

# 파일 입출력 라이브러리(기초)

### 학습내용

- 파일 입출력 기초
- 파일 입출력 함수

### 학습목표

- 스트림의 기본 개념에 대해 설명할 수 있다.
- 파일 입출력을 위한 함수의 기본 문법을 알고 구현할 수 있다.

# 파일 입출력 기초

1 기초

파일 입출력

입력과 출력의 대상이 파일인 경우

입출력 대상

텍스트 파일, 이진 파일

■ 라이브러리 함수

구분	텍스트 파일	이진 파일
입력	fgetc, fgets, fscanf	fread
출력	fputc. fputs, fprintf	fwrite
닫기	fclose	

### 2 입출력 과정

스트림 생성

- 파일 입출력을 위한 스트림을 생성, 파일 포인터
- fopen

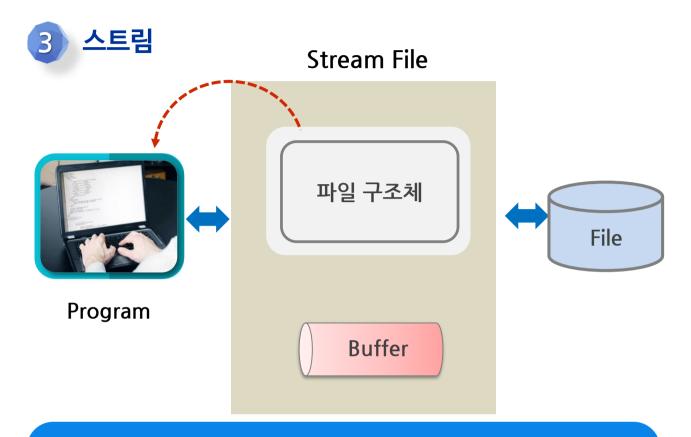
입출력

- fgets, fgetc, fputs
- fread, fwrite

스트림 닫기

fclose

# 파일 입출력 기초



### 운영체제에 의해 프로그램 실행 시 기본적으로 개방

스트림파일	용도	장치
stdin	표준 입력 스트림	키보드
stdout	표준 출력 스트림	모니터
stderr	표준 에러 스트림	모니터



- 파일 열기
  - 파일 입출력을 수행하려면 먼저 fopen 함수를 이용하여 파일을 열어야함
  - fopen 함수의 원형

#### FILE \*fopen(const char \*filename, const char \*mode);

filename : 파일 이름
 mode : 파일 열기 모드

• 리턴 값: 생성된 파일 스트림의 파일 포인터를 리턴, 파일을 열 수 없으면 널 포인터를 리턴

#### 디폴트 모드는 텍스트 파일

열기 모드	의미
" <b>r</b> "	• 입력용파일을 연다.파일이 없으면 열기 실패이다.
"w"	• 출력용파일을 연다.
	• 파일이 없으면 새로 생성하고, 기존 파일이 있으면 파일의 내용이 모두 사라진다.
"a"	• 파일의끝에 출력하는용도로 파일을 연다.
	• 파일이 없으면 새로 생성한다.
	• 파일의 어떤 위치도 읽을 수 있지만, 쓰기는 파일의 맨끝에 만할 수 있다.
"r+"	• 입출력용파일을 연다.파일이 없으면 열기 실패이다.





### 파일 열기

열기 모드	의미
"w+"	<ul> <li>입출력용 파일을 연다.</li> <li>파일이 없으면 새로 생성하고 기존 파일이 있으면 파일의 내용이 모두 사라진다.</li> </ul>
"a+"	<ul> <li>읽고 추가하는 용도로 파일을 연다. 파일이 없으면 새로 생성한다.</li> <li>파일의 어떤 위치도 읽을 수 있지만, 쓰기는 파일의 맨끝에 만할 수 있다.</li> </ul>
"t"	• 텍스트 모드에서 입출력을 수행한다.
"b"	• 바이너리 모드에서 입출력을 수행한다.

# ② 파일 닫기

- 1 파일 입출력이 끝나면 fdose 함수를 호출해서 반드시 파일을 닫아야 함
- 2 fopen 함수의 원형

#### int fclose(FILE \*stream);

• stream : 파일 포인터

• 리턴 값: 파일 닫기가 성공하면 0을 리턴하고,실패하면 EOF(-1)을

리턴





### 파일 입출력

텍스트 파일 입출력 함수	의미
int fgetc(FILE *fp);	파일에서 한 문자를 입력받는다.
int fputc(int c, FILE *fp);	파일로 한 문자를 출력한다.
<pre>char *fgets(char *str, int n, FILE *fp);</pre>	파일에서 한 줄의 문자열을 입력받는 다.
<pre>int fputs(const char *str, FILE *fp);</pre>	파일로 한 줄의 문자열을 출력한다.
int fscant(FILE *fp, ···);	형식 문자열을 이용해서 파일에서 입 력받는다.
int fprint(FILE *fp, ···);	형식 문자열을 이용해서 파일로 출력 한다.

