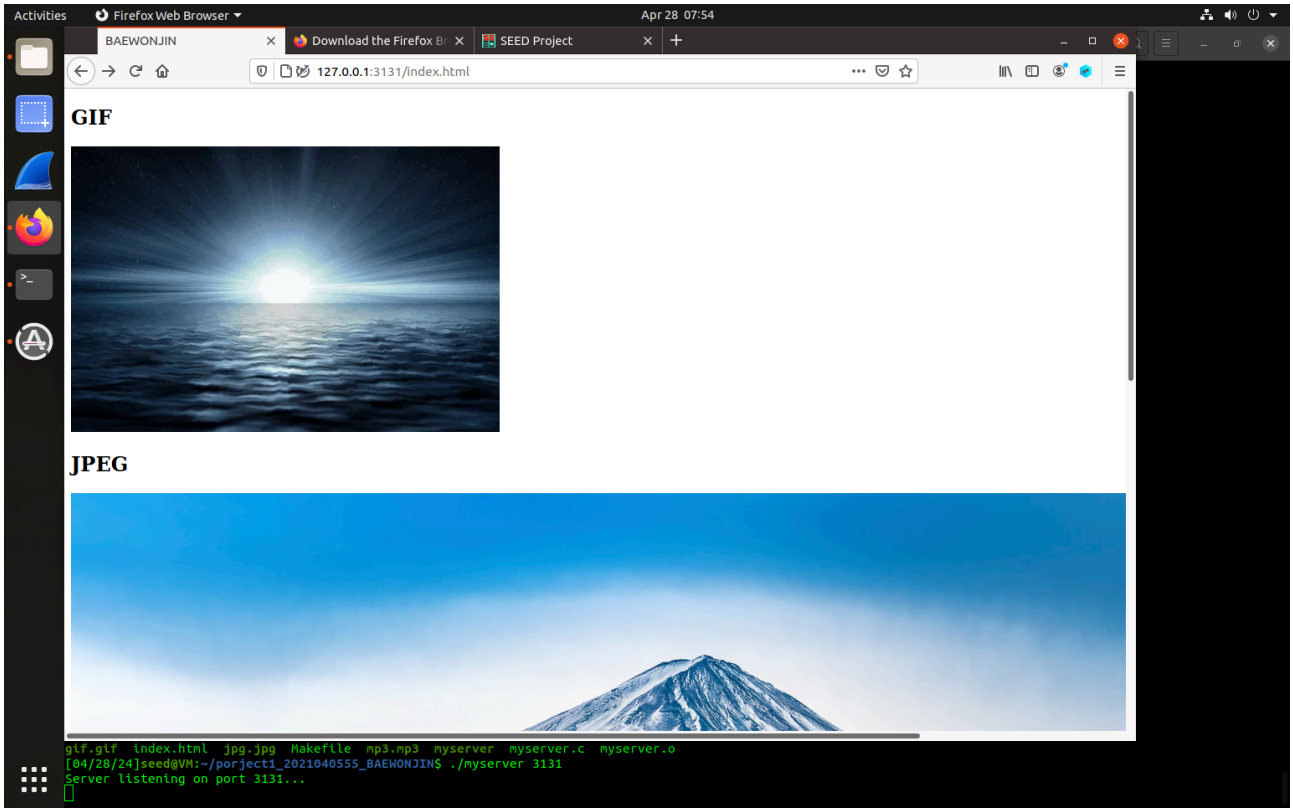
HTTP 헤더 처리하는 부분 : HTTP 요청과 응답의 헤더를 적절하게 처리하는 것이 어려웠습니다. 특히, Content-Type 헤더를 올바르게 설정하고 요청된 파일을 전송하는 과정에서 많은 어려움을 겪었습니다. 이를 해결하기 위해 파일의 확장자를 확인하여 적절한 Content-Type 헤더를 설정했습니다.

클라이언트 요청을 처리하는 부분에서 겪은 어려웠던 오류들:
파일 무한 읽기 루프: 파일을 읽어오는 과정에서 무한 루프가 발생했습니다. 이는 파일을 읽을 때 파일의 끝에 도달하여 fread 함수가 0을 반환하지 않아서 발생한 문제였습니다. 이를 해결하기 위해 파일을 읽을 때 파일의 끝에 도달했는지를 확인하고, 더 이상 읽을 데이터가 없을 때에만 루프를 종료하도록 수정하였습니다.

다중 클라이언트 요청 처리: 웹 서버는 다수의 클라이언트 요청을 동시에 처리할 수 있어야 합니다. 하지만 기존 코드는 한 번에 하나의 클라이언트 요청만을 처리하는 방식으로 동작하고 있었습니다. 이를 해결하기 위해 클라이언트 요청을 처리하는 함수인 handle_request에서 자식 프로세스를 생성하여 요청을 처리하도록 변경하였습니다. 이를 통해 여러 클라이언트의 동시 요청을 처리할 수 있게 되었습니다.

결과 화면





이처럼 3131포트에서 만들어 놓은 index.html 파일이 잘 열리는 것을 확인할 수 있습니다.