

Programming Assignment #0

Due : 29th Sep. (Sun), 11:59 PM

1. Introduction

텍스트 파일을 읽어 단어 별 출현 횟수를 세고 정렬한다.

본 과제를 통해 기본적 File I/O에 익숙해진다.

2. Problem specification

텍스트로부터 읽은 단어들을 원하는 구조(search tree, hash table etc.)를 이용해 저장한다.

단어의 대소문자는 구분되며 대문자가 더 높은 정렬 순위를 갖는다.

단어 입력이 끝나면 입력 받은 단어들을 출현 횟수와 함께 오름차 순으로 터미널에 출력한다.

홈페이지에 업로드 된 skeleton 파일은 아래의 파일들로 이루어져 있다.

- Makefile: GNU make도구를 위해 필요한 파일
- main.c: 텍스트를 단어별로 파싱하고, 정렬, 출현 횟수를 세는 파일

과제는 main.c에 구현한다.

3. Restriction

단어는 연속된 영문알파벳으로 정의되며, 영문 알파벳 '사이'의 -(하이픈)과 `(어퍼스트로피) 두 기호만 단어 구성으로 인정된다.

INPUT TXT	WORD
~ wake--up ~	wake--up
~ --apple- ~	apple
~ man.in.the.mirror ~	man / in / the / mirror

파일의 입출력은 open(), read(), write(), close(), lseek() 등의 system call을 사용하도록 한다. skeleton code에 주어진 것 외의 추가적 library는 사용하지 않는다.

4. Example

```
$ cat sample.txt
Row, row, row your boat
Gently down the stream
Merrily merrily, merrily, merrily
Life is but a dream
```

```
$ parser < sample.txt
a 1
boat 1
but 1
down 1
dream 1
Gently 1
is 1
Life 1
Merrily 1
```

	merrily 3 Row 1 row 2 stream 1 the 1 your 1
--	--

5. Hand in instructions

- 본 과제는 개인 과제이며 icampus의 과제란에 제출한다.
- 과제 제출 시간은 icampus 제출 시각 기준으로 하며, 매 24시간마다 10%씩 감점된다.
- 제출시 프로그램코드와 보고서를 함께 압축하여 제출한다.
 - ✓ 압축 파일은 Makefile, main.c, README.pdf로 이루어져 있어야 하며, 압축 파일의 이름과 확장자는 "학번.tar.gz"여야 한다.
 - ✓ 과제 보고서에는 자신의 사용한 저장 구조에 관한 간단한 설명과 컴파일 및 실행한 화면을 캡처하여 함께 제출한다. 보고서 포맷은 pdf, 이름은 README.pdf로 하며 형식에 제한은 없다.
 - ✓ 작성한 프로그램 코드(main.c) 상단에 이름과 학번을 적는다.
- Copy 할 경우, 연구실 자체 규정에 따라 처벌하며, 상당한 불이익이 있을 수 있다.

6. Additional

GNU make 도구는 큰 프로그램을 만들 때 유용하게 사용된다. 이는 프로그램을 제작하기 위해 어떤 코드를 (재)컴파일해야 하는지 결정해준다. Makefile이 준비되면, 어떤 소스 코드를 변경하던지 셸에서 단순히 make란 명령을 실행시키는 것으로 재 컴파일이 필요한 모든 파일을 알아서 찾아 다시 컴파일한다.