

실습: *Week 2*

Data Structures

Contents

■ C language

- 구조체(structure)
- 문자열(strings)
- 파일 I/O(file I/O)
- 헤더 파일(header file)

■ 실습

- 실습 . 학생 성적 관리 프로그램
 - 파일로부터 학생 성적 구조체로 성적 읽기
 - 성적 출력, 성적 찾기, 성적 수정, 성적 저장 등 명령 처리

구조체 (1/2)

- 구조체의 정의
 - 서로 다른 자료형의 변수들을 묶어서 만든 자료 구조
- 구조체 형 선언

```
typedef struct <name> {  
    type  member1;  
    type  member2;  
    ...  
} <name>
```

- Example

```
typedef struct {  
    char name[20];  
    int  id;  
} student
```

구조체 (2/2)

■ 직접 멤버 참조 - 도트 연산자(.)

```
typedef struct {  
    char name[20];  
    int id;  
} student;  
student st1;
```



```
st1.name = "김준태"; // error  
  
strcpy(st1.name, "김준태"); // ok  
st1.id = 123;
```

■ 간접 멤버 참조 (포인터 변수) - 화살표 연산자(->)

```
typedef struct {  
    char name[20];  
    int id;  
} student;  
student st1, *st2;  
st2 = &st1;
```



```
strcpy(st2->name, "김준태");  
st2->id = 123;
```

Strings(1/4)

- String
 - char 형의 1차원 배열
 - 문자열은 끝의 기호인 `'\0'` (널 문자, 모든 비트가 0)로 끝남
 - 문자열의 크기는 `'\0'` 까지 포함
- String Constant
 - 큰따옴표 안에 기술됨
 - `"abc"` : 크기가 4인 문자 배열
- 주의 - `"a"`와 `'a'`는 다름
 - `'a'` : 1 byte character
 - `"a"` : 2 byte string (첫 번째 원소 `'a'`, 두 번째 원소 `'\0'`)

Strings(2/4)

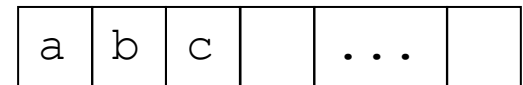
- 스트링의 지정(선언시)

```
char s[] = "abc"; 또는  
const char *s = "abc";
```

- 스트링의 지정(선언후)


```
char newstr[100];  
strcpy(newstr, "abc");
```

new



```
char *newstr;  
newstr = (char*)malloc(strlen("abc")+1);  
strcpy(newstr, "abc");
```

new



Strings(3/4)

- `string.h`에 정의
 - `size_t strlen(const char *s);`
 - `\0`을 뺀 문자의 개수를 리턴
 - `char *strcpy(char *s1, const char *s2);`
 - `s2`의 문자를 `\0`이 나올 때까지 `s1`에 복사
 - `int strcmp(const char *s1, const char *s2);`
 - `s1`과 `s2`를 비교하여, `s1`이 작으면 음수, 크면 양수, 같으면 `0`을 리턴
 - `char *strcat(char *s1, const char *s2);`
 - 두 문자열 `s1,s2`를 결합하고, 결과를 `s1`에 저장
- `strcpy, scanf, fopen, fscanf` 등 취약점이 있는 함수
 - 사용 시 경고 삭제를 위해 헤더 파일 상단에 다음 삽입
 - `#define _CRT_SECURE_NO_WARNINGS`

Strings(4/4)

- StudentRecord 구조체

```
typedef struct {  
    char    name[10];  
    int     id;  
    char    grade[4];  
} StudentRecord;  
StudentRecord r[MAX];
```

- r[]에서 name 이 n1 인 자료의 grade 를 g1 으로 바꾸기

```
for (i=0; i<MAX; i++) {  
    if(!strcmp(r[i].name, n1))  
        strcpy(r[i].grade, g1);  
}
```


파일 I/O (1/2)

■ FILE 구조체

- 파일에 대한 정보는 **FILE** 구조에 담겨 있음
- 파일에 접근하기 위해서 **FILE** 포인터 사용

```
FILE *fp;
```

■ 파일 open/close

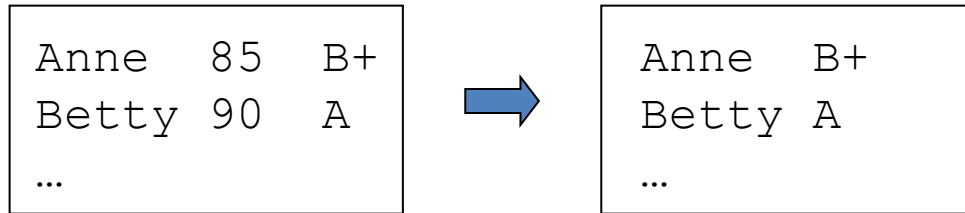
- 파일은 사용되기 전에 **open**, 사용 후 **close** 해야함

```
fp = fopen("sample.txt", "r");
```

