

# Standard I/O and File I/O

**Section 1** 표준 입출력 함수

**Section 2** 파일 입출력 함수



# 표준 입출력 함수

## ● 표준 입력

- stdin, standard input: 키보드로 입력
- scanf: 키보드에 입력이 들어왔는지 scan한다 f(지정 포맷)에 맞춰서
- gets: 키보드로부터 문자열(s: string)을 가져온다(get) → gets

겟츠라고 읽지마

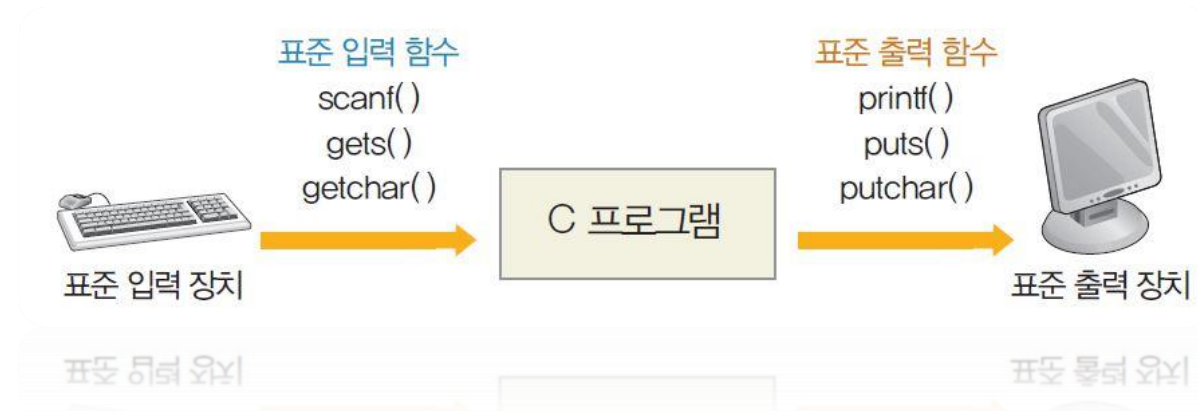
↳ get string

## ● 표준 출력

- stdout, standard output: 모니터로 출력
- printf: 모니터로 출력을 내보낸다, f(지정 포맷)에 맞춰서
- puts: 모니터로 문자열(s: string)을 가져다 둔다(put) → puts

풋츠라고 읽지마

↳ put string



# 서식이 지정된 입출력 함수

## ● printf(), scanf()

구문	설명
printf("서식", 출력할 매개변수들 ...)	표준 출력 장치(모니터)에 서식을 맞춰 출력한다.
scanf("서식", 입력할 매개변수들 ...)	표준 입력 장치(키보드)에서 서식에 맞춰 입력받는다.

scanf: 문자열만 받음.

## ● 서식의 위치에 올 수 있는 것들

서식	설명
%d	정수형(int)
%c	문자형(char)
%s	문자열(char*) 또는 문자 배열(char[])
%x	16진수 정수(int)
%o	8진수 정수(int)
%f, %lf	실수형(float, double)
%e	공학 계산용 형식
%p	포인터 주소

-b 참고: Regular Expression

## 문자열 입출력 함수

### ● 문자열 입출력 함수

- #include <conio.h> 필요
- printf( ), scanf( ) 함수: 모든 데이터 형식의 입출력 가능

- puts( ), gets( ) 함수: 문자열만을 취급

*gets -s*

구문	설명
gets(문자 배열)	표준 입력 장치(키보드)로부터 '문자 배열 크기 - 1'만큼 문자열을 입력받는다. 숫자를 입력해도 무조건 문자열로 취급한다.
puts(문자 배열)	표준 출력 장치(모니터)에 문자열을 출력한다.

- 문자열 입출력시에는 puts, gets가 편리
  - 서식 지정 없이 변수명만 지정해주면 되니까
  - 풋 스트링, 겟 스트링이라고 읽어야 한다. → 풋츠, 겟츠 아니다

- 참고: 문자열만 입출력할 경우에는 printf( ), scanf( ) 함수보다 gets( ), puts( ) 함수를 사용하는 것이 처리 속도가 더 빠름.

