Programming Assignment#1

Due : 11th Apr. (Mon), 11:59 PM

1. Introduction

이번 과제의 목표는 과제를 통해 FILE I/O 및 Data structure 사용에 익숙해지도록 한다.

2. Problem specification

Search engine에서 indexing이라는 기법은 사용자가 입력한 키워드를 이용하여 매우 크고 많은 문서에서 해당 키워드가 나타나는 위치를 재빠르게 찾을 수 있도록 하는 기술이다. 본 과제에서는 주어진 Holy Bible의 창세기(genesis)파일의 index를 만들고, 이 index를 저장하는 "Index Builder"와, 생성된 index를 이용하여 출력하는 "Index Printer"를 만드는 것이다. 향후 과제들은 본 과제를 바탕으로 이루어지며, 이번 과제에서 만든 Index builder를 업그레이드 할 예정이다.

2.1 Index Builder

- ✓ 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)
- ✓ Indexing 하는 실행 시간, 탐색 시간, 파일 크기 등을 고려하여 자료 구조를 설계한다.
  - ➢ BONUS : Hash Table을 이용하여 관리한다.
- ✓ Index 마다 단어, 나타난 장/절, 횟수, 절 내에서의 위치를 관리한다.
- ✓ Index는 파일로 저장한다.

2.2 Index Printer

- ✓ Index builder에서 생성된 index 파일을 읽어 indexing 된 내용을 정해진 형식에 따라 출력한다.
- ✓ 출력 파일의 첫 부분에는 성서 이름: 장 수, 총 절 수, 총 인덱스 수, 총 단어 수를

출력한다.

- ✓ 이 후에는, 매 index 마다 다음 형식으로 출력한다.
  - ➢ 단어: 총 출현 횟수, 장:절:위치, 장:절:위치, ...
- ✓ 단어를 sorting하여 출력할 필요는 없다.

3. Restriction

- ✓ 과제는 리눅스 환경에서 수행하고, 과제 보고서에 컴파일 및 실행한 화면을 캡쳐하여 추가한다.
- ✓ genesis.txt뿐만 아니라 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'
- ✓ Index를 저장할 파일은 자유롭게 정하는 대신, 자신만의 규칙을 반드시 보고서에 명시하여야 한다.

4. Hand in instructions

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

`Makefile:`	GNU `make`도구를 위해 필요한 파일
`main.c:`	시간 측정 및 `indexing, printing` 해주는 파일
`indexBuilder.c:`	`index build`를 위한 `C` 파일
`indexPrinter.c:`	`index print`를 위한 `C` 파일

- ✓ 작성한 프로그램 코드 상단에 이름과 학번을 적는다.
- ✓ 과제는 제출 시 "학번.tar.gz"로 압축한다.
  - ➢ 압축 파일은 Makefile, main.c, indexBuilder.c, indexPrinter.c, 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 PA1
  - ➢ 위 형식을 지키지 않으면 불이익(스팸함)이 생길 수 있다.
- ✓ 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.
- ✓ 본 과제는 혼자서 한다.
- ✓ GNU `make` 도구는 큰 프로그램을 만들 때 유용하게 사용되는데, 프로그램을 제작하기 위해 어떤 코드를 (재)컴파일해야 하는지 결정해준다. 일단 `Makefile`이 준비되면, 어떤 소스 코드를 변경하던지 셸에서 단순히 `make`란 명령을 실행시키는 것으로 재 컴파일이 필요한 모든 파일을 알아서 찾아 다시 컴파일한다.
- ✓ **Copy 할 경우, 연구실 자체 규정에 따라 처벌하며, 상당한 불이익이 있을 수 있다.**

5. Examples

```
$ ./indexBP genesis.txt                    // index builder & printer 실행
** Index Builder : Start          **
   Elapsed Time   : 62558(usec)            // index builder 수행시간이 usec 단위로 나타남
** Index Builder : End            **

** Index Printer : Start          **      // indexBP를 실행하면 index file인 genesis_index와 output file인
** Index Printer : End            **      // genesis_ouput이 생성됨
$ ./cat genesis_output
genesis: 50 1533 2510 38260                // 성서: genesis, 장수: 50, 절수: 1533, Index수: 2510, 단어수: 38260
leaves: 1, 3:7:94                          // 단어: 총 개수, 장:절:위치
ephraim's: 3, 48:14:58, 48:17:153, 50:23:15  // 단어: 총 개수, 장:절:위치, 장:절:위치, 장:절:위치
hadar: 2, 25:15:0, 36:39:43                // 위치는 절 내에서 단어가 나타나는 위치로, Byte 단위로 한다.
depart: 2, 13:9:149, 49:10:22
parted: 1, 2:10:73
...
$
```

Have fun!

이동윤, 담당 조교

컴퓨터시스템연구실