#### 1 fgetc, fputc

### 파일로부터 문자 하나를 입력받거나 출력할 때 사용됨

int fgetc(FILE \*stream);
int fputc(int c, FILE \*fp)



#### 1 fgetc, fputc

```
#indude \( \stdio,h \)
int main(void)
{
    FILE *fp;
        char ch;
    if (fp = fopen("/main.c", "i"))
    {
        while(EOF!=(ch = fgetc(fp)))
        putchar(ch);
    fdose(fp);
    }
    return 0;
}
```

```
#include \( \stdio.h \)
int main ()
{
   FILE * pFile;
   int c;
   int n = 0;
   pFile=fopen ("myfile.txt", "r");
   if (pFile==NULL) printf("Error opening file");
   else
   {
}
```

```
do {
    c = fgetc (pFile);
    if (c == '$') n++;
    } while (c != EOF);
    fclose (pFile);
    printf ("The file contains %d dollar sign characters ($).₩n",n);
    }
    return 0;
}
```

- ③ 파일 입출력
  - 2 fgets, fputs
  - 1 파일로부터 한 줄의 문자열을 입력받거나 출력할 때 사용

char \*fgets(char \*str, int n, FILE \*stream);
 int fputs(const char \*str, FILE \*fp)

2 fgets

용어	설명
str	<ul> <li>읽은 문자열을 저장할 문자 배열의 주소</li> <li>char *fgets(char *str, int n, FILE *stream);</li> </ul>
n	• 자 배열의 크기
stream	• 입력용 파일 포인터
리턴 값	• 파일로부터 문자열을 최대 n-1개만큼(또는 줄 바꿈 문자까지) 읽어서 str에 저장
	• 에러를 만나거나 EOF를 만나면 NULL을 리턴



2 fgets, fputs

3 fputs

용어	설명
str	<ul><li>저장할 문자열</li><li>int fputs(const char *str, FILE *fp)</li></ul>
fp	• 출력용 파일 포인터
리턴 값	• 저장한 문자 수

```
#include <stdio.h>
                                          if (fp_sour = fopen("./main.c", "r"))
int main(void)
                                             if (fp_dest = fopen("./backup.c", "w"))
  FILE
          *fp_sour;
                                              buf = malloc(buf_size+5);
  FILE
           *fp_dest;
  char
           *buf;
                                              while(fgets(buf, buf_size, fp_sour))
          buf_size;
  int
                                                fputs( buf, fp_dest);
  buf size = 1024;
                                              free(buf);
                                              fclose(fp_dest);
                                             fclose(fp_sour);
                                           return 0;
                                          }
```



### 파일 입출력

#### 2 fgets, fputs

```
#indude \( \stdio,h \)
int main()
{
    FILE * pFile;
    charmystring [100];

pFile = fopen ("myfile.txt", "i");
    if (pFile == NULL) printf ("Error opening file");
    else{
        if (fgets (mystring, 100, pFile)!= NULL)
            puts (mystring);
        fdose (pFile);
    }
    return 0;
}
```



1 ferror 함수

#### 파일 입출력 시 발생하는 에러를 확인

#### int ferror(FILE \*stream);

- 리턴 값:
  - 스트림에 에러가 발생했으면 0이 아닌 값을 리턴
  - 에러가 발생하지 않았으면 0을 리턴

#### 2 feof 함수

#### 파일의 끝인지를 검사

#### int feof(FILE \*stream);

- 리턴 값:
  - 읽는 위치가 파일의 끝이면 0이 아닌 값을 리턴
  - 읽는 위치가 파일의 끝이 아니면 0을 리턴

### 학습정리

# 1. 파일 입출력 기초

- •파일 입출력 시 먼저 스트림을 생성해야 함
- •파일스트림 생성 시 읽기, 쓰기, 추가 등의 모드를 지정
- •파일을 열고 닫는 라이브러리 함수는 fopen, fclose임

### 2. 파일 입출력 함수

- •파일로부터의 입력 함수로는 fgetc, fgets, fscanf, fread 등이 있음
- •파일에 출력하는 함수는 fputc, fputs, fprintf, fwrite 등이 있음
- •파일의 끝을 판단하는 함수는 feof가 있음