# 문자 입출력 함수

## ● 문자 입출력 함수

- #include <conio.h> 필요
- getch( ), getche( ), getchar( ) 함수 : 문자 하나만 입력하는 기능
- putchar(), putch() 함수 : 문자 하나만 출력하는 기능

입력 함수	설명
getch( )	키보드를 통해 문자 하나를 입력받으며, 입력한 내용을 모니터에 보여주지 않는다.
getche( )	키보드를 통해 문자 하나를 입력받으며, 입력한 내용을 모니터에 보여준다.
getchar( )	사용자가 키보드로 <u>Enter</u> 를 누를 때까지 입력한 것을 메모리(버퍼)에 모두 저장해놓고( <u>Enter</u> 도 저장됨) 그중에서 한 문자만 꺼낸다.

출력 함수	설명
putchar(문자형 변수)	표준 출력 장치(모니터)에 문자 하나를 출력한다.
putch(문자형 변수)	putchar( )와 기능이 동일하다.

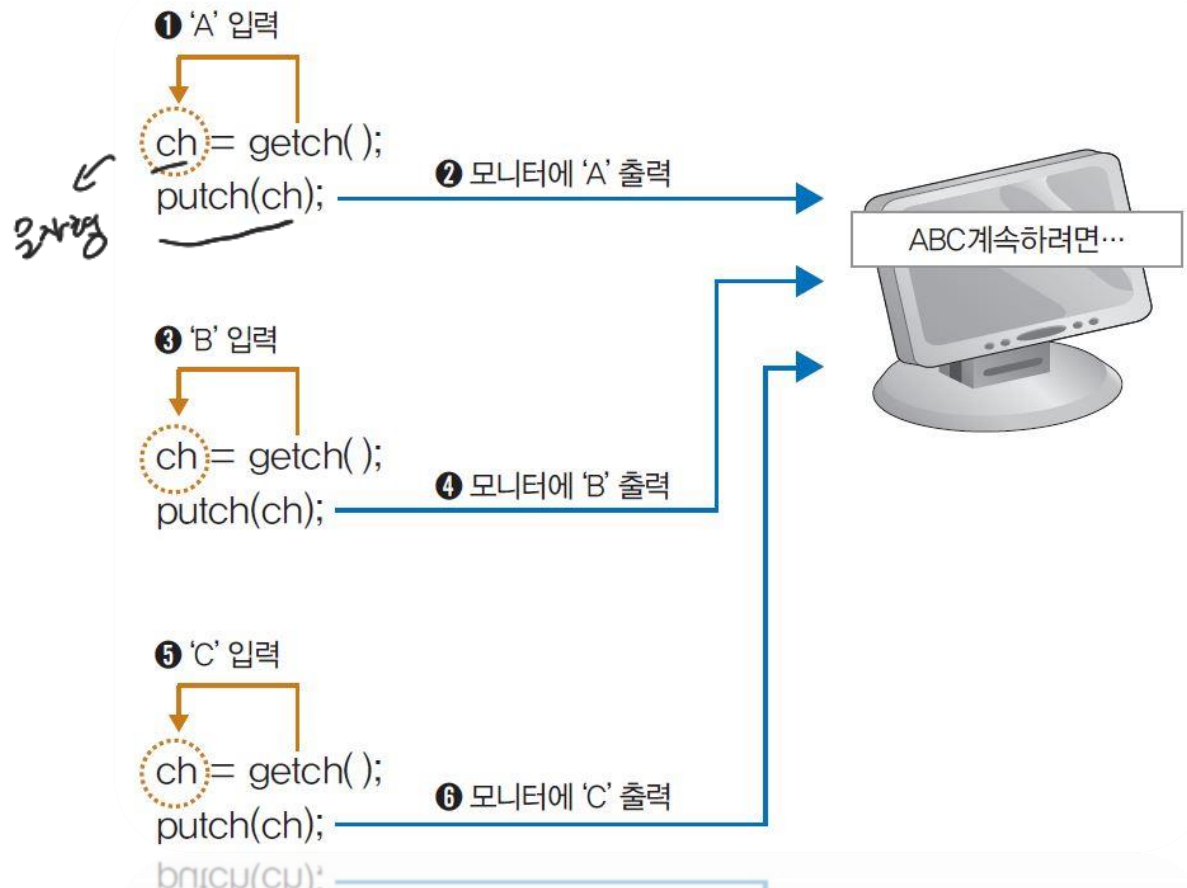
- getch(갯 캐릭터), putch(풋 캐릭터) → 갯치, 풋치 아니다.
- getche(갯 캐릭터 에코) → 갯체 아니다.
- getchar(갯 캐릭터 awaits return) → awaits return: 리턴(엔터)를 기다린다.

