

교학습내용

♀ 14.3 텍스트 파일 입출력





할 학습목표

♀ 14.3 텍스트 파일 읽기와 쓰기를 활용할 수 있다.





C프로그래밍및실습 파일 입출력 개요 2. 파일 입출력 절차 3. 텍스트 파일 입출력



지정 형식 파일 입력 함수: fscanf()

- 데이터를 지정한 형식으로 데이터를 읽어 오는 함수
- 여러 형태의 자료들(정수, 문자, 문자열 등)을 한번에 입력 가능
- 함수의 첫 번째 인자로 파일 포인터가 사용된다는 것을 제외하고는 scanf() 함수와 사용법 동일

함수원형	<pre>int fscanf(FILE *fp, char *format,);</pre>						
함수인자	fp	파일 포인터 변수명					
	format	형식 제어 문자열					
	•••	입력하고자 하는 변수리스트					
반환 값	✓ 성공 → 입력한 변수의 개수를 반환✓ 읽기 실패 → E0F를 반환						

scanf() 함수 사용 예

```
char name[10] = {'\0'};
double weight;
int age;
FILE *fp = fopen("input.dat", "r");
fscanf(fp, "%s %d %lf", name, &age, &weight);
         // input.dat 파일로부터 데이터를 입력받음
fscanf(stdin, "%s %d %lf", name, &age, &weight);
         // 키보드로부터 데이터를 입력받음
```

✓ 첫 번째 인자를 stdin으로 지정하면, 표준입력으로부터 입력 (scanf() 함수를 사용한 것과 동일한 기능)





■ 지정 형식 파일 출력 함수: fprintf()

- 지정한 형식에 맞추어 파일로 출력
- 함수의 첫 번째 인자로 파일 포인터가 사용된다는 것을 제외하고는 printf() 함수와 사용법 동일

함수원형	<pre>int fprintf(FILE *fp, char *format,);</pre>						
	fp	파일 포인터 변수명					
함수인자	format	형식 제어 문자열					
	•••	출력하고자 하는 변수리스트					
반환 값	✓ 성공 → 출력한 데이터의 바이트 수✓ 실패 → 음수를 반환						

ifprintf() 함수 사용 예

```
char name[10] = "Tommy";
double weight = 78.45;
int age = 25;
FILE *fp = fopen("output.dat", "w");
fprintf(fp, "%s %d %.2lf", name, age, weight);
           // output.dat 파일로 데이터를 출력
fprintf(stdout, "%s %d %.2lf", name, age, weight);
           // 모니터로 데이터를 출력...
```



문자 단위 파일 입출력 함수: fgetc() 와 fputc()

● 문자 한 개를 파일로부터 읽고 쓰기 위한 함수

함수 원형	<pre>int fgetc(FILE *fp);</pre>						
함수 인자	fp 파일 포인터 변수명						
반환 값	✓ 성공 → 읽은 문자 반환✓ 실패(파일 끝 또는 오류 발생) → EOF 반환						
함수 원형	<pre>int fputc(int ch, FILE *fp);</pre>						
하스 이기	ch	출력하고자 하는 문자 상수 또는 변수					
함수 인자	fp	파일 포인터 변수명					
반환 값	✓ 성공 → 출력하는 문자 ch를 반환✓ 실패 → EOF 반환						



※실습하기



프로그램 14-3: 문자 단위의 파일 입출력

- 1. 'x'가 입력될때까지 문자를 반복적으로 입력 받아 example.txt 파일에 문자 단위로 출력
- 2. example.txt 파일의 내용을 화면에 문자 단위로 출력

※실습하기



프로그램 14-3: 문자 단위의 파일 입출력 (주요 코드)

```
char fn[] = "example.txt";
                      // 출력 파일 열기
FILE *fp = fopen(fn, "w");
                        // 문자 하나를 표준입력으로 입력
ch = getchar();
                    // 'x'가 입력될 때까지 반복
while (ch != 'x') {
                        // 출력 파일로 문자 한 개 출력
  fputc(ch, fp);
                        // 키보드로부터 문자 한 개를 입력
  ch = getchar();
                        // 출력 파일 닫기
fclose(fp);
                        // 입력 파일 열기
fp = fopen(fn, "r");
                        // 입력파일에서 문자 한 개 읽음
ch = fgetc(fp);
                        // 파일의 끝까지 반복 (추후 학습)
while (!feof(fp)) {
                        // 읽은 문자를 모니터로 출력
  putchar(ch);
                        // 입력파일에서 문자 한 개 읽음
  ch = fgetc(fp);
                        // 입력 파일 닫기
fclose(fp);
```



■ 문자열 단위의 파일 입력 함수: fgets()

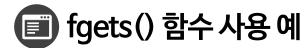
● 파일에 쓰여진 문자열을 읽는데 사용하는 함수

함수 원형	<pre>char *fgets(char *s, int n, FILE *fp);</pre>						
	S	파일로부터 읽은 문자열을 저장할 포인터					
함수 인자	n	저장할 문자의 최대 개수(널 문자 포함)					
	fp 파일 포인터 변수명						
 ✓ 성공 → 문자열 s를 반환 반환 값 ✓ 파일의 끝에 도달하거나 실패/오류 발생 → NULL을 반환 							

