

## 14주차 3차시 파일 입출력, 메모리 관리 실습

### 【학습목표】

1. 실습을 통하여 파일 입출력을 실행할 수 있다.
2. 실습을 통하여 메모리 관리를 실행할 수 있다.

### 학습내용1 : 파일 입출력 실습

- 프로그램상에서 mystory.txt라는 이름의 파일을 생성해서 본인의 이름, 주민번호, 전화번호를 저장하는 프로그램을 작성하자.  
단 저장의 형태는 다음과 같아야 한다. (#문자도 함께 저장되어야 한다).  
#이름: 홍길동  
#주민번호: 123456-7890123  
#전화번호: 010-9999-8888  
그리고 저장이 완료되면 메모장으로 확인이 가능해야 한다.
- 위 문제에서 작성한 파일에 데이터를 추가하자. 추가할 데이터는 즐겨먹는 음식의 정보와 취미이다.  
입력의 형태는 다음과 같아야 한다.  
#즐거먹는 음식: 짬뽕, 탕수육  
#취미: 등산
- 위의 문제들에서 생성된 파일에 저장된 정보 전체를 출력하는 프로그램을 작성하자.

- 서식에 따른 데이터 입출력 예제 실습(교재 14주차 2차시 예제 참조)

도서 source: ComplecFileWrite.c

```
int main(void)
{
    char name[10];
    char sex;
    int age;
    FILE * fp=fopen("friend.txt", "wt");
    int i;
    for(i=0; i<3; i++)
    {
        printf("이름 성별 나이 순 입력: ");
        scanf("%s %c %d", name, &sex, &age);
        getchar(); // 버퍼에 남아있는 \n의 소멸을 위해서
        fprintf(fp, "%s %c %d", name, sex, age);
    }
    fclose(fp);
    return 0;
}
```

저장하는 데이터가 문자열이므로  
텍스트 모드로 개방한다!

■ 서식에 따른 데이터 입출력 예제 실습(교재 14주차 2차시 예제 참조)

도서 source: ComplecFileRead.c

```
int main(void)
{
    char name[10];
    char sex;
    int age;
    FILE * fp=fopen("friend.txt", "rt");
    int ret;
    while(1)
    {
        ret=fscanf(fp, "%s %c %d", name, &sex, &age);
        if(ret==EOF)
            break;
        printf("%s %c %d \n", name, sex, age);
    }
    fclose(fp);
    return 0;
}
```

## 학습내용2 : 메모리 관리 실습

- 서식에 따른 데이터 입출력 예제 실습(교재 14주차 2차시 예제 참조)

도서 source: DynamicMemoryAllocation.c

```
int main(void)
{
    int * ptr1 = (int *)malloc(sizeof(int));
    int * ptr2 = (int *)malloc(sizeof(int)*7);
    int i;

    *ptr1 = 20;
    for(i=0; i<7; i++)
        ptr2[i]=i+1;

    printf("%d \n", *ptr1);
    for(i=0; i<7; i++)
        printf("%d ", ptr2[i]);

    free(ptr1);
    free(ptr2);
    return 0;
}
```

- 문자열을 반환하는 함수를 정의하는 문제 해결 예제 실습(교재 14주차 2차시 예제 참조)

도서 source: ReadStringRight.c

```
char * ReadUserName(void)
{
    char * name = (char *)malloc(sizeof(char)*30);
    printf("What's your name? ");
    gets(name);
    return name;
}
```

할당!

```
int main(void)
{
    char * name1;
    char * name2;
    name1=ReadUserName();
    printf("name1: %s \n", name1);
    name2=ReadUserName();
    printf("name2: %s \n", name2);

    printf("again name1: %s \n", name1);
    printf("again name2: %s \n", name2);
    free(name1);
    free(name2);
    return 0;
}
```

소멸!

### 【학습정리】

1. 콘솔 입출력과 마찬가지로 파일로부터의 데이터 입출력을 위해서는 스트림이 형성되어야 한다. 이럴 때 fopen 함수가 호출되고 FILE 구조체 변수가 생성된 후 생성된 FILE 구조체 변수에는 파일에 대한 정보가 담기게 된다.
2. 스트림의 소멸을 요청하는 fclose 함수는 운영체제가 할당한 자원을 반환하고 버퍼링 되었던 데이터를 출력한다.
3. 파일위치 지시자는 FILE구조체의 멤버중 하나로 “어디까지 읽었는가”에 대한 것은 read모드로 오픈된 파일 위치 지시자가 정보를 갖고 있고, “어디부터 이어서 쓰는가”에 대한 정보는 write 모드로 오픈된 파일 위치 지시자가 정보를 가지고 있다.