*getch( )*

# 편리한 문자 입력을 위한 함수들

## ● getch( )

- 키보드를 통해 문자 하나를 입력 받으며 입력한 내용을 화면에 보여주지 않음
- 입력 내용의 확인을 원할 경우 putch()와 함께 사용해야 함
- 명령 프롬프트와 같은 문자 입력 인터페이스에서 암호 입력에 매우 적합



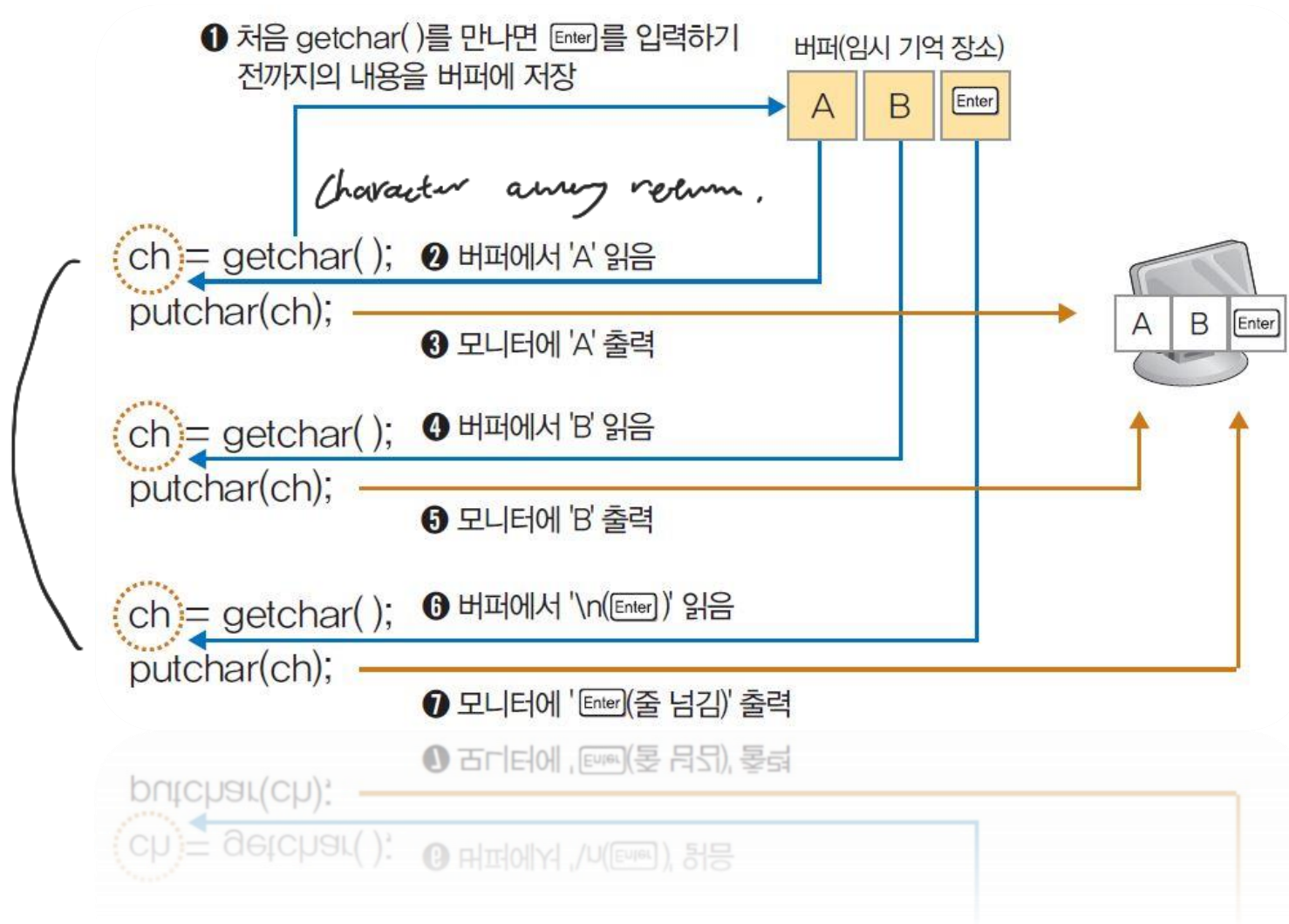
## 편리한 문자 입력을 위한 함수들

### ● getche( )

- Get Character **Echo** 함수
- **Echo**: 컴퓨터 공학 용어로, 내가 들은 말을 상대방에서 똑같이 하는 것:  
군대 용어로, 복명복창
- getche( ) 함수는 putchar( ) 함수를 사용하지 않아도 입력한 글자를 바로 화면에 출력함.
- 입력할 글자를 화면에 나타내려면 putchar( )나 putchar( )를 함께 사용해야 하는 getch( ) 대신 getche( ) 함수를 사용해야 간결함.