■ 파일 read/write

- 파일 입출력을 위한 함수 사용
 - `fprintf(file_pointer, str, variable_lists)` : 파일에 쓰기
 - `fscanf(file_pointer, conversion_specifiers, variable_addresses)` : 파일에서 읽기
 - `fgets(buffer, n, file_pointer)` : 파일에서 **n-1**개의 문자를 읽고 버퍼에 저장

파일 I/O (1/2)



```
int n;  
char name[10], grade[5];  
FILE *ifp, *ofp;  
  
ifp = fopen("data.txt", "r");  
ofp = fopen("out.txt", "w+");  
  
while(fscanf(ifp, "%s %d %s", name, &n, grade) == 3)  
    fprintf(ofp, "%10s %4s\n", name, grade);  
  
fclose(ifp);  
fclose(ofp);
```

헤더 파일 (*.h)

- A large program
 - Written in a directory - collection of .h and .c files.
- Header file 이 포함하는 것
 - `#include`
 - 상수 정의(`#define`)
 - 구조체 등 형 선언(`typedef of structures`)
 - 함수 원형(`function prototypes`)
- Header file 이 각 .c 파일의 상단에 위치
 - 프로그램들을 하나로 묶는 역할

실습 . 학생 성적 관리

- 다음과 같은 'grade.txt' 파일로부터 이름, 학번, 성적 자료를 읽어 아래와 같은 명령을 처리하는 프로그램을 작성하시오. 자료의 저장은 주어진 구조체의 배열을 사용한다.

grade.txt

Anne	20001001	A
Betty	20001002	B
Cindy	20001003	C
Diane	20001004	D

- 명령어
 - 성적 출력(P), 성적 찾기(S), 성적 수정(C), 파일에 저장(W)

실습 . 실행 예(1)

```
선택 Microsoft Visual Studio 디버그 콘솔
Reading record from grade.txt ...
4 records have read.
*****명령어 *****
P : Print all records
S: Search record
C : Change record
W : Write record
Q : Save and quit
*****

command> P
-----
      Anne  20001001      A
      Betty 20001002      B
      Cindy 20001003      C
      Diane 20001004      D
-----

command> S Search name : Anne
      Name : Anne
      ID   : 20001001
      Grade : A
```

실습 . 실행 예(2)

```
선택 Microsoft Visual Studio 디버그 콘솔
command> C
  Name : Diane
  Grade : A

Record changed!

command> P
-----
    Anne  20001001    A
    Betty 20001002    B
    Cindy 20001003    C
    Diane 20001004    A
-----

command> W
4 records have written to grade.txt

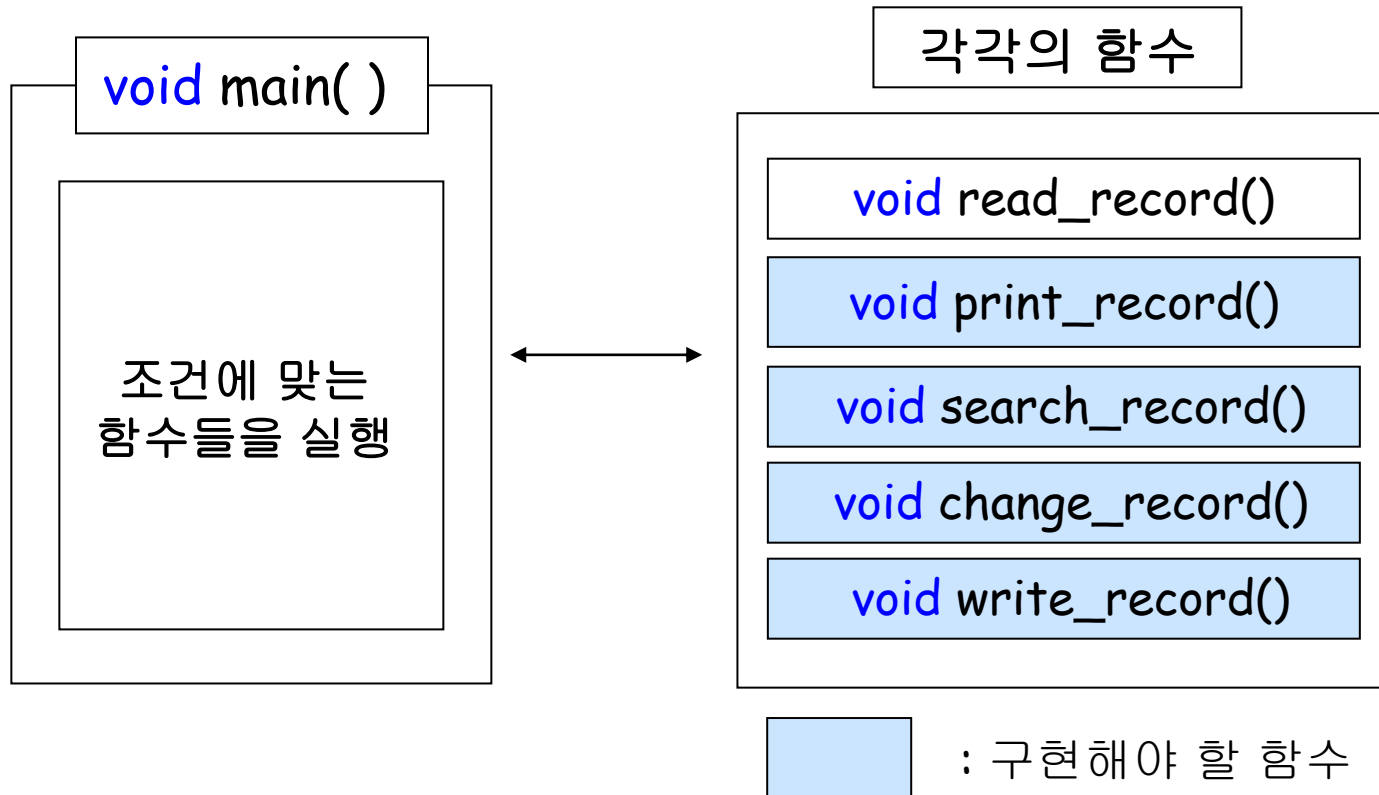
command> Q

C:\Users\ch1tn\source\2022_자료구조실습\week2\Debug\week2.exe(프
되었습니다(코드: 1개).
이 창을 닫으려면 아무 키나 누르세요...
```

```
Anne 20001001 A
Betty 20001002 B
Cindy 20001003 C
Diane 20001004 A
```

