

ECE20023, Spring 2021



오픈소스 소프트웨어 실습





5

CRUD (File IO), Library (C Programming)

문자열 입력

- 공백 포함하지 않고 문자열 입력

```
scanf("%s", str);
```

```
❏ ./main  
문자열? Hello World!!!  
Hello
```

- 공백 포함하여 문자열 입력

- scanf 함수 사용 : 문자열 + %ws (개행문자 포함안됨)

```
scanf("%ws", str);
```

```
❏ ./main  
문자열? Hello World!!!  
Hello World!!!
```

- fgets 함수 사용 : 문자열 + %ws (개행문자 포함)

```
fgets(str, sizeof(str), stdin);
```

```
❏ ./main  
문자열? Hello World!!!  
Hello World!!!
```

입력버퍼

```
#include <stdio.h>

int main(){

    char str[20];
    char c;

    scanf("%s", str);

    scanf("%c", &c);

    printf("%s %c\n", str, c);

    return 0;
}
```

```
> ./main
Hello
Hello
```

- 입력버퍼에 문자가 있다면 scanf()함수가 그냥 넘어가는 경우가 있음

예) 문자열 (숫자)입력 후 문자 입력받는 경우

- 해결방법

- getchar() 함수 사용

h e l l o \0

```
scanf("%s", str);
getchar();
scanf("%c", &c);
```

- scanf 함수 대신 fgets 함수 사용(단, 개행문자가 포함될 수 있음)

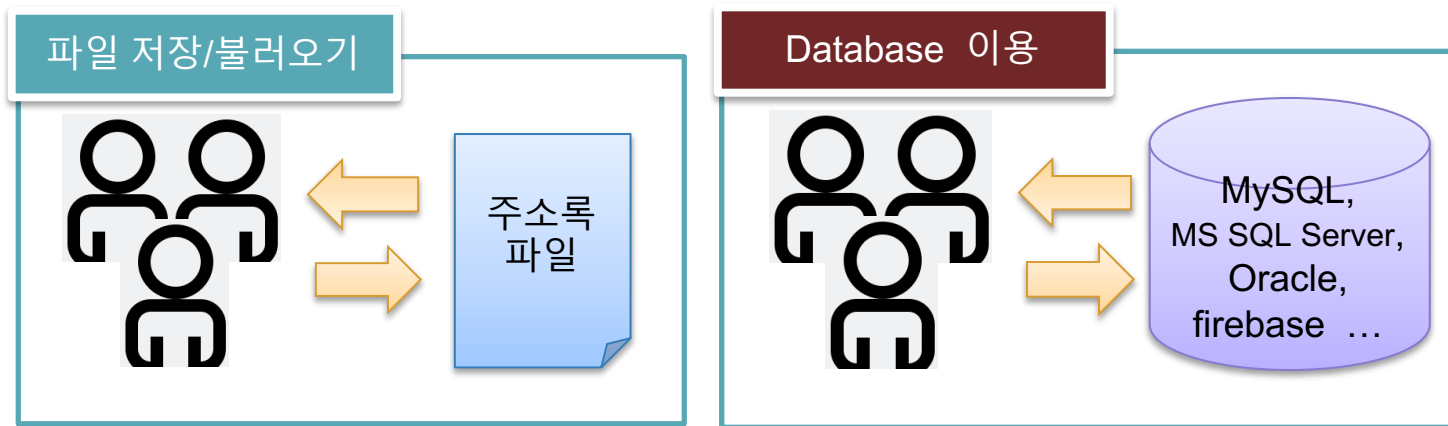
h e l l o \n \0

```
fgets(str, sizeof(str), stdin);
str[strlen(str)-1] = '\0';
```

- `scanf(" %c", &c);`

CRUD

- Create(추가), Read(조회), Update(수정), Delete(삭제)
- 다중 데이터 처리
- 데이터 저장 / 불러오기 기능(File or Database)
- 부가기능 : 검색, 보고서 기능