## 편리한 문자 입력을 위한 함수들

- **getchar( )**



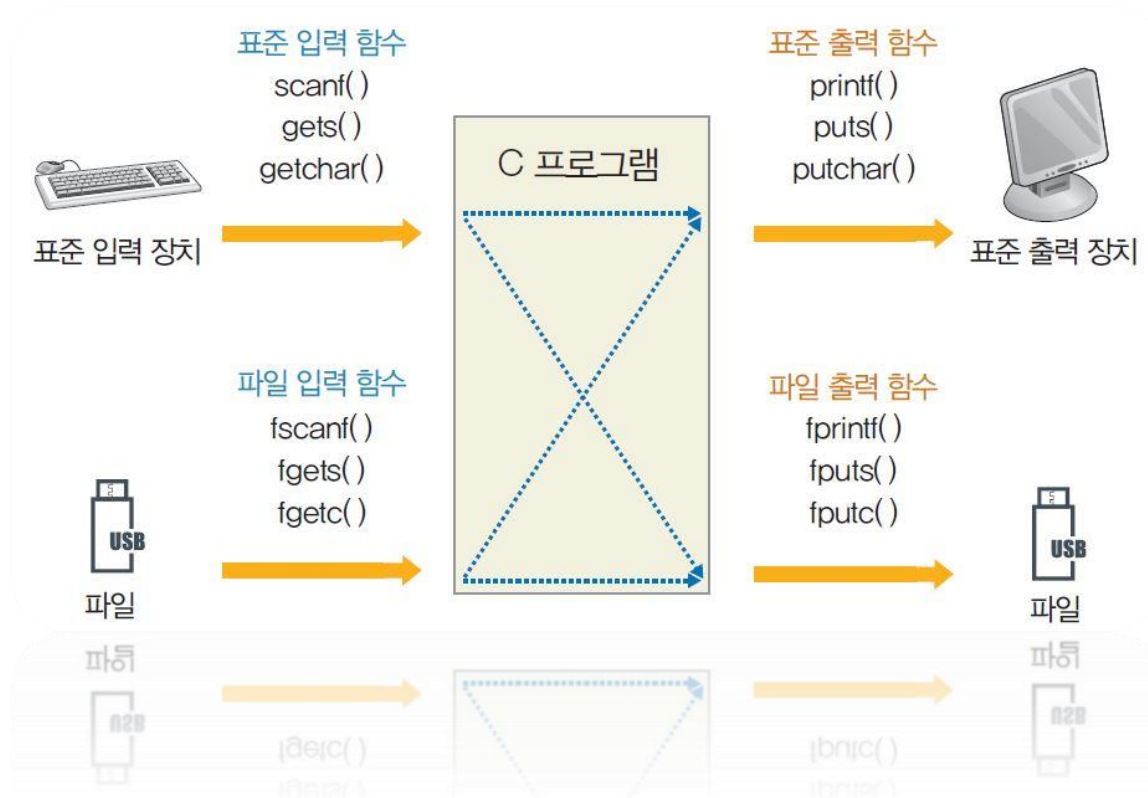


## 2 파일 입출력

# 파일 입출력 함수

### ● 표준 입출력과 파일 입출력 함수

- 사용하는 함수와 입출력 관련 장치가 다름



## 파일 입출력의 과정

- 1단계: 파일 포인터 선언하기

```
FILE *변수 이름;
```

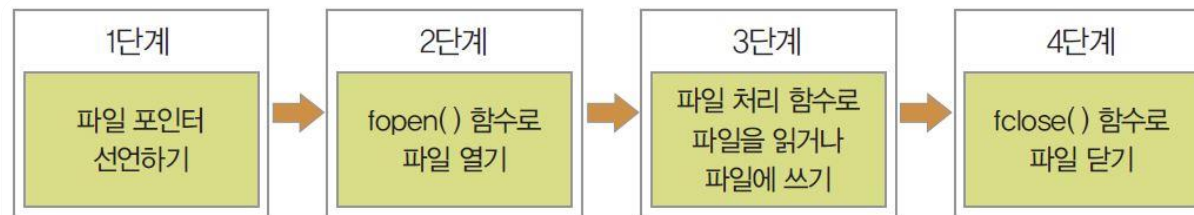
- 2단계: `fopen( )` 함수로 파일 열기

```
변수 이름 = fopen("파일 이름", "열기 모드");
```

- 3단계: 파일 처리 함수로 파일을 읽거나 파일에 쓰기

- 4단계: `fclose( )` 함수로 파일 닫기

```
fclose(파일 포인터);
```



## 파일을 이용한 입력

### ● 파일의 문자열 읽기 : fgets( )

- 파일로부터 값을 입력받을 때 사용.
- 파일 포인터에 지정된 파일에서 문자열을 읽어 문자 배열에 대입함.
- 읽어올 문자열의 최대 길이는 '읽을 최대 문자수 -1'

```
fgets(문자 배열, 읽을 최대 문자 수, 파일 포인터);
```



### ● 파일을 통해 데이터를 입력한 후 이를 모니터에 출력하는 프로그램 작성하기.

- 메모장 실행 → 'File Read Sample' 문장 입력 → 'C:\temp\data1.txt'로 저장



## DOS 명령어 type의 구현

- **type** : ~~지정한 파일의 내용을 화면에 출력하는 기능~~

type 파일 이름

- [시작] → [실행]을 선택 후 'cmd' 명령을 입력하여 명령 프롬프트를 연다.
- 다음 명령어 입력

type C:\windows\win.ini

```

C:\> type c:\windows\win.ini
; for 16-bit app support
[fonts]
[extensions]
[mci extensions]
[files]
[Mail]
MAPI=1

C:\>
  
```

```

C:\>
  
```

cd 경로  
경로 찾기.

