

REPORT

보고서 작성 서약서

1. 나는 타학생의 보고서를 복사(Copy)하지 않았습니다.
2. 나는 타학생의 보고서를 인터넷에서 다운로드 하여 대체하지 않았습니다.
3. 나는 타인에게 보고서 제출 전에 보고서를 보여주지 않았습니다.
4. 보고서 제출 기한을 준수하였습니다.

나는 보고서 작성시 위법 행위를 하지 않고,
성.균.인으로서 나의 명예를 지킬 것을 약속합니다.

과 목 : 전자전기 프로그래밍실습

과 제 명 : HW2

담당교수 : 민 형 복

학 과 : 전자전기공학부

학 년 : 1

학 번 : 2017311583

이 름 : 정해진

제 출 일 : 2018.4.2

1. Introduction

Text file을 read하고 write하는 방법을 배운다. Text file을 read할 때 read하려는 파일에 문제가 있는 경우, 해당하는 문제를 출력 파일과 console창에 출력함으로써 오류 보고도 수행한다.

2. Problem Statement

① Describe what is the problem.

- Main 함수 전에 선언한 "computeSize" 함수와 "readFile" 함수를 모델링하여 원활한 Main 함수의 실행을 꾀한다.

② Describe how do you solve the problem.

Text file "infile.txt"이 1개 주어진다. 이 파일에 여러 학생의 이름, 나이, 성별(M or F), 여러 과목의 성적 점수가 들어있다.

- computeSize 함수

text file의 크기를 측정하여 Console 화면에 출력한다. [1]에서 "ftell()" 함수와 "fseek()" 함수에 관한 내용을 참고하여 프로그램을 작성한다.

- readFile 함수

- text 파일의 첫번째 줄에는 학생의 숫자와 과목의 숫자가 있으며, 2번째 줄 이후에는 학생의 정보(이름, 나이, 성별, 여러 과목의 점수)가 1줄에 1명 분량만큼 들어있다.

- 이 text file을 읽어서, 모든 학생의 N과목 점수의 평균을 double floating point -number로 계산하여 소수점 둘째자리까지 console에 출력한다.

- M명을 성별로 숫자를 console에 출력한다.

- M명의 나이의 평균을 소수점 첫째자리까지 console에 출력한다.
- Console에 출력한 것과 똑같은 것을 "outfile.txt"에 동시에 출력한다.

Textfile을 읽는 과정에서 text 파일의 첫번째 줄 내용과 그 이후 내용을 구분해서 각각 다른 수치로써 저장할 필요가 있다. sscanf()를 적절히 활용하여 각각의 줄에 따라 필요한 정보들을 가져와야 한다.

- "/"로 시작하는 모든 줄은 comment이므로 건너뛴다. file read를 수행하지 않는다.

sscanf()를 이용하여 comment 줄을 파악해야 한다.

- 정보는 누락된 것이 있을 수 있다. 누락된 경우에는 적절한 error message를 준 후, 이 줄은 file read 과정에서 무시한다.

sscanf()의 return 값을 이용하여 오류 여부를 파악해야 한다.

Error 종류는 다음과 같다.

1. 특정 line의 이름, 나이, 성별 중 하나가 누락된 경우

-> 특정 line의 이름, 나이, 성별 중 하나가 누락되었음을 출력한다.

2. 특정 line의 성별이 잘못된 형태인 경우("M"이나 "F"가 아닌 경우)

-> 특정 line의 성별이 "infile.txt" 파일에 입력된 character임을 출력한다.

3. 특정 line의 과목 점수가 누락된 경우(과목 수와 과목 점수의 개수가 일치하지 않음)

-> 특정 line의 몇 번째 과목의 점수가 누락되었음을 출력한다.