grade.txt

자료구조 및 함수 구성



자료구조 및 함수 구성

- ```
typedef struct {
 char name[10];
 int id;
 char grade[4];
} StudentRecord;
```
- `void read_record(char *fname)`
  - **Requires** : 파일 이름
  - **Results** : 파일에서 레코드를 하나씩 읽어 구조체의 배열에 저장



# 자료구조 및 함수 구성

- `void print_record( )`
  - **Requires** : 없음
  - **Results** : 구조체의 배열을 차례로 읽어 자료 전체를 출력
- `void search_record( )`
  - **Requires** : 없음
  - **Results** : 이름을 입력 받아 해당 학생의 자료를 출력
- `void change_record( )`
  - **Requires** : 없음
  - **Results** : 이름과 성적을 입력 받아 해당 자료를 수정
- `void write_record(char *fname)`
  - **Requires** : 파일 이름
  - **Results** : 구조체의 배열 차례로 읽어 자료 전체를 파일에 저장

# ex2.h

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <conio.h>

// Constants
#define MAX 100

// Structures
typedef struct {
 char name[10];
 int id;
 char grade[4];
} StudentRecord;
```

# ex2.h

```
// Global variables
StudentRecord r[MAX]; // Student record: name, id, grade
int num_record = 0; // Total # of records read from file
const char *fname = "grade.txt"; // Grade file name

// Function prototypes
void read_record(char *fname);
void print_record();
void search_record();
void change_record();
void write_record(char *fname);
```

# ex2.c - main() 함수

```
#include "ex2.h"
```

```
int main(void)
```

```
{
```

```
 char c;
```

```
 printf("Reading record from %s\n",fname);
```

```
 read_record(fname);
```

```
 printf("%d records have read.\n", num_record);
```

```
 printf("***** 명령어 ***** \n");
```

```
 printf("P: Print all records \n");
```

```
 printf("S : Search record \n");
```

```
 printf("C : Change record \n");
```

```
 printf("W : Write record \n");
```

```
 printf("Q : Save and quit \n");
```

```
 printf("***** \n");
```

# ex2.c - main() 함수

```
while (1) {
 printf("\nCommand> ");
 c = getche();
 c = toupper(c);
 printf("\n");

 switch (c) {
 case 'P':
 print_record();
 break;
 case 'S':
 search_record();
 break;
 }
```

# ex2.c - main() 함수

```
 case 'C':
 change_record();
 break;
 case 'W':
 write_record(fname);
 break;
 case 'Q':
 printf("\n");
 exit(1);
 break;
 default:
 printf("Unknown command ! \n");
 break;
 }
}
return 0;
}
```

## ex2.c - read\_record() 함수

```
void read_record(char* fname)
{
 int i = 0;
 FILE* ifp;

 ifp = fopen(fname, "r");
 while (fscanf(ifp, "%s %d %s", r[i].name, &r[i].id, r[i].grade) == 3)
 ++i;

 num_record = i;
 fclose(ifp);
}
```