

파일 입출력 라이브러리(응용)

학습내용

- 텍스트 파일 입출력
- 이진 파일 입출력

학습목표

- 형식화된 데이터를 파일 입출력으로 처리하는 방법을 구현할 수 있다.
- 이진 파일 입출력 처리의 기본 문법에 대해 설명할 수 있다.





fscanf

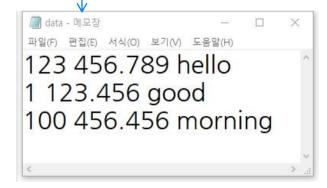
항목	내용
함수원형	int *fscanf(FILE *stream, const char *format,);
헤더	stdio.h
기능	형식화된 문자열을 파일로부터 입력
매개변수	FILE *stream 🏓 읽고자 하는 FILE 포인터
	const char *format 🏓 읽어들일 데이터 서식
반환값	읽기에 성공했다면 읽어들인 항목 개수를 반환하고 실패나 오류가 발생하면 -1을 반환

변환문자열	내용
%0	8진 정수 형식으로 입력
%d	10진 정수 형식으로 입력
%ld	long형 10진 정수 형식으로 입력
%x	16진 정수 형식으로 입력
%f	float형 형식으로 입력
%lf	double형 형식으로 입력
%с	문자 형식으로 입력
%s	문자열 형식으로 입력



```
#include <stdio.h>
int main(void)
  FILE
         *fp;
         buf[1024];
  char
 int
         num;
 float
         real:
 if ( fp = fopen( "data.txt", "r"))
   while(0 < fscanf(fp, "%d %f %s", &num, &real, buf))
     printf("%d %f %s\n", num, real, buf);
   fclose(fp);
 return 0;
}
```

data.txt



123 456.789001 hello 1 123.456001 good 100 456.455994 morning



rewind

항목	내용
함수원형	void rewind(FILE *stream);
헤더	stdio.h
기능	파일의 읽기/쓰기 위치를 파일의 처음 위치로 초기화
매개변수	FILE *stream ▶ 대상 파일 스트림
반환값	void

```
#include \( \stdio.h \)
#include \( \stdio.h \)

int main () {
    char str1[10], str2[10], str3[10];
    int year;
    FILE * fp;

    fp = fopen ("file.txt", "w+");
    fputs("We are in 2020", fp);

rewind(fp);
    fscanf(fp, "%s %s %s %d", str1, str2, str3, &year);
```

```
printf("Read String1 |%s|₩n",
str1 );
printf("Read String2 |%s|₩n",
str2 );
printf("Read String3 |%s|₩n",
str3 );
printf("Read Integer |%d|₩n",
year );
fclose(fp);
return(0);

Read String1 |We|
Read String2 |are|
Read String3 |in|
Read Integer |2020|
```



3 fprintf

항목	내용
함수원형	<pre>int *fprintf(FILE *stream, const char *format,);</pre>
헤더	stdio.h
기능	형식화된 문자열을 파일로 출력
매개변수	FILE *stream 🔷 출력하고자 하는 FILE 포인터
	const char *format 🏓 출력할 데이터 서식
반환값	출력에 성공하면 출력한 문자열을 반환하고 실패나 오류가 발생하면 -1을 반환

형식 문자열의 구조

%[*] [width] [{h | 1 | L}]type





fprintf

flags	의미	디폴트 값	사용 예	출력 결과
-	전체 폭에 대하여 왼쪽으로 정렬	오른쪽으로 정렬	printf("%-10d", 123);	123□□□□□□
+	부호 있는 값에 대하여 +/-를 출력	음수에 대해서만 -를 출력	printf("%+d", 123);	+123
빈칸	양수만 부호 자리에 빈칸을 출력	빈칸을 출력하지 않음	printf("% d", 123); printf("% d", 123);	□123 -123
0	정수의 폭에 맞춰서 0으로 채움	0으로 채우지 않음	printf("%010d", 123);	0000000123
#	o, x, X와 함께 사용되면 0, 0x, 0X를 함께 출력	0, 0x, 0X를 출력하지 않음	printf("%#x", 123);	0x7b





fprintf

```
#include <stdio.h>
int main ()
 FILE * pFile;
 int n;
 char name [100];
 pFile = fopen ("myfile.txt", "w");
 for (n=0; n<3; n++)
  puts ("please, enter a name: ");
   gets (name);
 fprintf (pFile, "Name %d [%-10.10s]₩n",n+1,name);
 fclose (pFile);
 return 0;
                                    · 명령 프롬프트
}
                                     Name 1 [kim
                                     Name 2 [choi
                                     Name 3 [park
```





fread

항목	내용
함수원형	size_t fread(void *ptr, size_t size, size_t nitems, FILE *stream);
헤더	stdio.h
기능	이진(Binary) 파일로부터 입력
	void *ptr → 파일 내용을 읽어들일 메모리 포인터
ᇚᆘᆌᅛ	size_t size 🏓데이터 하나의 크기
매개변수	size_t nitems 🏲 읽을 데이터의 개수
	FILE *stream 📦 대상 파일 스트림
반환값	읽어들인 데이터 개수를 반환, 오류가 발생하면 -1을 반환

```
#include \( \stdio,h \)
#include \( \stdlib,h \)

int main () {
    FILE * pFile;
    char buffer[500];
    size_t result;

pFile = fopen ( "myfile.bin", "rb" );
    if (pFile==NULL) {fputs ("File error", stderr); exit (1);}

fread (buffer,1,|Size,pFile);
    ........

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





fwrite

항목	내용
함수원형	size_t fwrite(void *ptr, size_t size, size_t nitems, FILE *stream);
헤더	stdio.h
기능	파일에 버퍼 내용을 출력
매개변수	void *ptr ▶파일 내용에 출력할 메모리 포인터 size_t size ▶ 데이터 하나의 크기 size_t nitems ▶ 쓰기 할 데이터의 개수 FILE *stream ▶ 대상 파일 스트림
반환값	출력한 데이터 개수를 반환, 오류가 발생하면 -1을 반 환



fwrite

