

Programming Assignment #6

Due : 20th Jun. (Mon), 11:59 PM

1. Introduction

이번 과제의 목표는 Multi-thread 방식의 서버를 구현하여 thread를 이용한 multi-thread 프로그래밍과 concurrent 프로그래밍 그리고 socket 사용에 익숙해지도록 하는 것이다.

2. Problem specification

- ✓ PA4 과제에서의 Server는 한 client에게 서비스하고 있는 동안 다른 client의 서비스 요청을 처리할 수 없다. 한번에 하나만의 요청을 처리하는데, 이를 개선하여 여러 client로부터 들어오는 서비스를 동시에 처리할 수 있는 다중 접속 서버의 구조로 바꾼다. client로부터의 서비스 요청에 대해 새로운 쓰레드를 생성하여 처리한다.
- ✓ Server, Client의 세부 동작은 모두 PA5와 동일하며, 검색 조건은 이전 과제에서 수행했던 것들과 모두 동일하다.
- ✓ Zombie thread가 발생하지 않도록 한다.
- ✓ !q를 입력하는 경우 server와의 통신을 종료한다.
- ✓ Server를 실행하는 양식은 다음과 같다.
 - ./server [IP주소] [Port] [성서1] [성서2] [성서3] ...
 - ex) ./server 115.145.175.21 6578 genesis daniel exodus
- ✓ Client를 실행하는 양식은 다음과 같다.
 - ./client [IP주소] [Port]
 - ex) ./client 115.145.175.21 6578
- ✓ 즉, server를 실행할 때 검색할 대상이 되는 성서들을 미리 argument로 넣어줄 것이며, 이에 대해서 server는 해당 성서에 대한 서비스를 제공해야 한다.
- ✓ 추가로, PA6에서는 다음과 같은 기능을 구현하도록 한다.
 - 1. Client가 서버로 "!history_recent"를 전송하면, Server는 최근 검색한 키워드 10개를 시간 순으로 정렬하여 client로 전송한다.
 - 가장 최근에 검색한 내역이 가장 위로 가고, 가장 나중에 검색한 내역이 가장 밑으로 가도록 한다.
 - 중복되는 경우는 없도록 한다.
 - 최근 검색한 키워드가 10개 이하일 경우 검색한 키워드에 대해서만 시간에 대한 순서로 정렬하여 출력한다.
 - 출력 양식은 아래와 같다.
 - ◆ (순위:숫자). (키워드) : (검색횟수)

- 2. Client가 서버로 "!history_most"를 전송하면, Server는 가장 많이 검색된 키워드 10개를 많이 검색한 순으로 정렬하여 client로 전송한다.
- 가장 많이 검색된 키워드가 가장 위로 가게 정렬한다.
- 중복되는 경우는 없도록 한다.
- 동점이 있는 경우, 더 최근에 검색한 적이 있는 키워드가 더 높은 순위에 위치하도록 한다. (3.client example 참고)
- 키워드가 10개 이하일 경우 검색된 키워드에 대해서만 정렬하여 출력하도록 한다.
- 출력 양식은 아래와 같다.
 - ◆ (순위:숫자). (키워드) : (검색횟수)

3. Client Example

- Server에서의 출력은 PA4, 5의 example과 같이 하도록 한다.
- Client에서의 출력은 기본적으로 PA4, PA5와 같으나, 새로 추가된 !history_recent, !history_most에 대해서는 아래와 같이 출력한다.

```
./client 115.145.175.21 30000
```

```
...
```

```
...
```

```
> !history_recent
```

```
1. the : 20
2. and : 17
3. why : 41
4. and she said : 1
5. "and * he * said" : 4
6. god created : 102
7. god : 140
8. peace : 4
9. devil : 1
10. kingdom : 11
```

```
> !history_most
```

```
1. god : 140
2. god created : 102
3. why : 41
4. the : 20
5. and : 17
6. kingdom : 11
```

- 7. "and * he * said" : 4
- 8. peace : 4
- 9. and she said : 1
- 10. devil : 1

4. Restriction

- ✓ genesis.txt뿐만 아니라 <http://www.stewartonbibleschool.org.uk/bible/text/>의 임의의 파일에 대해서도 동작하여야 한다.
- ✓ **Standard C library 이외의 library를 사용하지 않는다.**
 - string.h의 사용은 금지한다. 과제 0에서 만든 my_string.c와 my_string.h를 사용하며, 필요한 경우 함수를 추가해서 사용하도록 한다.
 - 파일의 입출력은 system call을 사용하도록 한다.
- ✓ 단어의 대, 소문자는 구별할 필요가 없으며, 영문 알파벳, - (하이픈), ' (어퍼스트로피) 를 제외한 문자는 단어에서 제외하여 인덱스가 추가하도록 한다.
 - 예를 들어, index로 관리되는 단어들은 다음과 같다.
 - god, and, adam, brother's, priests', kirjath-arba, sons'
 - And 와 and 는 같은 키워드이며, gOD와 god 역시 같은 키워드이다.
- ✓ 본 과제는 혼자서 한다.
- ✓ **Copy 할 경우, 연구실 자체 규정에 따라 처벌하며, 상당한 불이익이 있다.**

5. Hand in instructions

- ✓ 이번 과제에서는 각자 c 파일을 자유롭게 추가하거나 수정할 수 있다. 다만, 해당 압축 파일을 풀고, 'make'를 입력하면 실행 파일 2개(client, server)가 생성되어야 하며, 각 실행 파일은 각각 client와 server의 역할을 하면 된다. 즉, 각자의 상황에 맞게 Makefile을 자유로이 수정해도 된다. (indexBuilder, keywordSearcher의 argument를 수정해도 좋다)
- ✓ 작성한 프로그램 코드 상단에 이름과 학번을 적는다.
- ✓ 과제는 제출 시 "학번.tar.gz"로 압축한다.
- ✓ 이번 과제에 대한 보고서는 제출할 필요가 없으며, 추후 발표 자료로 이를 대신한다.
- ✓ 과제는 dylee@csl.skku.edu 로 보내고, 제목은 반드시 아래의 형식을 따른다.
 - [SSE2033] 2014123456 홍길동 PA6
 - 위 형식을 지키지 않아 생기는 불이익은 본인의 책임이다.
- ✓ 과제 제출 결과는 과제 페이지에서 확인할 수 있다.
 - <http://csl.skku.edu/SSE2033S16/Projects>
- ✓ 과제 제출 시간은 메일 도착 시간을 기준으로 하며, 늦을 경우 추가 규정에 따라 감점한다.