dir 해당 경로의 파일을  
보여줌

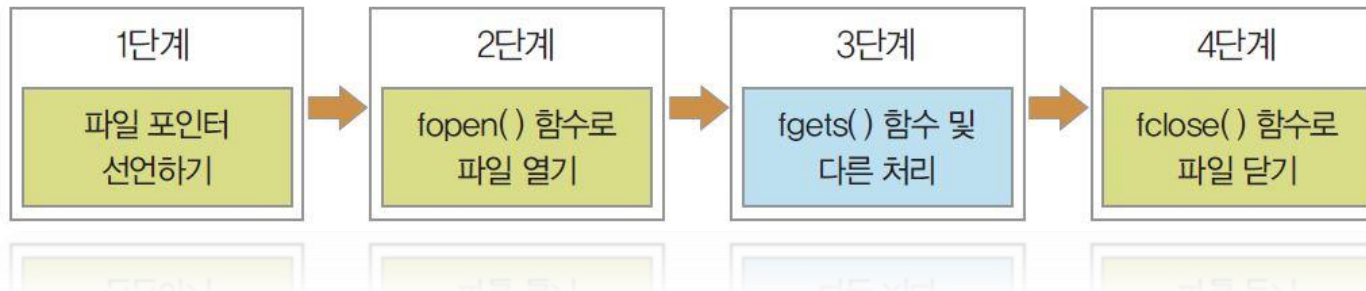
# DOS 명령어 type의 구현 - 파일 입력 기능 구현

## ● 서식을 지정하여 파일 읽기 : fscanf( )

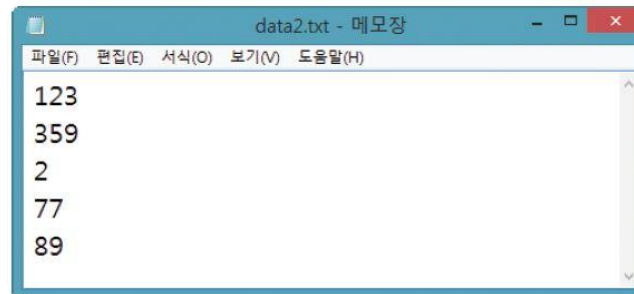
- 파일 포인터를 사용하는 것을 제외하고 scanf( )와 사용법이 동일

```
fscanf(파일 포인터, "서식", 입력할 매개변수들 ...);
```

- 'C:\temp\data2.txt' 파일에 정수 다섯 줄을 쓰고 fscanf( ) 함수로 읽어온 후 그 숫자들을 합하는 프로그램의 작성 과정



먼저 5줄의 숫자를 메모장에 적고 'C:\temp\data2.txt'로 저장함.