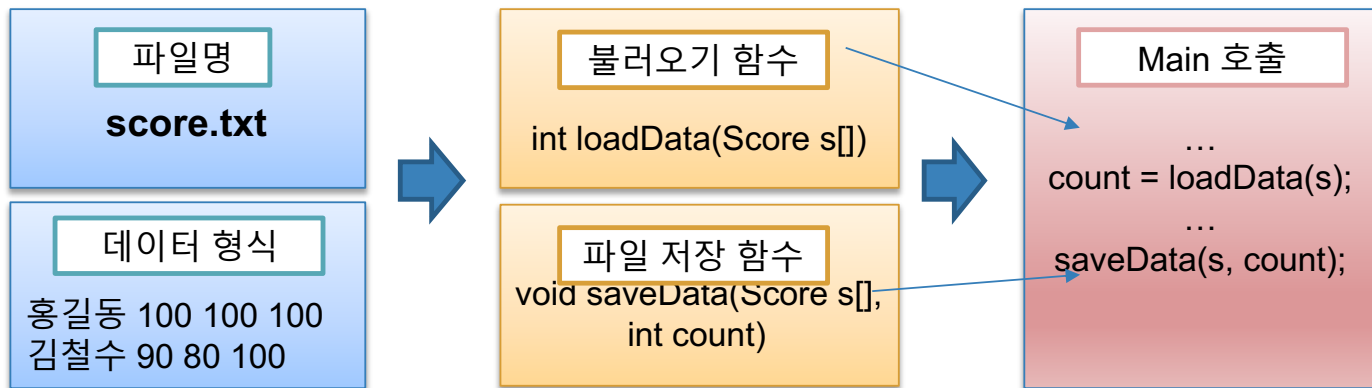
CRUD 예제 : 성적관리

```
이름은? 홍길동
국어는? 90
영어는? 90
수학은? 95
```

1. 사용자 데이터 타입 정의
2. 하나의 데이터를 추가하는 기능(함수) 구현 (C)
3. 하나의 데이터 조회 기능(함수) 구현(R)
4. 하나의 데이터 수정 기능(함수) 구현(U)
5. 하나의 데이터 삭제 기능(함수) 구현(D)
6. 각 함수를 사용하여 전체 프로그램을 구성하는 main 함수 구현
7. 다중 데이터 처리
8. 파일에 데이터 저장 기능 구현
9. 파일 데이터 가져오기 기능 구현
10. 검색 / 라이브러리 생성

File IO

1. 파일명, 파일에 작성할 데이터 포맷 정의
2. 파일에 데이터 저장하는 기능(함수) 구현
3. 파일에서 데이터 불러오는 기능(함수) 구현
4. main 함수에서 호출



CRUD 예제 : 성적관리

8. 파일에 데이터 저장 기능 구현

1. 데이터를 파일에 저장하는 함수 정의
 - saveData()/데이터배열, 개수/ 결과값 없음
2. 파일 포인터 생성
3. 파일 열기
4. 파일에 데이터 출력
5. 파일 닫기
6. main함수에서 호출

```
void saveData(Score *s, int count)
{
    FILE *fp;
    fp = fopen("score.txt", "wt");

    for(int i = 0; i < count; i++){
        if(s[i].kor == -1) continue;
        fprintf(fp, "%s %d %d %d\n"
                ,s[i].name,s[i].kor,s[i].eng
                ,s[i].math);
    }
    fclose(fp);
    printf("=> 저장됨! ");
}
```

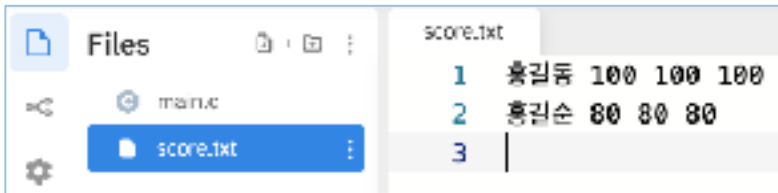

CRUD 예제 : 성적관리

8. 파일에 데이터 저장 기능 구현

- 메뉴추가 / main함수에 연결

```
*** 점수계산기 ***
1. 조회
2. 추가
3. 수정
4. 삭제
5. 저장
0. 종료
```

```
⇒ 원하는 메뉴는? 5
⇒ 저장됨!
```



```
int main(void) {
    Score slist[100];
    int count=0, index = 0, menu;

    while(1){
        menu = selectMenu();
        if(menu == 0) break;
        if(menu == 1 || menu == 3 || menu == 4)
            if(count == 0) continue;
        if(menu == 1){
            listScore(slist, index);
        }

        . . .

        else if(menu == 5){
            saveData(slist, index);
        }
    }
    return 0;
}
```

