

SWE2033: System Programming Experiment 2 (Fall 2020)

Programming Assignment #1

Due: October 9th, 11:59 PM

1. Introduction

과제를 통해 자료구조에 대한 이해와 File I/O 사용에 익숙해진다.

2. Specification

이 과제의 목표는, shell command로 입력된 영화 시나리오 파일을 읽고 특정 검색 기능을 수행하는 코드를 만들어 보는 것이다. 프로그램이 시작된 후, 프로그램은 사용자의 키워드 입력을 기다리며 아래에 언급된 검색 기능을 수행한다.

① Searching single word locations

프로그램에서 하나의 단어를 입력 받는 경우, 단어의 위치를 찾는다.

- 영화 시나리오에서 입력 받은 단어가 포함된 행을 찾아 다음의 형식으로 stdout에 출력한다.

"[line number]:[start index of the word]"

② Searching several words locations

프로그램이 여러 단어(공백 하나로 구분)를 입력 받는 경우, 모든 단어를 포함하는 행을 찾는다.

- 영화 시나리오에서 입력 받은 단어가 모두 포함된 행을 찾아 stdout에 출력한다.

"[line number]"

③ Searching several consecutive words locations

프로그램에서 ""로 싸여진 구문을 입력 받는 경우, 구문의 위치를 찾는다.

- 영화 시나리오에서 입력 받은 구문이 포함된 행을 찾아 다음의 형식으로 stdout에 출력한다.

"[line number]:[start index of the phrase]"

④ Searching simple regular expressing keyword locations

프로그램에서 [keyword1]*[keyword2]로 구성된 "두개의 키워드"를 입력 받는 경우, [keyword1]과 [keyword2] 사이에 하나 이상의 단어(빈칸 제외)가 포함된 행을 찾는다. ([keyword1]과 *, *과 [keyword2] 사이에는 공백이 없다.)

- 영화 시나리오에서 위에서 설명한 행을 찾아 stdout에 출력한다.

"[line number]"

⑤ Supplementary explanations

- line number와 start index of the word (or phrase)에 대한 설명은 다음과 같다.

*line number – 단어(혹은 구문이나 키워드)를 포함하는 행의 번호이며, 값은 1부터 시작한다. 빈 행 또한 개수를 세어야 한다.

*start index of the word (or phrase) – 행에 있는 단어(혹은 구문)의 시작 번호이며, 값은 0부터 시작한다. 빈 칸과 writing symbol 또한 세어야 한다.

- 입력 받은 단어가 영화 시나리오에 여러 번 나타나면, 해당하는 행을 모두 출력해야 한다.

- ①, ③의 경우, 같은 행에 존재하더라도 모든 위치를 출력해야 한다.
- ②, ④의 경우, 중복된 행에 대해서 한 번만 출력하면 된다.

- 하나의 검색 결과 위치 뒤에 빈칸이 추가되어야 한다. 그리고 검색이 끝나면 새 줄을 출력해야 한다. 예를 들면 다음과 같다.

15:23 17:10 23:4

15 7 23

- 입력에서 " 또는 * 문자에 대해서는 다음과 같이 분류한다.

- 입력으로는 "와 *가 동시에 포함되지 않는다.

- 입력에 " 문자가 포함되어 있을 경우, ③에 해당한다.
- 입력에 * 문자가 포함되어 있을 경우, ④에 해당한다.
- 입력이 " 와 * 문자가 모두 포함되어 있지 않은 2개 이상의 단어로 구성된 경우, ②에 해당한다.
- 입력이 " 와 * 문자가 모두 포함되어 있지 않은 1개의 단어로 구성된 경우, ①에 해당한다.
- 검색 결과는 파일의 상단에서 순차적으로 출력되어야 한다.
- 단어 입력에 대한 세부사항은 "6. Example" 참조

3. Score Policy

- 100점 만점을 기준으로 하며 위에서 설명한 형식을 따라야 하며, 그렇지 않으면 점수를 받을 수 없다.
- 마감 이후 제출은 하루 단위로 10점씩 감점이 적용된다. (10일이 지난 후, 제출 시 0점)
- 어느 정도의 discussion은 허용하지만, 소스코드는 스스로 작성해야 한다.

4. Restriction

- 이 과제는 리눅스 환경에서 하는 것을 기준으로 한다.
- * **stdio.h, string.h 라이브러리를 사용할 수 없다. (사용 시 0점)**
- 단어: 빈칸으로 구분되어지는 문장이며(탭, 스페이스, 개행), 대소문자를 구분하지 않는다.
- 단어 검색에 사용될 수 있는 단어의 예시는 다음과 같다.

e.g.) god, and, adam, brother's, priests', kirjath-arba, sons'

- Writing symbol을 포함하는 단어들은 서로 다른 단어들로 간주된다. 예를 들어, he를 검색하면 her, he's가 아닌 오직 he만 검색 결과에 포함해야 한다.

5. Hand in instructions

- * 과제 제출 시, "Student_ID.tar.gz"으로 압축하여 i-Campus에 제출한다. 그리고 압축파일 안에는, 디렉토리 가 아닌 "*.c, *.h, makefile"만 존재해야 한다.

6. Example (빨간색으로 강조 표시된 단어가 입력으로 주어짐)

```
$ ./assignment1.out 500-Days-of-Summer_s.txt
```

500 days

6 6419

he is

50 1039 1822 1955 2256 2315 3494 3503 4353 4360 4445 4831 5101 5885 6325

“he is”

1955:33 2315:42 5885:22

he*is

2315 4445 5101

she he

1370 1512 1513 3423 3473 3478 3550 4255 4510 4515 5413 5672 6154 6188

loved

106:30 1122:9 1150:24 3961:25 4739:17 5921:28

...

```
$ ./assignment1.out 500-Days-of-Summer_s.txt > result.out
```

he

hey

she

tom

summer is

"summer is"

landscape

together

we*her

no much

immediately

no*much

quarterback

we

(ctrl + c)

```
$ diff -bsq result.out answer.out
```

Files result.out and answer.out are identical