

자료구조2 실습 레포트



과목명		자료구조2 실습
담당교수		홍 민 교수님
학과		컴퓨터소프트웨어공학과
학년		2학년
학번		20204059
이름		이예빈

목 차

1. 파일 입출력과 구조체, 동적 할당을 이용한 학생 성적 프로그램

1.1 문제 분석

1.2 소스 코드

1.3 소스 코드 분석

1.4 실행창

1.5 느낀점

파일 입출력과 구조체, 동적 할당을 이용한 학생 성적 프로그램



구조체 & 포인터

■ 파일에서 자료 읽어오기

- data.txt에 학생 이름, 수학 점수, 영어 점수, 국어 점수가 저장되어 있다. 이 정보를 이용하여 아래 조건에 맞게 출력하는 프로그램을 작성하라.
 - 파일 data.txt에 저장되어 있는 정보를 사용할 것
 - 읽어온 정보를 Student 구조체를 이용하여 저장
 - 동적 할당 이용
 - 저장된 정보들을 화면에 출력
 - 각 과목 점수의 총합과 평균을 출력

C:\WINDOWS\system32\cmd.exe						
학 번	이름	수학	영어	국어	총 점	평균
20170245	김지연	40	40	100	180	60.00
20170285	김준섭	60	20	85	165	55.00
20170488	전희원	80	40	20	140	46.66

계속하려면 아무 키나 누르십시오 . . .

data.txt						
파일(F)	편집(E)	서식(O)	보기(V)	도움말(H)		
20170245	김지연	40	40	100		
20170285	김준섭	60	20	85		
20170488	전희원	80	40	20		

1. 문제 분석

이 문제는 순서대로 각 학생의 학번, 이름, 수학/영어/국어 점수가 한 줄 씩 저장되어 있는 파일을 읽어 동적으로 할당 받은 구조체에 저장하고, 저장된 정보들을 각 학생의 성적에 대한 총점과 평균과 함께 화면에 출력하는 프로그램이다. 구조체, 파일 입출력 함수, 동적으로 구조체 배열에 대한 메모리 할당, 메모리 사용과 해제를 적절히 이용해야 하는 문제이다.

먼저, 학생의 학번, 이름, 성적, 그리고 총점과 평균까지 함께 저장할 학생 구조체를 선언해야 하고, 파일을 열고 파일의 정보들을 읽기 위한 파일 입출력 함수들이 필요하다. 동적 할당을 하기 위해서는 구조체 배열 요소의 수(학생 수)를 알아내야 한다. 따라서 파일을 처음 읽을 때에는 학생의 정보를 임시 구조체 변수에 저장하여 먼저 학생 수가 몇 명인지 세도록 한다. 이 때, while 반복문을 사용하여 파일이 끝에 도달할 때까지의 조건을 사용하여 학생 수를 구하도록 한다. 다시 파일의 처음으로 돌아가 파일로부터 각 학생의 정보들을 읽으며, 학생 수만큼 동적 할당 받은 구조체 배열에 정보들을 저장하도록 한다. 이 때는 학생의 수만큼 for 반복문을 실행하여 정보들을 저장한다. 저장함과 동시에, 수학, 영어, 국어 점수를 더하고 나눈 값을 각각 총점과 평균에 저장하고, 형식에 맞게 화면에 출력하도록 한다. 모든 작업이 끝난 후에는 동적 할당 받은 메모리를 해제하고, 파일을 닫는다.