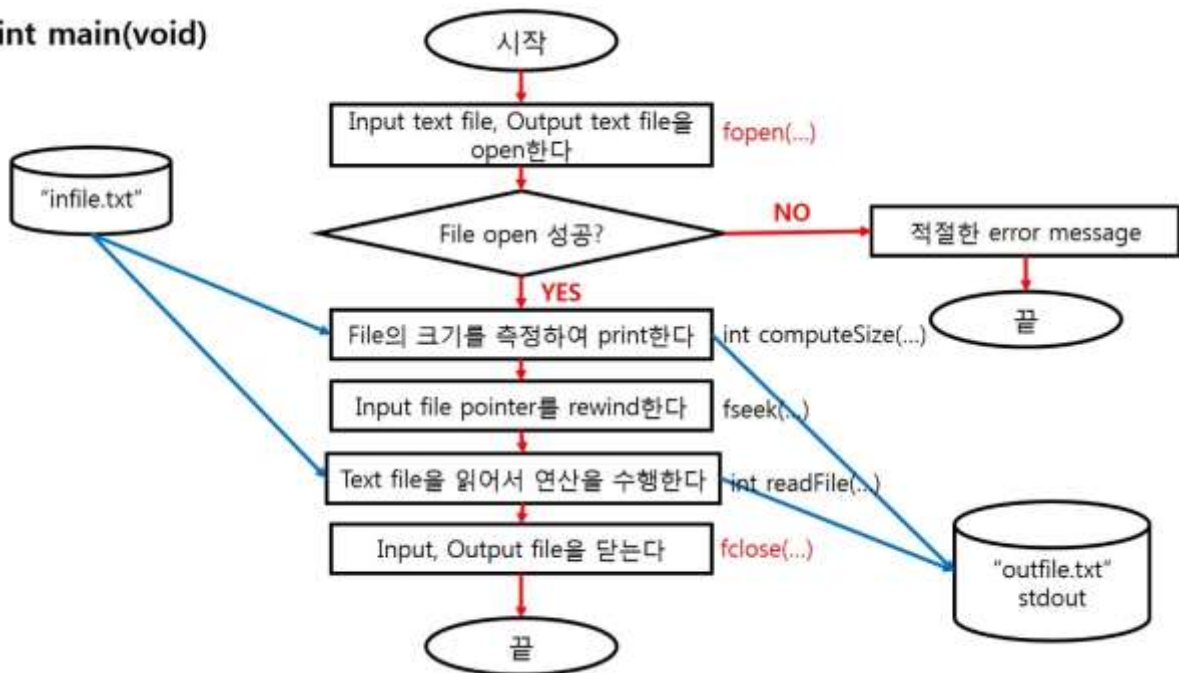
4. 위 1,2,3번 오류가 한 번 이상 난 경우

-> 정상 line 개수와 자료가 담겨있을 것으로 예상되는 line 개수를 출력한다.

③ Draw a flowchart of your algorithm

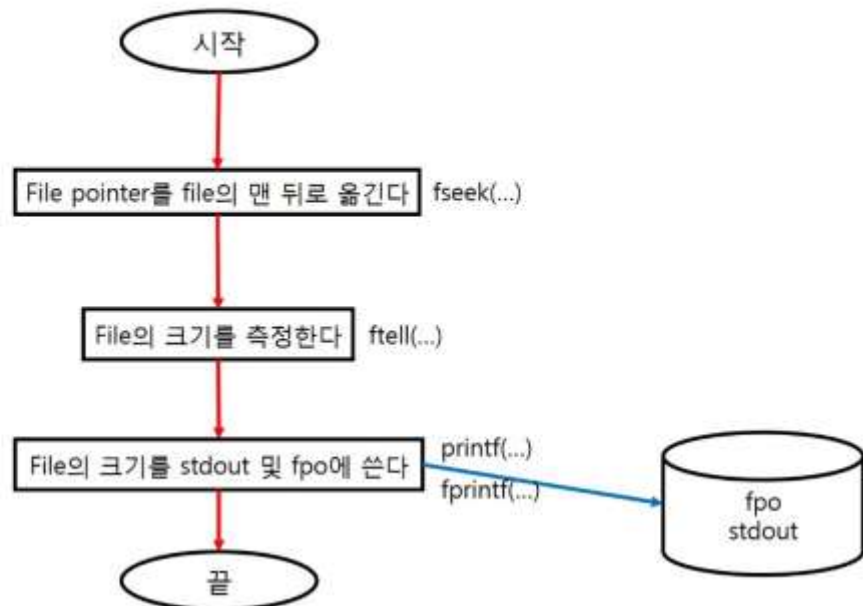
1. Main함수

int main(void)