strcmp

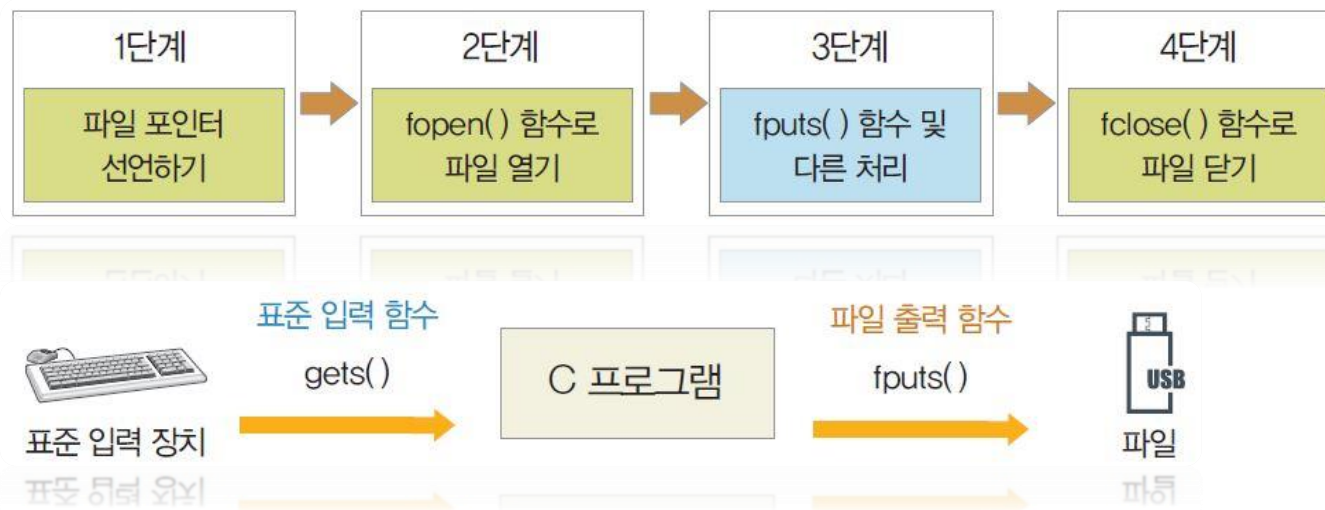
# DOS 명령어 type의 구현 - 파일 출력 기능 구현

## ● 파일의 문자열 출력 : fputs( )

- 파일에서 데이터를 읽어와 모니터에 출력하는 대신 파일에 내용을 씀
- 단, 파일 포인터에 지정된 파일에 문자열을 출력

fputs(출력할 데이터, 파일 포인터);

W → 덮어쓰기 모드임.



# DOS 명령어 copy의 구현

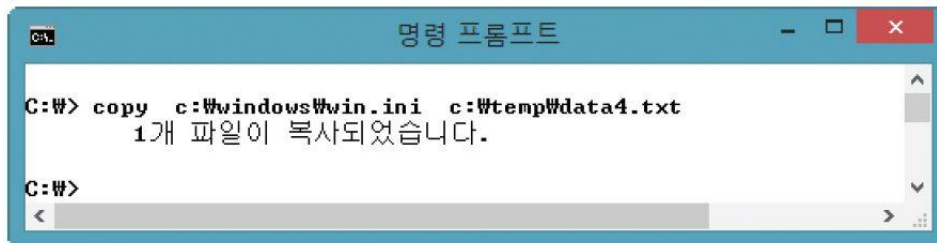
- **copy** : 주어진 파일을 복사하여 똑같은 파일을 하나 더 만드는 명령어

```
copy 소스_파일 타겟_파일
```

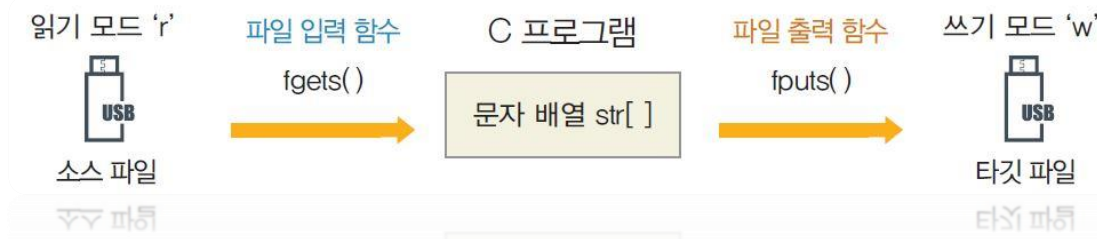
- [시작] → [실행]을 선택한 후 'cmd' 명령을 입력
- 명령 프롬프트 창을 열고 다음 명령어 입력

```
copy C:\windows\win.ini C:\temp\data4.txt
```

복사 속도가 상당히 빠름



- 파일을 이용한 출력 과정



## DOS 명령어 copy의 구현

### ● 서식을 지정하여 파일 출력 : fprintf( )

- 파일에 숫자를 출력할 때는 서식을 지정할 수 있는 fprintf( ) 함수를 사용하는 것이 편리함
- printf( ) 함수와 사용법이 동일하지만, 파일 포인터를 지정한다는 점이 다름

```
fprintf(파일 포인터, "서식", 출력할 매개변수들 ...);
```