[fgets()의기능

- 다음 조건이 만족될 때까지 파일 fp로 부터 문자를 읽어서 s가 가리키는 곳에 저장
 - ✓ n-1 개의 문자(공백, 탭 문자 포함)를 읽거나,
 - ✓ 개행 문자를 만나거나,
 - ✓ 파일의 끝에 도달
- 문자열 마지막에 널 문자 추가
- 다른 입력 함수와 다른 점
 - ✓ fscanf와 달리, 공백, 탭 문자를 만나도 계속 읽음
 - ✓ 개행 문자도 문자열에 저장





info.txt

Neungdong-ro, ↓ Gwangjin-gu, Seoul, Korea. ↓

1) fgets(str1, 20, fp);

✓ 중간에 개행 문자를 만나 읽기 종료. 마지막에 널 문자 추가

N	е	u	n	g	d	0	n	g	_	r	0	,	\n	\0			
	l		l	•				•		l		_	_	_			

2) fgets(str2, 20, fp);

✓ 19개의 문자를 읽은 후 뒤에 널 문자를 합쳐 str2 배열에 저장

G	W	а	n	g	j	i	n	-	g	u	,		S	е	0	u	l	,	\0
---	---	---	---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	---	----

3) fgets(str3, 20, fp);

✓ 이전에 읽은 곳 다음부터 읽다가 19개의 문자를 읽기 전에 개행 문자를 읽게 되어, 그 뒤에 널 문자를 합쳐 str3 배열에 저장

K	0	r	е	а	•	\n	\0						



■ 문자열 단위의 파일 출력 함수: fputs()

- 파일에 문자열을 출력하는 함수
- 문자열의 끝을 나타내는 널 문자는 파일에 쓰지 않으며, 그 뒤에 개행 문자도 <mark>자동으로</mark> 들어가지 <mark>않음</mark>
 - ✓ (비교) puts() 함수는 문자열 출력 후 자동으로 개행

함수 원형	<pre>int fputs(char *str, FILE *fp);</pre>					
하스 이기	str	출력하고자 하는 문자열				
함수 인자	fp	파일 포인터 변수명				
반환 값	 ✓ 성공 → 출력한 바이트 수를 반환 ✓ 실패/오류발생 → E0F를 반환 					

※ 실습하기



프로그램 14-4 : 문자열 단위 파일 입출력

- 1. 표준입력으로부터 국가명을 입력 받고, 이를 Cnty_list.txt 파일에 저장
- 2. Cnty_list.txt 파일의 내용을 화면에 문자열 단위로 출력

※실습하기



프로그램 14-4: 문자열 단위 파일 입출력 (주요 코드)

```
char fname[] = "Cnty list.txt", temp[50] = \{ '\0' \};
FILE *fp = fopen(fname, "w"); // 출력 파일 열기
for (i = 0; i < 4; i++) {
  fgets(temp, 50, stdin); // 문자열을 입력받아 temp에 저장
  fputs(temp, fp);
                // temp에 저장된 문자열을 파일에 출력
fclose(fp);
                      // 출력 파일 닫기
fp = fopen(fname, "r"); // 입력 파일 열기
for (i = 0; i < 4; i++) {
  fgets(temp, 50, fp); // 입력파일로부터 문자를 읽어 temp에 저장
  printf("%s", temp); // temp에 저장된 문자열을 모니터로 출력
fclose(fp);
                       // 입력 파일 닫기
```



3. 텍스트 파일 입출력

파일의 끝확인하기: feof()

파일의 끝까지 데이터를 모두 읽어 들인 상태인지를 확인

함수 원형	<pre>int feof(FILE *fp);</pre>					
함수 인자	fp 파일 포인터 변수명					
반환 값	 ✓ 파일의 끝이면 → 0이 아닌 값을 반환 ✓ 파일의 끝이 아니면 → 0를 반환 					

```
FILE *fp = fopen("test.txt", "r");
char str[50]= {'\0'};

fgets(str, sizeof(str), fp); // 데이터 읽은 후, feof 호출
while(!feof(fp)) { // 파일 끝이 아닌 동안
printf("%s", str);
fgets(str, sizeof(str), fp);
}
```



- EOF (End Of File)란?
 - 파일의 끝을 표현하기 위해 정의해 놓은 상수 (-1)
 - EOF도 파일의 내용으로 간주되므로 주의
- EOF vs. feof() 함수
 - 모든 파일의 끝에는 EOF가 존재 → EOF는 파일의 한 부분임
 - feof() 함수는 EOF를 읽으면, **0을 반환!**
 - 그이후에 0이 아닌 값을 반환!
 - ➤ 파일로부터 데이터를 먼저 읽은 후에 feof() 함수 사용 앞 슬라이드의 예제 및 프로그램 14.3 참조



하습 정리

- fscanf()와 fprintf() 함수는 형식을 지정하여 사용하는 파일 입출력 함수
- **fgetc()**와 **fputc()**는 문자 단위 파일 입출력 함수
- fgets()와 fputs()는 문자열 단위 파일 입출력 함수
- 파일의 끝을 확인하기 위해서는 feof() 함수를 이용

