

Programming Assignment #5

Due: November 27, 11:59:59 PM

1. Introduction

- 이번 과제를 통해 HTTP/1.0을 따르는 멀티 쓰레드 웹 서버를 만들어본다.

2. Overview

- PA #4에서 작성했던 웹 서버에서, 새로운 접속이 발생할 때마다 새로운 쓰레드를 생성하여 동시에 여러 요청을 처리한다.

3. Specification

- (참고) 여기 언급되지 않은 기본적인 서버의 동작은 PA #4에서 했던 것과 동일하다.
- (참고) `make_http_header()`와 `make_report()`는 thread-safe하다.
<서버 생성>
 - 서버의 기본 포트는 1234이고, 프로그램 인자가 1개 존재하면 그것을 포트로 쓴다.
 - 소켓에 접속 요청이 들어오면, 연결을 맺고 `pthread_create()`로 새로운 쓰레드를 생성하여 해당 연결을 전담하여 처리하게 하고, 메인 쓰레드는 바로 다음 요청을 처리하기 위해 대기한다.
 - 쓰레드를 생성할 때, **원치 않는 쓰레드간 공유**가 발생하지 않도록 한다. (e.g. Socket descriptor)
<요청의 처리>
 - 소켓에서부터 클라이언트의 요청을 전부 읽어낸다.
 - ✓ (참고) 채점시 서버로 들어오는 한 요청의 크기는 1024바이트를 넘지 않는다.
 - 클라이언트의 요청의 첫째 줄을 분석해 적절한 문서를 찾는다.
 - ✓ 클라이언트가 요청하는 파일이 `"/stat.html"` 일 경우, 현재까지 전송한 파일에 대한 기록을 `make_report()` 함수를 이용하여 생성한 뒤 보내준다.

- 파일 확장자는 html, png, gif, jpg만 허용한다. (다른 확장자는 접근할 권한이 없다)

<응답의 처리>

- 클라이언트의 요청에 대응하는 응답을 소켓으로 보낸다.
- HTTP 헤더는 다음 정보를 이용해 만든다.
 - ✓ http.c 에 구현된 make_http_header() 함수를 이용하면, HTTP 헤더로 사용할 수 있는 문자열을 동적으로 생성해준다. **이 헤더를 사용한 후 반드시 free()로 할당된 자원을 해제하여야 한다.**
 - ✓ Status-code는 다음 3 경우만 고려한다.

정상일 때	NO_ERR
해당 파일에 접근할 권한이 없을 때	ERR_NOT_ALLOWED
파일이 존재하지 않을 때	ERR_NO_FILE

- ✓ Content-length는 해당 파일의 크기이다. <파일의 내용>의 크기는 Content-length와 일치하여야 한다.
- ✓ Content-type은 해당 파일의 확장자에 따라서 http.h에 선언된 적절한 enum 값을 이용한다.

HTML 파일	TYPE_HTML
JPG 파일	TYPE_JPEG
PNG 파일	TYPE_PNG
GIF 파일	TYPE_GIF
모든 에러 발생	TYPE_UNKNOWN

- 파일의 내용은 기존에 하던 것처럼 해당 파일을 열어 읽고 소켓에 쓴다.
 - ✓ 만약 에러가 발생한 경우, "loading failed"이란 문자열을 <파일의 내용>으로 간주하고 소켓에 쓴다. 이 경우 Content-length는 14바이트이다. (문자열 길이)

- 전송을 마치고 소켓을 닫는다.

<연결 종료 이후>

- 메인 스레드는 스스로 종료하지 않고 무한히 연결을 기다린다.
- 자식 스레드는 클라이언트에게 전송을 마치면 **수행 기록**을 남기고 종료한다.
 - ✓ 부모 스레드가 pthread_join() 을 할 수 없기 때문에, 메모리 유출을 막기 위해 자식 스레드는 종료되기 전 반드시 **detached** 상태이어야 한다.