1.2 소스 코드

```

1  /*
2  →  작성자: ·이예빈 (20204059)
3  →  작성일: ·2021.09.06
4  →  프로그램명: ·파일·입출력과·구조체의·동적·메모리·할당을·이용한·학생·성적·프로그램
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 //·학생·구조체
11 typedef struct student {
12 → //·학번, ·이름, ·수학/영어/국어·점수, ·총점, ·평균
13 → int id;
14 → char name[10];
15 → int math, eng, kor;
16 → int sum;
17 → double avg;
18 } STUD;
19
20 int main() {
21 → FILE* fp = NULL;
22 → int n = 0, i; // n: 학생 수, i: 반복문 제어 변수
23 → STUD tmp; // 파일로부터 읽은 학생 정보를 임시로 저장해 놓을 학생 구조체
24 → STUD* studList = NULL; // 학생 정보를 담을 메모리를 동적 할당받기 위한 포인터
25
26 → // 파일 열기.. (실패 코드 포함)
27 → fp = fopen("data.txt", "rt");
28 → if (fp == NULL) {
29 → → fprintf(stderr, "파일 열기 실패");
30 → → exit(1);
31 → }
32
33 → // 학생 수 n 구하기
34 → while (!feof(fp)) {
35 → → // 파일에서 읽어 tmp에 임시로 저장
36 → → fscanf(fp, "%d %s %d %d %d", &tmp.id, tmp.name, &tmp.math, &tmp.eng, &tmp.kor);
37 → → n++;
38 → }
39 → rewind(fp); // 다 읽은 후에는, 포인터를 다시 파일의 처음으로 되돌리기
40
41 → // 학생 수만큼 동적 할당
42 → studList = malloc(sizeof(STUD) * n);
43
44 → printf("=====|\n");
45 → printf("| 학...번 | 이름 | 수학 | 영어 | 국어 | 총 점 | 평 균 |\n");
46 → printf("=====|\n");
47
48 → // 학생들 구조체에 정보 저장 후, 출력
49 → for (i = 0; i < n; i++) {
50 → → fscanf(fp, "%d %s %d %d %d", &studList[i].id, studList[i].name,
51 → → &studList[i].math, &studList[i].eng, &studList[i].kor);
52 → → studList[i].sum = studList[i].math + studList[i].eng + studList[i].kor;
53 → → studList[i].avg = studList[i].sum / 3.0;
54
55 → → printf("| %d | %s | %3d | %3d | %3d | %3d | %.2f |\n", studList[i].id, studList[i].name,
56 → → studList[i].math, studList[i].eng, studList[i].kor, studList[i].sum, studList[i].avg);
57 → }
58 → printf("=====|\n");
59
60 → free(studList); // 메모리 해제
61 → fclose(fp); // 파일 닫기
62 }

```

1.3 소스 코드 분석

1. 작성자, 작성일, 프로그램명을 주석으로 작성하고, 표준 입출력 헤더 파일, 그리고 동적 메모리를 관리하는 표준 라이브러리 헤더파일을 추가한다. 이후 학생의 학번(정수형), 이름(최대 10 문자를 저장하는 char 배열), 수학/영어/국어 점수(정수형), 그리고 총점(정수형)과 평균(double 실수형)을 멤버로 하는 학생 구조체를 선언한다. typedef 를 사용해 STUD 라는 이름의 구조체로 선언하였다.

```

/*
  > 작성자: .이예빈 (20204059)
  > 작성일: .2021.09.06
  > 프로그램명: .파일.입출력과.구조체의.동적.메모리.할당을.이용한.학생.성적.프로그램
*/

#include<stdio.h>
#include<stdlib.h>

//.학생.구조체
typedef struct student{
  > //.학번, .이름, .수학/영어/국어.점수, .총점, .평균
  > int id;
  > char name[10];
  > int math, eng, kor;
  > int sum;
  > double avg;
}STUD;

```

2. main 함수의 시작 부분이다. 파일 포인터 fp 를 선언하고, 학생 수를 저장할 정수형 n 을 0 으로 초기화하여 선언하고, 반복문에서 사용할 제어 변수 i 를 선언한다. 학생 수를 구하기 위해 파일을 첫번째로 읽을 때 학생의 정보들을 임시로 저장할 STUD 구조체 변수 tmp 를 선언하고, 실제로 학생 정보들을 파일로부터 읽어 저장할 동적 메모리 공간을 위한 STUD 형 포인터 studList 를 선언한다.

```

int main(){
  > FILE* fp=NULL;
  > int n=0, i; //.n: .학생.수, .i: .반복문.제어.변수
  > STUD tmp; //.파일로부터.읽은.학생.정보를.임시로.저장해.놓을.학생.구조체
  > STUD* studList=NULL; //.학생.정보를.담을.메모리를.동적.할당받기.위한.포인터
}

```

3. 먼저, data.txt 파일을 읽기 모드로 연다. 파일 읽기를 실패하여 파일 포인터가 NULL 값을 반환하는 경우에 대한 실패 코드도 작성한다.

```

→ //파일 열기. (실패 코드 포함)
→ fp=fopen("data.txt", "rt");
→ if (fp==NULL) {
→     fprintf(stderr, "파일 열기 실패");
→     exit(1);
→ }

```

4. 먼저, 파일에 저장된 학생 수를 구하기 위한 while 반복문을 실행한다. 반복문이 실행될 조건으로는, 파일 포인터가 파일의 끝을 가리키지 않을 조건인 !feof(fp) 를 사용한다. fscanf() 함수를 이용하여 파일로부터 한 줄씩 각 학생의 정보들을 읽어 임시 구조체 변수인 tmp 의 각 멤버에 임시로 저장한다. 한 줄을 읽을 때마다 학생의 수를 저장한 n 이 0 부터 차례대로 증가한다. 파일의 끝에 도달하여 반복문이 종료된 후에는, rewind() 함수를 이용하여 다시 파일 포인터를 파일의 처음으로 옮긴다. 후에 구한 n 을 이용하여 malloc() 함수를 이용해 STUD 구조체 크기를 n 만큼 곱하여 메모리를 동적으로 할당 받고 이를 studList 포인터가 가리키도록 한다.

```

→ //학생 수 n 구하기
→ while (!feof(fp)) {
→     //파일에서 읽어 tmp에 임시로 저장
→     fscanf(fp, "%d%s%d%d%d", &tmp.id, &tmp.name, &tmp.math, &tmp.eng, &tmp.kor);
→     n++;
→ }
→ rewind(fp); //다 읽은 후에는, 포인터를 다시 파일의 처음으로 되돌리기

→ //학생 수만큼 동적 할당
→ studList = malloc(sizeof(STUD) * n);

