

Programming Assignment #2

Due : 2th May. (Mon), 11:59 PM

1. Introduction

이번 과제의 목표는 FILE I/O 및 Data structure 사용에 익숙해지도록 하는 것이다.

2. Problem specification

이번 과제는 PA 1에서 작성한 index builder에 추가적인 기능을 구현하고, index 파일을 이용하여 사용자의 키워드를 검색하여 장, 절, 위치를 출력해주는 "Keyword Searcher"을 구현하는 것이다.

2.1 Keyword Searcher

- ✓ Index Builder에서 만든 index 파일을 읽어 memory에 로드한다.
- ✓ 사용자가 키워드를 입력하면 해당하는 키워드가 나타나는 장, 절, 위치를 출력한다.
- ✓ 키워드 입력에 따른 출력은 아래의 규정을 따르도록 한다.

=====

1. 키워드 개수에는 제한이 없으며, 총 255자(byte)까지 입력할 수 있다.

2. 키워드가 1개인 경우, 장, 절, 위치를 출력한다.

예) > created

1:1:21

1:21:8

1:27:7

...

3. 키워드가 여러 개인 경우, 아래와 같이 출력한다.

(1) 키워드가 2개 이상인 경우 해당 키워드들이 **순서에 관계없이 모두 나타나는** 장과 절을 출력한다.

예) > created god

1:1

1:21

1:27

2:3

....

(2) 키워드가 큰 따옴표(“”)로 감싸져 있는 경우, 해당 **순서를 반드시 지키는** 장과 절을 출력한다. 즉, “god created”로 입력한 경우, god 다음 바로 created가 나오는 장, 절을

찾아서 출력한다.

예) > "god created"

1:1
1:21
1:27
2:3
...

- (3) 키워드가 큰 따옴표("")로 감싸져 있고, * 이 들어있을 경우, 하나 이상의 다른 단어가 들어있는 장과 절을 출력한다. 즉, "god * created"로 출력할 경우 한 절 안에서 god와 created 사이에 하나 이상의 다른 단어가 존재하는 장과 절을 출력한다.

예) god a created => 해당 장과 절 출력
god a b created => 해당 장과 절 출력
god created => 해당 장과 절 출력 X

- (4) *은 "god * created * the * heaven" 과 같이 사용할 수 있어야 한다.
- (5) *은 연속해서 나올 수는 없으며, 왼쪽 끝과 오른쪽 끝에 쓰지는 않는다.

4. PA2에 대해, PA1의 indexPrinter도 정상적으로 동작해야 한다. 즉, main에서 indexPrint를 호출한다면, PA1에서 명시한 형식대로 output 파일을 생성해야 한다.

- (1) Bonus를 구현한 경우, :add 하여 여러 개의 성서가 인덱스에 올라가 있는 경우에도 indexPrint를 통해 각각의 output파일을 생성해야 한다.

5. !q를 호출하면 프로그램은 정상 종료해야 한다.

BONUS) 여러 개의 성서를 인덱싱에 추가할 수 있는 :add 및 :delete 명령어를 추가한다.

- (1) 입력을 :add 성서명 으로 넣은 경우, 해당 성서를 인덱스에 추가하고, 앞으로 출력할 모든 명령어에 대해 추가한 성서도 출력하도록 한다.

예) > :add exodus
Done
> god created
genesis:1:1
genesis:1:21
exodus:3:21
...

- (2) :delete 성서명 으로 넣은 경우, 인덱스에서 해당 성서의 단어 인덱싱을 제외한다. 즉 delete 한 후 keyword search를 하거나 indexprint를 하면 해당 성서를 인덱스에서 제외한 결과를 출력한다.

=====

- ✓ 주어진 창세기 텍스트 파일을 이용하여 단어 별로 index를 만든다.
- ✓ 입력 파일은 <http://www.stewartonbibleschool.org.uk/bible/text/genesis.txt> 으로 한다.
- ✓ 입력 파일에서 장:절: 이 포함된 라인에 대해서만 index를 만들고, 장:절:이 포함되지 않은 라인은 indexing 및 검색에서 제외한다.
- ✓ 성서 이름은 입력 파일에서 확장자를 제외한 이름으로 한다.
 - (ex. genesis.txt => genesis)
- ✓ Index 마다 단어, 나타난 장/절, 횟수, 절 내에서의 위치를 관리한다.
- ✓ Index는 파일로 저장한다.

3. Restriction

- ✓ 과제는 리눅스 환경에서 수행하고, 과제 보고서에 컴파일 및 실행한 화면을 캡처하여 추가한다.
- ✓ genesis.txt뿐만 아니라 <http://www.stewartonbibleschool.org.uk/bible/text/>의 임의의 파일에 대해서도 동작하여야 한다.
- ✓ Database(NoSQL 등), Lex & Yacc는 사용하지 않는다.
- ✓ **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 open(), read(), write(), close(), lseek() 등의 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.

4. Hand in instructions

- ✓ 홈페이지에 업로드 된 skeleton 파일은 아래의 파일들로 이루어져 있다.

Makefile:	GNU make도구를 위해 필요한 파일
main.c:	
main.h:	
indexBuilder.c:	PA1에서 구현한 파일
indexPrinter.c:	PA1에서 구현한 파일

keywordSearcher.c: keyword searching을 위한 C 파일

- ✓ 작성한 프로그램 코드 상단에 이름과 학번을 적는다.
- ✓ 과제는 제출 시 "학번.tar.gz"로 압축한다.
 - 압축 파일은 Makefile, main.c, main.h, indexBuilder.c, indexPrinter.c, keywordSearcher.c, my_string.c, my_string.h, README.pdf로 이루어져 있어야 하며, 압축파일의 이름과 확장자는 학번.tar.gz 여야 한다.
- ✓ 프로그램 코드와 별도로, 구현에 대한 내용을 담은 보고서를 함께 제출한다. 보고서의 형식은 pdf로 제한하며, 형식에는 제한이 없다. 제목은 README.pdf 로 한다.
- ✓ 과제는 dylee@csl.skku.edu 로 보내고, 제목은 반드시 아래의 형식을 따른다.
 - [SSE2033] 2014123456 홍길동 PA2
 - 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.

6. Examples

```

$ ./keywordSearcher genesis.txt // keyword searcher 실행
** Index Builder : Start ** // genesis.txt를 입력 받아 genesis_index 파일 생성
    Elapsed Time : 33736(usec)
** Index Builder : End **

** Keyword Searcher : Start **
> created // 찾고자 하는 keyword 입력
1:1:21 // created가 나타나는 장, 절, 위치를 출력
1:21:8
1:27:7
1:27:57
1:27:89
2:3:113
2:4:73
...
> created god // 순서에 관계없이 created와 god이 나타나는 장, 절 출력
1:1
1:21
1:27
2:3
2:4
...
> "god created" // god 다음에 created가 나타나는 장, 절 출력
1:1
1:21
1:27
2:3
...
> "created * god" // created 다음에 하나 이상의 단어가 나타나고 god이 나타나는
1:21 // 장, 절 출력
1:27
2:4
...
> !q // !q 후에 enter를 입력하면 프로그램 종료
    Elapsed Time : 2671856(usec)
** Keyword Searcher : End **
$

```