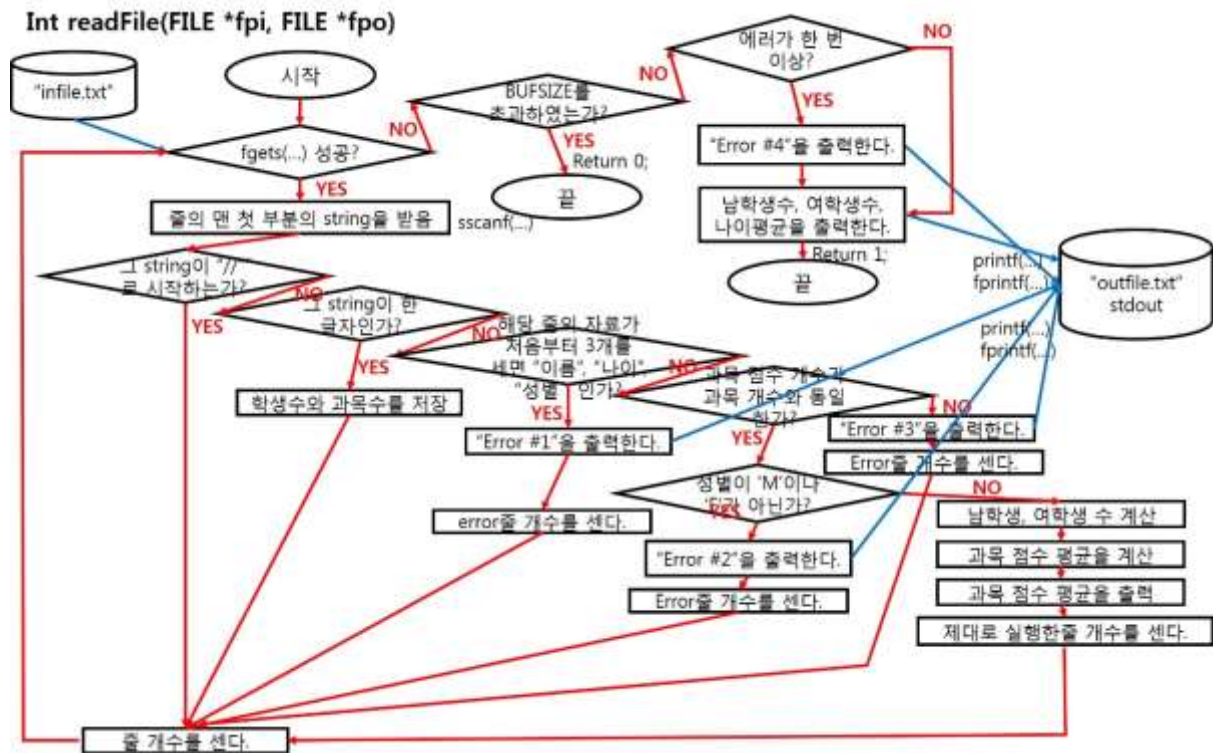


2. computeSize함수

Int computeSize(FILE *fpi, FILE *fpo)



3. readFile함수



3. Implementation

1. Main 함수

FILE 자료형 포인터 변수 "fpi"가 character 자료형 포인터 변수 "input_file"을 가리키고, "input_file"은 파일 이름 string을 가리킨다. 그러므로 fopen 함수가 원활히 구동될 수 있다. 비슷한 방식으로 FILE 자료형 포인터 변수 "fpo"도 마찬가지다. "fpi"와 "fpo"가 NULL 값을 반환할 때, 즉 "infile.txt"와 "outfile.txt"가 원활히 open되지 않았을 때 에러 메시지를 출력한다.

이후로 "computeSize" 함수, "fseek" 함수, "readFile" 함수가 각각 NULL을 반환할 때마다 PRINTF 함수를 통해 적당한 에러 메시지를 출력하고, CLOSE2 함수를 통해 파일을 닫는다. 그리고 Main 함수는 각기 다른 값을 return한다.