```

5. 형식에 맞게 일부분을 출력하고, n 만큼 for 문을 반복하도록 하여 파일에 저장된 학생들의 정보를 읽어 studList 포인터가 가리키는 동적 메모리 공간의 구조체 배열에 저장한다. 저장한 후에는, 다시 sum 멤버 변수에 수학, 영어, 국어 점수를 더하여 저장하고, sum 을 3.0 으로 나눈 평균 값을 avg 멤버 변수에 저장한다. 이후 반복문이 끝나기 전에, 화면에 형식에 맞게 각 학생의 정보와 성적을 출력하도록 한다.

```

→ printf("=====|\n");
→ printf("|학...번|..이름|..수학|..영어|..국어|..총점|..평균|\n");
→ printf("=====|\n");

→ //학생들 구조체에 정보 저장 후, 출력
→ for (i=0; i<n; i++) {
→     fscanf(fp, "%d%s%d%d%d", &studList[i].id, &studList[i].name,
→         &studList[i].math, &studList[i].eng, &studList[i].kor);
→     studList[i].sum = studList[i].math + studList[i].eng + studList[i].kor;
→     studList[i].avg = studList[i].sum / 3.0;

→     printf("|%d|..%s|..%3d..|..%3d..|..%3d..|..%3d..|..%.2f|\n", studList[i].id, studList[i].name,
→         studList[i].math, studList[i].eng, studList[i].kor, studList[i].sum, studList[i].avg);
→ }
→ printf("=====|\n");


```

6. 마지막으로 `free()` 함수로 메모리를 해제하고 `fclose()` 함수로 파일을 닫은 후, 전체 프로그램을 종료하도록 한다.

```
→ free(studList); // 메모리 해제  
→ fclose(fp); // 파일 닫기  
}
```

1.4 실행창

우선 data.txt 파일에는 과제 화면에 나온 것과 같이 총 세 명의 학생 정보들이 각 줄에 저장되어 있었다.


 data.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
20170245 김지연 40 40 100
20170285 김준섭 60 20 85
20170488 전희원 80 40 20
```

data.txt

실행창은 다음과 같이, 형식에 맞게 각 학생의 학번, 이름, 수학/영어/국어 점수, 그리고 계산하여 구한 총점과 평균이 출력되었다.

 선택 Microsoft Visual Studio 디버그 콘솔

```
=====
학  번 | 이  름 | 수학 | 영어 | 국어 | 총 점 | 평  균 |
=====
20170245 | 김지연 | 40 | 40 | 100 | 180 | 60.00 |
20170285 | 김준섭 | 60 | 20 | 85 | 165 | 55.00 |
20170488 | 전희원 | 80 | 40 | 20 | 140 | 46.67 |
=====
```

C:\Users\User\Desktop\test1\Debug\test1.exe(7784 프로세스)이(가) 0 코드로
이 창을 닫으려면 아무 키나 누르세요.

1.5 느낀점

2학기가 시작되고 1학기에 이어 자료구조2 수업이 시작되었다. 방학 동안에는 다른 새로운 언어를 접하기도 하고, 여러가지 프로젝트를 진행하느라 C언어를 거의 사용하지 않았다. 때문인지 기본적인 C언어의 함수들을 사용하는 부분에서 많이 헛갈렸다. 학생 수 n 을 구하는 과정에서, while 조건문 안에 파일의 끝을 가리키는 `feof()` 함수를 사용해야 하는데, 파일 포인터가 null 값인지 확인하는 조건을 넣는 어이없는 실수를 하였다. 또 단순한 동적 할당을 이용하여 구조체 배열을 선언해야 하는데, 연결 리스트를 사용하는 것과 헛갈려서 자체 참조 구조체 포인터를 구조체 안에 선언할 뻔하기도 하였다.

앞으로 다시 자료구조 수업을 듣고, 과제들을 해 나가면서 헛갈리던 것들도 다시 복습하고 정리해야 할 것 같다. 자료구조2에서는 앞으로 트리, 그래프, 해싱, 탐색 등을 비롯해 새로운 자료의 구조들과 다양한 알고리즘 기법들을 배우게 될텐데, 정말 열심히 공부할 것이고 배우는 것들을 바탕으로 여러 알고리즘 문제들도 함께 풀어보고 싶다.