*fputs랑 같음. 리스*



#include <stdio.h> #include <conio.h>

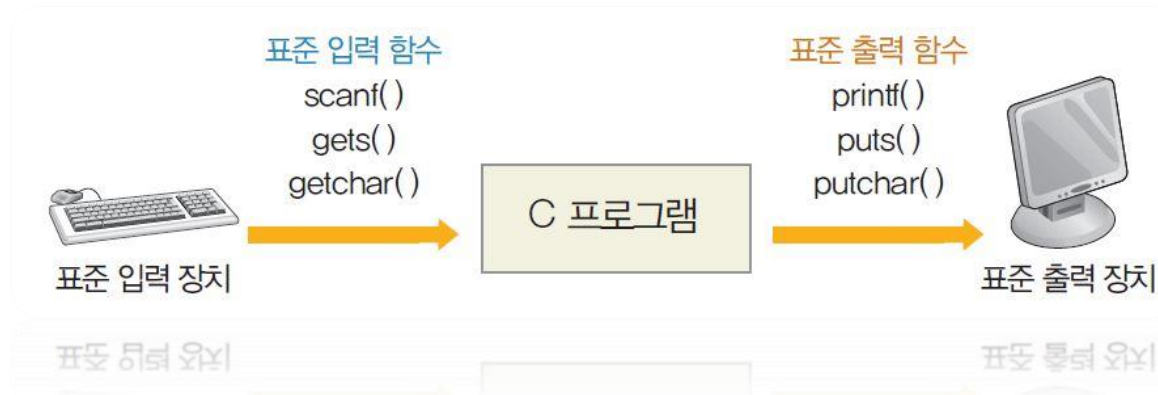
## 1. 표준 입출력 함수

### ① 키보드로 입력하는 것을 '표준 입력'이라 하며

표준 입력 함수에는 scanf( ), gets( ), getchar( ) 등이 있다.

### ② 모니터로 출력하는 것을 '표준 출력'이라 하며

표준 출력 함수에는 printf( ), puts( ), putchar( ) 등이 있다.

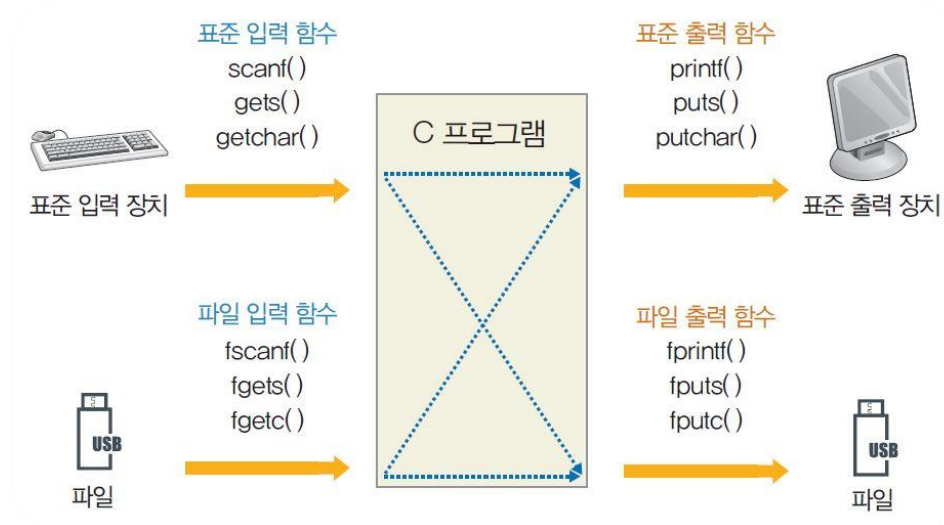


Standard IO  
vs  
Conio IO

# #include <file.h>

## 2. 파일 입출력 함수

- ❶ 키보드 대신 파일을 통해 입력받는 함수를 '파일 입력 함수'라고 하며  
파일 입력 함수에는 fscanf( ), fgets( ), fgetc( ) 등이 있다.
- ❷ 실행 결과를 모니터 대신 파일에 출력하는 함수를 '파일 출력 함수'라고 하며  
파일 출력 함수에는 fprintf( ), fputs( ), fputc( ) 등이 있다.



## 3. 파일 입출력의 기본 과정