"computeSize" 함수, "fseek" 함수, "readFile" 함수가 모두 제대로 실행된 경우에는 CLOSE2 함수를 통해 파일을 닫고 Main 함수는 0을 return한다.

2. computeSize 함수

"infile.txt"를 읽을 때 쓰이는 포인터를 "fseek" 함수를 통해 "infile.txt"의 맨 끝으로 이동시킨다. 그리고 long 자료형의 변수 "file_size"에 "ftell" 함수의 반환값을 저장한다. "file_size"이 0인 경우 함수는 0을 리턴하고, 아닌 경우에는 "infile.txt" 파일의 크기를 console 창과 "outfile.txt"에 출력한다.

3. readFile 함수

"infile.txt"를 읽을 때 한 줄씩 읽기 위하여 "fgets" 함수의 return 값에 따라 반복 유무가 결정되는 while구문을 사용한다. 안전한 파일 읽기를 위해 "fgets" 함수를 사용하였으며, 함수에 필요한 parameter 들로 character 포인터 자료형의 "read_line_buf"와 int 자료형의 1024의 값을 가지는 BUFSIZE를 선언하였다. "read_line_buf"의 동적 메모리 할당 방식을 사용했다. 이 과정에서 "infile.txt"의 특정 줄에서 buffer overflow가 일어난 경우 readFile 함수는 0을 return 한다. int 자료형의 age, score[4]와 character 자료형의 gender와 name[10]은 "fgets" 함수를 통해 읽은 한 줄을 sscanf 함수를 통해 분리한 데이터 값을 저장하기 위한 변수들이다.

int 자료형의 lineNums, validLNums, invalidLNums는 "infile.txt"에서 특정 데이터값에 이상이 있을 때 출력하는 error 문장에서 쓰이는 변수들이다. 각각의 error 문장들은 2-② 항목의 error 종류들 중 하나를 따른다. 그리고 이 문장들은

console 창과 "outfile.txt"에 동시에 출력된다. lineNums는 "infile.txt"의 몇 번째 줄인지를 세는 변수, validLNums는 아무 이상이 없는 줄의 수를 세는 변수, invalidLNums는 이상이 있는 줄의 수를 세는 변수이다.

int 자료형의 sumGrades는 성공적으로 읽은 특정 학생의 데이터들 중 그 학생의 과목 점수들의 총합을 저장하기 위한 변수이다. 그리고 이 값을 이용해 double 자료형의 변수 aveGrade에 과목 평균을 계산하여 저장하고, 이를 console 창과 "outfile.txt"에 출력한다.

int 자료형의 studNums와 subjNums는 각각 "infile.txt"에서 확인 가능한 총 학생 수와 총 과목 수를 각각 저장하기 위한 변수들이다. int 자료형의 maleNums와 femaleNums는 "infile.txt"에서 확인 가능한 총 남학생 수와 총 여학생 수를 각각 저장하기 위한 변수들이다.

int 자료형의 sumAges는 성공적으로 읽은 특정 학생의 데이터들 중 그 학생의 나이를 더하며 저장하기 위한 변수이다. 그리고 이 값을 이용해 double 자료형의 변수 averAges에 모든 학생들의 나이 평균을 계산하여 저장하고, 이를 console 창과 "outfile.txt"에 출력하며 함수를 종료한다. 성공적으로 마치면 readFile 함수는 1을 return 한다.

