

Programming Assignment #6

Due: December 22, 11:59:59 PM

1. Introduction

- 이번 과제를 통해 HTTP/1.0을 따르는 웹 서버를 만든다.

2. Overview

- 과제 4, 5에서 만든 웹 서버를 빠르게 수행시키기 위해 어떻게 디자인 할 지 연구한다.

3. Specification

- 본 과제에서 제공하는 스켈레톤 코드는 최소한의 기능만 구현된 웹 서버이다.
 - ✓ 주어진 요청에 맞춰 순차적으로 하나씩 파일을 열어 클라이언트에게 전송하며, 에러 처리는 하지 않는다.
- 서버를 빠르게 수행하도록 만들기 위해 **제약 사항을 지키는 범위 내에선 어떤 방법을 사용해도 좋다. 원한다면, 처음부터 코드를 작성해도 된다.**
 - ✓ 프로세스, 스레드 중 무엇을 사용할지, 아니면 둘 다 사용할지 고민할 수 있다.
 - ✓ 동시에 몇 개의 요청을 처리할 것인지 등을 고려할 수 있다.
- 구현해야 하는 서버의 역할은 PA #4, 5와 같다. 아래는 예외 처리 부분을 강조하기 위해 다시 한번 반복한 것이다.
 - ✓ /stat.html, /index.html은 이번 과제에서 구현할 필요가 없다.

<서버 생성>

<요청의 처리>

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

<응답의 처리>

- 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)

<연결 종료 이후>

- 서버는 스스로 종료하지 않고 끊임없이 연결을 받아 처리할 수 있어야 한다.

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, 320GB HDD를 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.

- 프로세스 생성, 쓰레드, 동기화 관련 함수는 강의 교안의 제시된 함수를 이용한다.
- 추가적으로 `stat()`, `fstat()` 시스템 콜을 사용할 수 있다.
- 필요한 경우 함수를 직접 구현하여 사용한다.
- 본 과제에서는 `server3.c` 만 수정할 수 있다.
- 어떤 자원을 할당 받으면, 반드시 해당 자원을 **직접 해제**한다.
 - ✓ 예를 들어, 자식 프로세스가 종료되면 부모에 의해 처리되어야 하고, 쓰레드가 생성되었다면 `detached` 상태여야 한다.
- 컴파일 시 발생하는 모든 **warning**을 잡는다.

5. Skeleton Codes

- 본 과제 수행을 위하여 아래와 같이 4개의 파일이 주어진다.
 - ✓ `Makefile` GNU make utility를 위해 사용되는 파일.
 - ✓ `server3.c` `server3` 를 구현해 넣을 파일. 기본 코드가 구현되었다.
 - ✓ `http.c` 제공 함수들이 구현된 파일
 - ✓ `http.h` 프로그램 작성에 필요한 정보가 있는 헤더 파일
- 본 스켈레톤 코드를 `make`를 이용하여 컴파일하면 `-O2` 옵션으로 코드를 최적화한다.

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> 서버 시간을 기준으로 하며, **지연 제출은 불가하다.**
 - ✓ 실습 서버 구현 상, 과제 제출 기한이 이틀 전으로 설정되고 실제 제출 기한은 지연 제출이라고 보이는데, 실제로는 서버에 제출이 가능한 마지막 순간까지가

본 과제의 기간이다.

- 본 과제도 문서를 잘 작성하는 것이 중요하다.
 - ✓ 본 과제를 만들기 위해 서버를 어떤 식으로, 왜 그렇게 디자인했는지 설명한다.
 - ✓ 구현한 코드에 대한 설명이 충분하지 않을 경우, 부분 점수만 받는다.
- 본 과제는 서버가 정해진 일(테스트 케이스)을 수행하는데 얼마나 시간이 걸리는지를 기준으로 평가한다.
 - ✓ 서버가 실행되면 클라이언트가 정해진 개수의 요청을 서버에게 병렬적으로 전송하는데, 첫 번째 요청이 발생한 순간부터 마지막 요청에 해당하는 파일이 전송되는 순간까지를 기준으로 시간을 microsecond 단위로 측정한다.
 - ✓ 제출한 코드가 정상적으로 수행되면 실행 시간을 기준으로 순위를 매기고, 이는 서버 Ranking에 실시간으로 공지된다.
 - ✓ 서버가 빨리 수행될수록 더 높은 순위를 받고, 최종 순위는 과제 마감 이후 순위를 기준으로 한다.
 - ◆ 과제 제출이 종료되기 전, 본인에게 가장 유리한 결과에 final 버튼을 누른다.
 - ✓ 본 과제를 31초(31000000 us) 안으로 수행시키면 만점을 받고, 만점 대상자에 한해 순위 상위 N명은 과제 코드 점수에 다음의 추가 점수를 받는다.
 - ◆ 1등 50%
 - ◆ 2등 30%
 - ◆ 3등 20%
 - ◆ 4-8등 10%
 - ◆ 9-16등 5%
- 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.