CRUD 예제 : 성적관리

9. 파일 데이터 가져오기 기능 구현

1. 파일 데이터를 가져오는 함수 정의
 - loadData()/데이터배열/ 데이터개수
2. 파일 포인터 생성
3. 파일 열기
4. 데이터 읽어오기
5. 파일 닫기
6. main함수에서 프로그램 시작시 바로 호출

```
int loadData(Score *s){
    int count = 0, i = 0;
    FILE *fp;
    fp = fopen("score.txt", "rt");
    for(; i < 100; i++){
        fscanf(fp, "%s", s[i].name);
        if(feof(fp)) break;
        fscanf(fp, "%d", &s[i].kor);
        fscanf(fp, "%d", &s[i].eng);
        fscanf(fp, "%d", &s[i].math);
    }
    fclose(fp);
    printf("=> 로딩 성공!\n");
    return i;
}
```

```
int main(void) {
    Score slist[100];
    int count=0, index = 0, menu;

    count = loadData(slist);
    index = count;

    while(1){
```

CRUD 예제 : 성적관리

```
=> 원하는 메뉴는? 6  
검색할 이름? 철수
```

No	Name	Kor	Eng	Math	Sum	Avg
2	김철수	90	90	100	280	93.3

```
=> 원하는 메뉴는? 6  
검색할 이름? 영희
```

No	Name	Kor	Eng	Math	Sum	Avg
----	------	-----	-----	------	-----	-----

```
=> 검색된 데이터 없음!
```

10. 이름 검색 기능 구현

- 이름검색 함수 생성
 - `searchName()` / 배열, 배열개수 / 결과값 없음
- `strstr`(대상 문자열, 검색문자열) 함수 사용
 - `#include <string.h>` 에 선언
- 메뉴에 추가 / `main` 함수에서 호출

CRUD 예제 : 성적관리

10. 이름 검색 기능 구현

=> 원하는 메뉴는? 6
검색할 이름? 철수

No	Name	Kor	Eng	Math	Sum	Avg
=====						
2	김철수	90	90	100	280	93.3

=> 원하는 메뉴는? 6
검색할 이름? 영희

No	Name	Kor	Eng	Math	Sum	Avg
=====						
=> 검색된 데이터 없음!						

```
#include <stdio.h>
#include <string.h>

...

void searchName(Score *s, int count){
    int scnt = 0;
    char search[20];

    printf("검색할 이름? ");
    scanf("%s", search);

    printf("\nNo Name Kor Eng Math Sum Avg\n");
    printf("===== \n");
    for(int i = 0; i < count ; i++){
        if(s[i].kor == -1) continue;
        if(strstr(s[i].name, search)){
            printf("%2d ", i+1);
            readScore(s[i]);
            scnt++;
        }
    }
    if(scnt == 0) printf("=> 검색된 데이터 없음!");
    printf("\n");
}
```

Library

1. 필요한 기능을 함수로 구현하여 모아 놓은 집합 파일로 여러 프로그램에서 재사용
2. 미리 컴파일 되어 있어서 컴파일 시간 단축됨
3. 링크될 수 있도록 컴파일된 형태의 목적 코드(Object code) 형태로 존재
4. 소스 외부 유출 방지 효과
5. 종류
 - 정적 라이브러리 (*.lib, *.a) : 프로그램을 컴파일하는 과정(link)에서 라이브러리 파일을 실행 바이너리에 포함시킴
 - 동적 라이브러리 (*.dll) : 프로그램을 실행할 때 메모리에 로드하여 동적으로 포함시킴

