

SSE2033: System Software Experiment2 (Spring 2017)

Programming Assignment #2

Due: April 17, 11:59 PM

1. Introduction

과제를 통해 File I/O 및 Data structure 사용에 익숙해지도록 한다.

2. Specification

2.1 Keyword Searcher

Index builder에서 생성된 index 파일을 읽어 들여 인덱스 된 내용을 정해진 형식에 따라 출력한다.

- 출력 파일의 첫 부분에는 성서 이름: 장 수, 총 절 수, 총 인덱스 수, 총 워드 수를 출력해야 한다.
 - 키워드가 한 개인 경우 장, 절, 위치를 출력한다.
 - 키워드가 여러 개인 경우 장, 절을 출력한다.
- 키워드는 여러 개(개수 제한없음) 입력할 수 있으며, 아래의 기능을 구현한다.
 - 검색어는 최대 255자까지 입력할 수 있다.
 - 사용자가 heaven를 입력하면 heaven이 들어간 장, 절, 위치를 출력한다.
 - 사용자가 heaven earth를 입력하면 하나의 절 안에서 heaven과 earth의 순서에 관계없이 둘 다 있는 장, 절을 출력한다.
 - 사용자가 "heaven earth"를 입력하면 하나의 절 안에서 heaven과 earth가 둘 다 있으며 heaven 다음에는 바로 earth가 나오는 장, 절을 출력한다.
 - **Bonus** - 사용자가 "heaven * earth"를 입력하면 하나의 절 안에서 heaven과 earth 사이에 하나 이상의 다른 단어가 존재하는 장, 절을 출력한다.
'*'를 사용할 때에는 반드시 왼쪽과 오른쪽에는 키워드가 나와야 한다. 또한 "heaven * earth * to"와 같이 키워드는 여러 개 사용이 가능하다.

3. Restriction

과제는 본인이 직접 설치한 리눅스 환경에서 수행하고, 실행한 화면을 캡처하여 추가한다.

예제 코드에 include된 것 이외의 library는 사용하지 않는다.

파일의 입출력은 open(), close(), read(), write(), lseek() 등의 system call 중 필요한 것을 선택하여 사용해야 한다.

단어의 대, 소문자는 구별할 필요가 없으며 영문 알파벳, - (하이픈), ' (어퍼스트로피)를 제외한 문자는 단어에서 제외하여 인덱스에 추가한다.

예를 들어, index로 관리되는 단어들은 다음과 같다.

god, and, adam, brother's, priests', kirjath-arba, sons'

genesis.txt 뿐만 아니라 <http://www.stewartonbibleschool.org.uk/bible/text/>에 있는 임의의 파일에 대해서도 동작해야 한다.

4. Hand in instructions

과제 및 스크린샷은 제출시 "학번.tar.gz" 파일로 압축하여 제출한다. (예: 2012345678.tar.gz)

과제는 기본적으로 "sys.skku.edu"에 제출하고,

기한이 지나 제출이 안 된다면 sanghoon.han@csl.skku.edu로 아래와 같은 형식을 맞춰 보낸다.

[SSE2033] PA2, 학번, 이름

5. Logistics

과제 제출 결과는 <http://csl.skku.edu/SSE2033S17> 에서 확인할 수 있다.

과제 제출 시간은 메일 도착시간을 기준으로 하며, 기한 이후엔 10%씩, 최대 60%까지 감점 될 수 있다.

과제를 compile할 때, "-Wall -W" 옵션 기준으로 warning이나 error가 전혀 없어야 한다. Warning 하나 당 1%씩, 최대 10%까지 감점될 수 있다.

과제에 대한 의논은 함께 할 수 있으나, 프로그램 소스코드 작성은 스스로 해야 한다.

다른 사람의 과제를 copy 한 경우, 두 사람 모두 0점 처리한다. 인터넷 등에서 찾은 소스 코드를 그대로 copy 한 경우에도 0점 처리한다. 두 번 이상 이와 같은 이유로 0점 처리된 경우 F 학점을 받을 수 있다.

6. Example

```
$ ./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 **
$
```