<수행 기록의 생성과 전송>

- 자식 쓰레드가 종료되기 전, 다음 수행 기록을 업데이트한다.
 - ✓ 발생한 커넥션의 횟수 (자식 쓰레드가 종료되기 직전에 업데이트할 것)
 - ✓ 헤더를 제외하고 보낸 데이터 크기의 합(bytes). Content-length의 합이라 생각해도 좋다.
 - ✓ 각 파일 확장자마다, 정상적으로 전송한 횟수의 합 (make_http_header() 함수에서 사용했던 것과 동일함)
- 여러 쓰레드에서 동시에 수행 기록을 업데이트 하지 않도록 lock 등을 이용하여 동기화한다.
- "/stat.html" 파일을 요청받으면, make_report() 함수를 호출해 수행 기록에 대한 리포트를 만든다.
 - ✓ `char *make_report(int connections, long long int sent_bytes, int html, int jpeg, int png, int gif, int unknown);`
 - ✓ http.c 에 구현된 make_report() 함수는 인자를 바탕으로 수행 결과에 대한 리포트를 동적으로 만들어준다.
 - ✓ "/stat.html"이 요청된 시점에 커넥션이 3번 발생, 총 2400 바이트를 전송, html, jpg, hwp 파일을 요청 받았다면 다음과 같이 make_report() 함수를 호출한다.
 - ✓ `make_report(3, 2400, 1, 1, 0, 0, 1);`
 - ✓ 확장자가 정상이지만 에러가 발생해 전송하지 못한 경우, make_http_header()에서 하는 것과 같이 unknown이 하나 증가하여야 한다.
 - ✓ `connections = html + jpeg + png + gif + unknown` 을 만족하여야 한다.
- make_report() 로 만들어진 문자열을 <파일의 내용>으로 클라이언트에게 전송한다.
 - ✓ 이 때 HTML 파일을 정상적으로 전송한 것으로 간주한다.
 - ✓ 파일의 크기는 make_report() 이 생성한 문자열의 길이이다.
- 전송 이후 자기 자신에 대한 수행 기록을 업데이트하고, free()로 할당받은 메모리를 해제한 뒤 쓰레드를 종료한다.

4. Restriction

- 과제는 본인이 직접 설치한 리눅스 환경에서 구현한다.
 - ✓ 테스트 서버에서 컴파일, 실행시킬 수 없는 코드는 과제 제출물로 인정하지 않는다.

- ✓ 테스트 서버는 Linux kernel 3.0-32bit, gcc 4.6.1 를 이용한다.
- 시스템 콜을 사용하여 프로그램을 작성하며, Standard C library의 malloc(), free(), getenv() 이외 함수는 사용하지 않는다.
- 파일 입출력의 경우 open(), read(), write(), close(), lseek() 시스템 콜 함수 중 필요한 것을 선택하여 사용한다.
- 소켓의 경우 강의 교안에 제시된 함수와 관련 함수를 사용한다.
- 스레드 생성과 동기화는 강의 교안의 제시된 함수를 이용한다.
- 추가적으로 stat(), fstat() 시스템 콜을 사용할 수 있다.
- 필요한 경우 함수를 직접 구현하여 사용한다.
- 본 과제에서는 webserver.c 만 수정할 수 있다.
- 어떤 자원을 할당 받으면, 프로그램 종료 전에 반드시 해당 자원을 **직접 해제**한다.
- 컴파일 시 발생하는 모든 warning을 잡아야 한다.

5. Skeleton Codes

- 본 과제 수행을 위하여 아래와 같이 7개의 파일이 주어진다.
 - ✓ Makefile GNU make utility를 위해 사용되는 파일.
 - ✓ webserver.c server2 를 구현해 넣을 파일. 기본 코드가 구현되었다.
 - ✓ http.c 제공 함수들이 구현된 파일
 - ✓ http.h 프로그램 작성에 필요한 정보가 있는 헤더 파일
 - ✓ index.html 기본 제공 server2 프로그램을 테스트할 수 있는 페이지
 - ✓ daum.png index.html을 그리기 위해 포함된 그림 (다음 로고)
 - ✓ naver.gif index.html을 그리기 위해 포함된 그림 (네이버 로고)

6. Hand in instruction

- 작성한 프로그램 코드 상단에 이름과 학번을 주석으로 표기한다.
- 작성한 과제 코드는 "**학번.tar.gz**" 형태로 압축해 <http://sys.skku.edu>에 제출한다.
 - ✓ 작업 디렉토리에서 "\$ make tar 학번" 명령을 이용해 파일을 "학번.tar.gz" 으 로 압축할 수 있다.
- 작성한 과제 코드의 디자인과 별도로 구현에 관한 내용을 담은 보고서를 "**학번.pdf**" 파일로 채점 서버에 제출한다.

7. Logistics

- 과제 제출 결과는 <http://csl.skku.edu/SSE2030F13/Assignments> 에서 확인할 수 있다.
- 과제 제출 기한은 <http://sys.skku.edu> 서버시간을 기준으로 하며, 기한 이후 24시간 내에 제출할 경우 30%, 48시간 내에 제출할 경우 60% 감점한다. 그 이후 제출이 불가능하며, 0점 처리한다.
- **본 과제에선 문서를 잘 작성하는 것이 중요하다.**
 - ✓ 동기화 등 여러 이슈를 해결하기 위해 어떤 디자인을 이용했는지 설명한다.
- 과제 점수는 컴파일/실행 가능 여부, 작성한 함수의 완성도, 출력 결과 및 작성한 문서에 의하여 평가된다.
- 과제에 대한 의논은 함께 할 수 있으나, 프로그램 소스코드 작성은 스스로 해야 한다.
 - ✓ 다른 사람의 과제를 copy 한 경우, 두 사람 모두 0점 처리하며 **학점상의 불이익이 있다.** 인터넷 등에서 찾은 소스 코드를 그대로 copy 한 경우에도 0점처리한다. **두 번 이상 적발되면 F 학점**을 받는다.