정적 라이브러리 실습

1. 헤더파일 생성(*.h)
2. 라이브러리에 포함할 함수 생성(*.c)
3. 라이브러리 컴파일(*.o)
4. 정적라이브러리 생성(*.a)
5. 라이브러리를 이용하여 main함수 구현
6. Object 파일을 사용한 컴파일 및 실행 (Object파일의 모든 코드 포함)
7. 정적라이브러리를 이용한 컴파일 및 실행(정적 라이브러리에서 필요한 코드만 포함)

정적 라이브러리 실습

```
#include <stdio.h>
```

```
int add(int n1, int n2);  
int subtract(int n1, int n2);  
int multiply(int n1, int n2);  
int divide(int n1, int n2);  
  
void printResult(int n1, int n2, int r, char op);
```

```
int main(){  
    int n1,n2, result;  
  
    char op;  
  
    printf("두 수를 입력 :");  
    scanf("%d %d", &n1, &n2);  
    getchar();  
    printf("연산자 입력(+, -, *, /) :");  
    scanf("%c", &op);  
  
    if(op == '+') result = add(n1, n2);  
    else if(op == '-') result = subtract(n1, n2);  
    else if(op == '*') result = multiply(n1, n2);  
    else if(op == '/') result = divide(n1, n2);  
    else{  
        printf("=> 잘못 입력!\n");  
        return 0;  
    }  
    printResult(n1,n2,result,op);  
    return 0;  
}  
  
int add(int n1, int n2){return n1 + n2;}  
int subtract(int n1, int n2){return n1 - n2;}  
int multiply(int n1, int n2){return n1 * n2;}  
int divide(int n1, int n2){return n1 / n2;}  
void printResult(int n1, int n2, int r, char op){  
    printf("=> %d %c %d = %d\n",n1,op,n2,r);  
}
```

정적 라이브러리 실습

1. 헤더파일 생성(calculator.h)

```
int add(int n1, int n2);
int subtract(int n1, int n2);
int multiply(int n1, int n2);
int divide(int n1, int n2);
void printResult(int n1, int n2, int r, char op);
```

2. 라이브러리에 포함할 함수 생성(calculator.c)

```
#include <stdio.h>
#include "calculator.h"
```

```
int add(int n1, int n2){return n1 + n2;}
int subtract(int n1, int n2){return n1 - n2;}
int multiply(int n1, int n2){return n1 * n2;}
int divide(int n1, int n2){return n1 / n2;}
void printResult(int n1, int n2, int r, char op){
    printf("=> %d %c %d = %d\n",n1,op,n2,r);
}
```


정적 라이브러리 실습

3. 라이브러리 컴파일(*.o)

```
User1-ui-Macmini inc % gcc -c calculator.c
User1-ui-Macmini inc % ls
calculator.c    calculator.h    calculator.o
```

4. 정적라이브러리 생성(*.a) : 여러 Object 파일을 하나로 묶음

```
User1-ui-Macmini inc % ar rv libcal.a calculator.o
ar: creating archive libcal.a
a - calculator.o
User1-ui-Macmini inc % ls
calculator.c    calculator.h    calculator.o    libcal.a
```

정적 라이브러리 실습

5. 라이브러리를 이용하여 main함수 구현 (mycal.c)

```
#include <stdio.h>
#include "calculator.h"

int main(){
    int n1,n2, result;
    char op;

    printf("두 수를 입력 :");
    scanf("%d %d", &n1, &n2);
    getchar();
    printf("연산자 입력(+, -, *, /) :");
    scanf("%c", &op);

    if(op == '+') result = add(n1, n2);
    else if(op == '-') result = subtract(n1, n2);
    else if(op == '*') result = multiply(n1, n2);
    else if(op == '/') result = divide(n1, n2);
    else{
        printf("=> 잘못 입력!\n");
        return 0;
    }
    printResult(n1,n2,result,op);
    return 0;
}
```

정적 라이브러리 실습

6. Object파일을 이용한 컴파일 및 실행 파일 생성

- Object파일의 모든 코드를 가져와 실행파일 생성

```
User1-ui-Macmini 5_lab % gcc ./inc/calculator.o mycal.c -o mycal2
User1-ui-Macmini 5_lab % ls mycal2
mycal2
User1-ui-Macmini 5_lab % ./mycal2
두 수를 입력 :10 20
연산자 입력(+, -, *, /) :+
=> 10 + 20 = 30
```

