

PA 5 코드 설명

Global variables

1. connections, sent_bytes, ..., html, css, error의 변수는 각 파일이 전송된 횟수를 의미한다.
2. thread_cnt는 현재 살아있는 유효한 thread의 개수를 의미한다. (main thread 제외)
3. reset_flag는 현재 reset 상태인지 아닌지 판별하는 변수이다.

main function

1. main thread는 accept 상태에서 대기하다가 client의 요청이 들어오면 받아들이고, connfd (socket descriptor)를 지정해준다.
2. 이제 pthread_create로 이를 담당할 thread를 생성하여 connfd를 parameter로 전달한다.
3. connfd의 의도치 않은 공유를 막고, 새로운 thread 생성으로 예상치 못하게 connfd가 바뀌는 것을 막기 위해, malloc을 통해 공간을 동적 할당 한 후에 connfd를 저장하였다.
4. detach 옵션을 사용하여 pthread_join이 필요 없도록 하였다.

client_proc function

1. mutex lock을 해준 후에 현재 thread_cnt를 늘려준다.
2. 각종 필요한 변수를 초기화 해준 후, client에게서 request message를 받아온다.
3. request message로부터 client가 요구하는 파일의(또는 경로의) 이름과 확장자를 알아낸다.
- 4-1. **stat일 경우** 파일 형식을 HTML로 지정해준 후, report를 생성한다. report를 생성하는 도중 해당 parameter 값들이 변경되지 않도록 lock를 걸어준다. 그 후, 해당 report의 크기를 알아내고 http header를 생성한다.
- 4-2. (1) **reset일 경우** 파일 형식을 HTML로 지정해준다. reset_flag = 1로 지정하여 현재 reset 상태임을 알린다. 이렇게 되면 main 함수 내부의 thread도 현재 reset thread가 끝날 때까지 대기 상태에 빠지게 된다.
- 4-2. (2) 각 thread가 끝날 때마다 thread_cnt가 1씩 줄어들게 되는데, 그때마다 pthread_cond_wait에 빠진 reset thread에게 thread_cnt가 줄어들었음을 알리게 된다.
- 4-2. (3) thread_cnt가 1이 되게 되면 (reset thread만 남은 상태라면), 대기 상태를 빠져 나

오게 된다. 이제 `sent_bytes = 0`으로 초기화 시키고 나머지는 `stat`과 같은 행동을 하게 된다.

4-3. (1) 만약 명령어가 아닌 파일요청이 들어오게 되면 해당 파일 이름(경로)로 `open`을 한다.

4-3. (2) `fd < 0` 이면, `http header`에 `TYPE_ERROR` 및 에러 타입을 입력한다.

4-3. (3) `fd > 0` 이면, 확장자를 판별하여 유효한 확장자인지 검사한다. 유효한 확장자가 아닌 경우에는 `E_NOT_ALLOWED`와 동일하게 처리한다. 만약 유효한 확장자라면, `lseek`를 통하여 파일 크기를 알아낸 뒤, `content`에 해당 파일 내용을 복사해주고, `http header`을 만들게 된다.

5. `http header`와 `null line`을 전송한다.

6. 만약 상단 과정에서 `error`가 발생하여 보낼 `content`가 존재하지 않을 경우, 아무것도 보내지 않는다 (`file_size = 0`). 상단 과정이 정상적으로 진행되었다면, `file_size`만큼 파일을 `client`에게 전송하게 된다.

7. 요청한 자원을 모두 해제한다.

8. 전역 변수가 보호되기 위해 우선 `lock`를 걸어준다. 수행 기록을 업데이트하고, `thread_cnt`를 1 감소시킨다. 그 후, `thread_cnt`가 감소되었다는 `signal`을 발생시킨다. 만약 해당 `thread`가 `reset thread` 였다면, `main` 함수가 정상적으로 재개하도록 `restart condition`에 또한 `signal`을 보낸다.

Appendix

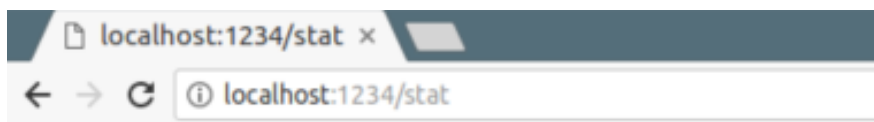
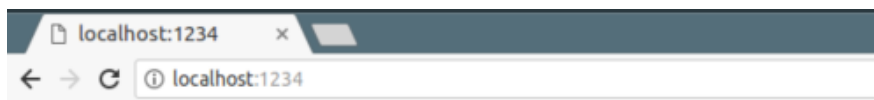
1. 웹 브라우저에서 직접 시험해본 결과, 무리하게 반복하여 새로고침 하여도, `connfd`의 충돌이나 다른 자원의 충돌이 발생하지 않아 무한 로딩이나, 이미지 로드 실패와 같은 오류가 발생하지 않는다.

2. `favicon.ico` 등의 이슈로 최초 실행 시 `error`이 1 ~ 2 카운트 되는 것을 확인할 수 있다. `a hover`과 같은 하이퍼링크의 태그의 경우 `thread_cnt`가 감소하지 않는 오류가 발생한다. 이는 과제에 제시되지 않은 `client` 요청의 또 다른 종류가 원인인 것으로 추정된다.

3. `string_sw.c`는 별도로 구현하여, `template`의 `string_sw.c`와는 차이가 있어 혼란을 방지하고자 `webserver.c` 외에 컴파일에 필요한 모든 파일을 포함하였다. 또, `학번.c`가 컴파일 되도록 `Makefile`을 수정하였다.

3. 웹 브라우저 실행 화면

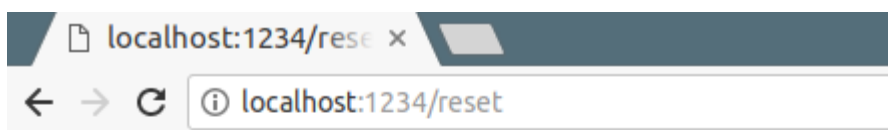
하단 사진 참조



Status Report for webserver

34 connections / 72457 bytes sent

html	jpg	png	gif	js	css	error
11	0	11	11	0	0	1



Status Report for webserver

32 connections / 0 bytes sent

html	jpg	png	gif	js	css	error
17	0	6	6	0	0	3