```
#include \( \stdio.h \)
enum \{ SIZE = 5 \};
int main(void)
{
    double a[SIZE] = \{1.,2.,3.,4.,5.\};
    FILE *fp = fopen("test.bin", "wb"); // must use binary mode
    fwrite(a, 8, SIZE, fp); // writes an array of doubles
    fclose(fp);
```

```
double b[SIZE];
                                                                intmain(void)
 fp=fopen("test,bin","rb");
 size tret code=fread(b, 8, SIZE, fp); // reads
an array of doubles
 if(ret code = SIZE){
    puts("Array read successfully, contents: ");
   for(int n = 0; n < SIZE; ++n) printf("%f".
b[n]);
    putchar('₩n');
 }else{//error handling
   if (feof(fp))
    printf("Error reading test, bin: unexpected
end of file\hbar n");
   else if (ferror(fp)) {
     perror("Error reading test,bin");
                                                                  fclose(fp sour);
                                                                  fclose(fp dest);
                                                                  return 0;
 fdose(fp);
```

```
intmain(void)
{

FILE *fp_sour; //파일원본을위한파일스트림포인터
FILE *fp_dest; //복사대상을위한파일스트림포인터
char buff[1024]; //파일내용을읽기/쓰기를위한버퍼
size_t n_size; //읽거나쓰기를위한데이터의개수

fp_sour =fopen("/main.bd", "\");
fp_dest =fopen("/main.bd", "\");

while(0<(n_size=fread(buff, 1, 1024, fp_sour)))
{
fwrite(buff, 1, n_size, fp_dest);
}

fdose(fp_sour);
fdose(fp_dest);
return 0;
}
```





fseek

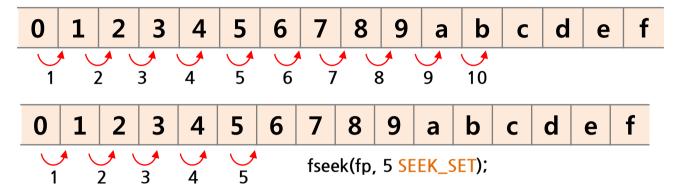
항목	내용
함수원형	int fseek(FILE *stream, long offset, int whence);
헤더	stdio.h
기능	파일의 읽기/쓰기 위치를 이동
	FILE *stream → 대상 파일 스트림
매개변수	long offset > 이동할 바이트 수
	int whence → 시작 시점
반환값	함수 실행이 성공하면 0을 반환, 오류가 발생하면 0이 아닌 값을 반환

whence	설명
SEEK_SET	파일의 시작
SEEK_CUR	현재 읽기/쓰기 포인터 위치
SEEK_END	파일의 끝





fseek(fp, 10, SEEK_SET);



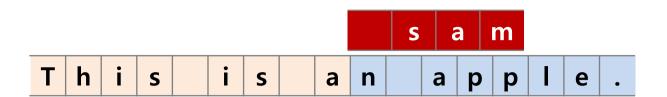
1 2 3 4 5 fseek(fp, 5 SEEK_CUR);

```
0 1 2 3 4 5 6 7 8 9 a b c d e f

fseek(fp, -5 SEEK_END); -5 -4 -3 -2 -1
```

```
#include \( \stdio.h \)
int main ()
{
    FILE * pFile;
    pFile = fopen ( "example.txt" , "wb" );
    fputs ( "This is an apple." , pFile );
    fseek ( pFile , 9 , SEEK_SET );
    fputs ( " sam" , pFile );
    fclose ( pFile );
    return 0;
}
```





항목	내용
함수원형	long ftell(FILE *stream);
헤더	stdio.h
기능	파일의 읽기/쓰기 위치를 확인
매개변수	FILE *stream ➡대상 파일 스트림
반환값	읽기/쓰기 위치를 반환, 오류가 발생하면 -1을 반환

4 ftell

```
📆 명령 프롬프트
                                        pointer = 0
                                                          char = #
intmain(void)
                                        pointer = 1
                                                          char = i
                                        pointer = 2 char = n
 FLE *fp;
 char chr_byte;
                                        pointer = 3
                                                          char = c
 int ndx:
                                        pointer = 4
                                                          char = I
fp=fopen("/main.c", "r");
for(ndx=0;ndx<5;ndx++)
 printf("pointer=%d", ftell(fp));
 printf(" ");
 printf("char=%cn", fgetc(fp));
fdose(fp);
 retum 0;
```

학습정리

1. 텍스트 파일 입출력

- •형식화된 문자열을 이용한 파일 입출력은 fscanf, fprintf를 이용
 - int *scanf(FILE *stream, const char *format, ...);
 - int *fprintf(FILE *stream, const char *format, ...);
- •rewind는 파일의 읽기/쓰기 위치를 처음 위치로 이동임

2. 이진 파일 입출력

- •이진 파일의 입출력은 fread, fwrite를 이용
- •fseek는 지정한 거리만큼 포인터를 이동
- •ftell은 현재 포인터의 위치를 반화