정적 라이브러리 실습

7. 정적 라이브러리를 이용한 컴파일 및 실행 파일 생성

- 필요한 코드만 가져와 하나의 실행파일로 생성함

```
User1-ui-Macmini 5_lab % ls ./inc  
calculator.c    calculator.h    calculator.o    libcal.a  
User1-ui-Macmini 5_lab % gcc mycal.c -o mycal -I./inc -L./inc -lcal  
User1-ui-Macmini 5_lab % ls mycal  
mycal
```

- **-o mycal** mycal 실행파일 생성
- **-I./inc** 현재 폴더 아래 inc 폴더에서 헤더파일을 찾는 옵션(calculator.h)
- **-L./inc** 현재 폴더 아래 inc폴더에서 라이브러리를 찾는 옵션(libcal.a)
- **-lcal** 정적라이브러리인 libcal.a와 링크하라는 옵션



요약

1. 파일 입출력 과정의 이해
2. 파일 입출력에 필요한 함수 사용법
3. 검색 함수의 이해(strstr)
4. 정적라이브러리 vs 동적라이브러리
5. 정적라이브러리 생성법
6. Object파일을 이용한 실행파일 생성
7. 정적라이브러리를 이용한 실행파일 생성

과제설명

- * 모든 과제 제출시 **학번_이름.pdf** 하나의 파일에 각 단계를 저장하여 제출할 것
- * 과제를 위한 결과 참고 자료 링크 : <https://docs.google.com/document/d/1VtNjxp9rDc3YMS-QwV74bTKho1FbVZsYUoJe-mdkq/edit?usp=sharing>

1. LAB2(3번)에서 만든 소스(배열버전)를 활용하여 파일 입출력과 이름검색이 가능하도록 코딩하여 [소스\(텍스트\)](#), [결과\(텍스트\)](#)와 [\(소스와 결과화면이 보이는\)전체 vscode 화면 캡처](#)를 제출하세요. (vscode로 개발, 배열버전: score4.c 결과화면은 뒤 참고링크 참조)

* 반드시 결과 화면을 참조하여 테스트할 때 동일한 결과가 나오야 함

* 프로그램을 시작하면 score.txt 파일을 찾아서 파일이 없다면 "=> 파일 없음"을 출력하고, 파일이 있다면 파일에 있는 정보를 읽어와야 함

파일 내용 : score.txt

```
홍길동 100 100 100
김사광 80 80 30
최소당 90 80 90
```

2. (목욕) LAB2(4번)에서 만든 소스(배열버전)를 활용하여 위 예제의 출력결과를 참조하여 파일 입출력과 이름검색이 가능하도록 코딩하여 [소스\(텍스트\)](#), [결과\(텍스트\)](#)와 [\(소스와 결과화면이 보이는\)전체 vscode 화면 캡처](#)를 제출하세요. (vscode로 개발) (배열버전: menu4.c)
3. [라이브러리 생성] 1번에서 작성한 소스를 이용하여 CRIUD 함수를 라이브러리로 생성하고, main함수와 빌드한 후 실행해 보고, [완성된 파일별 소스 코드와 실행화면 결과화면 캡처](#)하여 제출하세요. (score.h, score.c, scoremain.c)
4. dymalist 링크에 아래 항목을 추가하여 문법 정리한 후 링크 제출합니다. (sharing 필수), 검색한 링크를 추가해도 됩니다. 이전에 마루정도 수정해도 됩니다. (보인이 추후 계속사용할 수 있도록 알찬 문서로 만들어 보세요)
참조링크 : <https://jy1elisl.usd/A/HuslKc79c3cy6DTwae31T>

- 파일에 저장하는 과정/파일에서 데이터 불러오는 과정
- 동지열 검색방법
- 라이브러리란?
- C언어 포인터와 브래킷?
- 중괄호 브래킷과 통적 라이브러리
- 라이브러리 생성방법
- 라이브러리 컴파일 및 라이브러리를 이용한 델트 방법(gcc)