4. Result

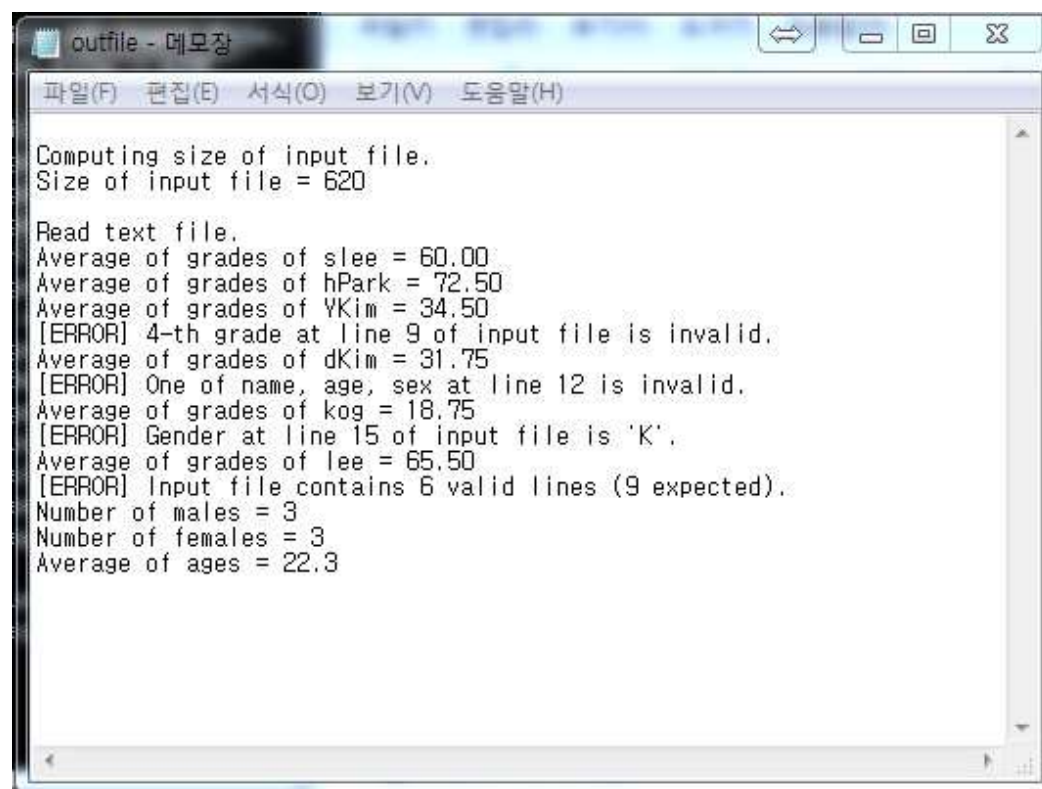
"2017311583.JungHaeJin.HW2.c" 를 컴파일 후 실행한 결과이다.

위에서부터 차례대로 console창, outfile.txt의 모습이다.

```
Computing size of input file.
Size of input file = 620

Read text file.
Average of grades of slee = 60.00
Average of grades of hPark = 72.50
Average of grades of YKim = 34.50
[ERROR] 4-th grade at line 9 of input file is invalid.
Average of grades of dKim = 31.75
[ERROR] One of name, age, sex at line 12 is invalid.
Average of grades of kog = 18.75
[ERROR] Gender at line 15 of input file is 'K'.
Average of grades of lee = 65.50
[ERROR] Input file contains 6 valid lines (9 expected).
Number of males = 3
Number of females = 3
Average of ages = 22.3

Process returned 0 (0x0)   execution time : 0.014 s
Press any key to continue.
```



```
Computing size of input file.
Size of input file = 620

Read text file.
Average of grades of slee = 60.00
Average of grades of hPark = 72.50
Average of grades of YKim = 34.50
[ERROR] 4-th grade at line 9 of input file is invalid.
Average of grades of dKim = 31.75
[ERROR] One of name, age, sex at line 12 is invalid.
Average of grades of kog = 18.75
[ERROR] Gender at line 15 of input file is 'K'.
Average of grades of lee = 65.50
[ERROR] Input file contains 6 valid lines (9 expected).
Number of males = 3
Number of females = 3
Average of ages = 22.3
```

파일을 read하는 반복문이 원활하게 실행되었다. 그리고 "infile.txt"에서 문제가 있는 자료에 대해 error 메시지를 console 창과 "outfile.txt"에 출력이 잘 되었다.

5. Conclusion & Evaluation

실습을 통해 text file을 다루는 과정을 좀 더 이해하게 된 것 같다. ftell과 fseek 함수가 생소해서 c파일을 작성하는데 어려움이 있었지만, [1]과 관련한 강의 내용을 떠올리며 찬찬히 작성해보니 잘 되었다. 앞으로 이 실습을 진행하는 학생들은 [1]에서 많은 힌트를 얻을 수 있을 것이다.

6. 참고 문헌

[1] Min, H. B. and SKKU, “textfile.pdf”