



실 습 문 제

제16장

1. [텍스트 파일] 텍스트 파일에 데이터를 읽고 쓰는 과정에 대하여 실습을 하여 보자.

```
#include <stdio.h>

int main(void)
{
    FILE *fp;
    int ch;

    fp = fopen("proverb.txt", "r");
    if( fp == NULL )
    {
        printf("파일 proverb.txt을 열수 없습니다.\n");
        return 1;
    }

    while( (ch = fgetc(fp)) != EOF )// ①
        putc(ch);

    fclose(fp);

    return 0;
}
```

- (a) 메모장을 실행하여 다음과 같은 내용을 입력하고 test.txt란 이름으로 저장한다.

```
A chain is only as strong as its weakest link
A change is as good as a rest
A fool and his money are soon parted
A friend in need is a friend indeed
```

- (b) 위의 프로그램을 컴파일하여 실행하고 그 결과를 기록하라.

실행결과

- (c) putc(ch);을 fputc(ch, stdout);로 변경하면 어떤 결과가 얻어지는가?

- (d) 문장 ①을 while(1)과 break문을 사용하여 작성하여 보라.

- (e) 파일 이름을 사용자로부터 받도록 프로그램을 수정하여 보라.

- (f) fgets()와 fputs()를 사용하도록 프로그램을 수정하여 보라.

- (g) proverb.txt에서 읽은 내용을 output.txt라 이름의 파일로 저장하게끔 프로그램을 수정하여 보라.



2. [형식화된 입출력] 텍스트 파일에 형식을 지정하여 입출력을 하는 실습을 하여 보자.

```
#include <stdio.h>

int main(void)
{
    FILE *fp;
    char name[80];
    int k, m, e;
    double average;

    fp = fopen("input.txt", "r");
    if( fp == NULL )
    {
        printf("파일 input.txt를 열수 없습니다.\n");
        return 1;
    }

    while( 1 )
    {
        if( fscanf(fp, "%s %d %d %d", name, &k, &m, &e) != 4 )
            break;
        average = (k + m + e) / 3.0;
        printf("%s %f\n", name, average);
    }

    fclose(fp);
    return 0;
}
```

(a) 메모장을 실행하여 다음과 같은 내용을 입력하고 input.txt란 이름으로 저장한다.

```
김철수 90 88 70
김세희 60 72 80
한철균 90 90 86
```

(b) 위의 프로그램을 컴파일하여 실행하고 그 결과를 기록하라.

실행결과

(c) 화면으로 출력하지 말고 output.txt로 출력하도록 프로그램을 수정하여 보라.

(d) 출력 형식을 다음과 같이 변경하여 보라. 학생 이름은 필드폭이 10에 오른쪽 정렬, 평균은 필드폭 10에 소수점 이하 3자리, 왼쪽 정렬하여 보라.

(e) 출력시에 각 라인에 번호를 매겨서 출력하여 보라.

3. [텍스트 파일과 이진 파일] 텍스트 파일과 이진 파일의 차이점을 살펴보자.

```
#include <stdio.h>

int main(void)
{
    FILE *fpt, *fpb;
    int i = 1234567890;

    fpt = fopen("test.txt", "w");
    fpb = fopen("test.dat", "w");

    if( fpt == NULL || fpb == NULL )
    {
        printf("파일을 열수 없습니다.\n");
        return 1;
    }
    fprintf(fpt, "%d", i);
    fwrite(&i, sizeof(int), 1, fpb);

    fclose(fpt);
    fclose(fpb);

    return 0;
}
```

- 위의 프로그램을 컴파일하여 실행하고 현재 디렉토리에 생성되는 파일 `test.txt` 와 `test.dat`의 크기를 비교하여 보라. 차이를 설명하라. 파일의 내용을 살펴보자.
- 100개의 난수를 발생시켜서 `test.txt`와 `test.dat`에 저장한 후에 파일 크기를 비교하여 보라.
- 정수가 아닌 1.234567890과 같은 실수를 저장하는 경우에는 어떻게 되는지를 실험하여 보라.

4. [이진 파일 출력] 사용자로부터 다음과 같은 형식으로 이름과 전화번호를 입력받아서 이진 파일로 저장한다.

```
#include <stdio.h>

struct phonebook {
    char name[20];
    int tel;
};

int display_menu(void)
{
    int choice;
```



```
printf("-----+-----\n");
printf("1 --- 레코드 입력\n");
printf("2 --- 레코드 탐색\n");
printf("3 --- 레코드 변경\n");
printf("4 --- 레코드 삭제\n");
printf("5 --- 종료\n");
printf("-----\n");

scanf("%d", &choice);
return choice;
}

int main(void)
{
    FILE *fp;
    struct phonebook e;
    int selected;

    fp = fopen("phonebook.dat", "a");
    if( fp == NULL )
    {
        printf("파일 phonebook.dat를 열수 없습니다.\n");
        return 1;
    }
    while( (selected = display_menu()) != 5 )
    {
        switch( selected )
        {
            case 1:
                printf("이름과 전화번호를 입력하시오:\n");
                scanf("%s %d", e.name, &e.tel);
                fwrite(&e, sizeof(e), 1, fp);
                break;
        }
    }

    fclose(fp);
    return 0;
}
```

(a) 다음과 같이 입력하여 보자.

```
김철수 0111111111
김세희 0102222222
한철균 0163333333
```

(b) 2번 메뉴인 탐색은 구현되어 있지 않다. 사용자로부터 이름을 입력받아서 그 이름에 해당하는 전화번호를 출력하는 코드를 작성하고 테스트하라. seek()를 이

용하여 이진 파일의 처음으로 이동하여서 탐색을 수행한다.

- (c) 3번 메뉴인 변경도 구현되어 있지 않다. 사용자로부터 이름을 입력받아서 그 이름에 해당하는 전화번호를 새로운 전화번호로 변경하는 코드를 작성하고 테스트하라.
- (d) 4번 메뉴인 삭제도 구현되어 있지 않다. 사용자로부터 이름을 입력받아서 그 이름에 해당하는 레코드를 삭제한다. 여기서는 이진 파일에서 실제로 삭제하는 것은 아니고 해당 위치에 있는 레코드를 빈 레코드로 대체하도록 한다.