

# 13차시 - 파일(1)

## ◆ 지금까지의 main 함수

```
int main (void) {  
  
}
```

```
C:\> Program.exe ↵
```

## ◆ 프로그램 실행 시 인수를 넣어줄 수 있다.

- 함수 호출시의 인수 전달처럼 프로그램 실행에 인수를 넣는다.

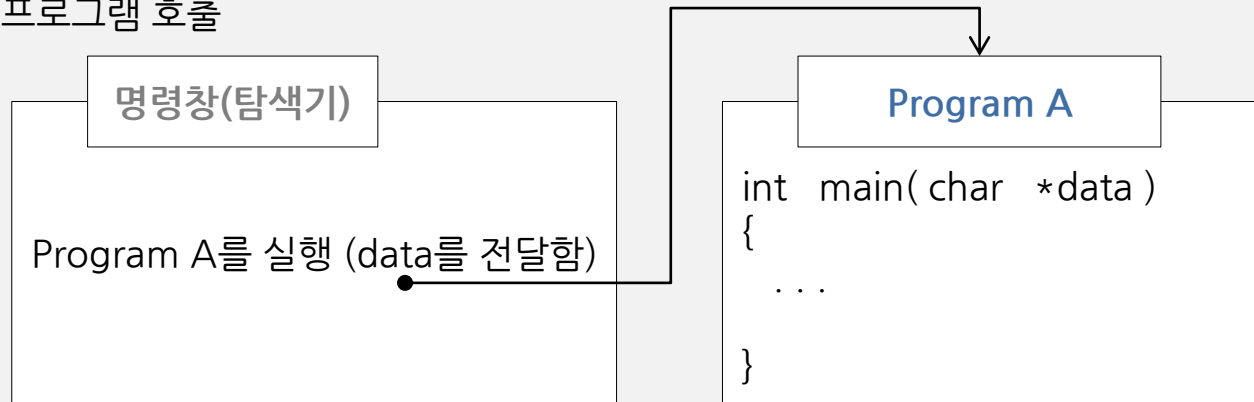
```
C:\> Program.exe param
```



프로그램에 “param”이라는 문자열을 전달

# 명령 인수

프로그램 호출



프로그램 호출



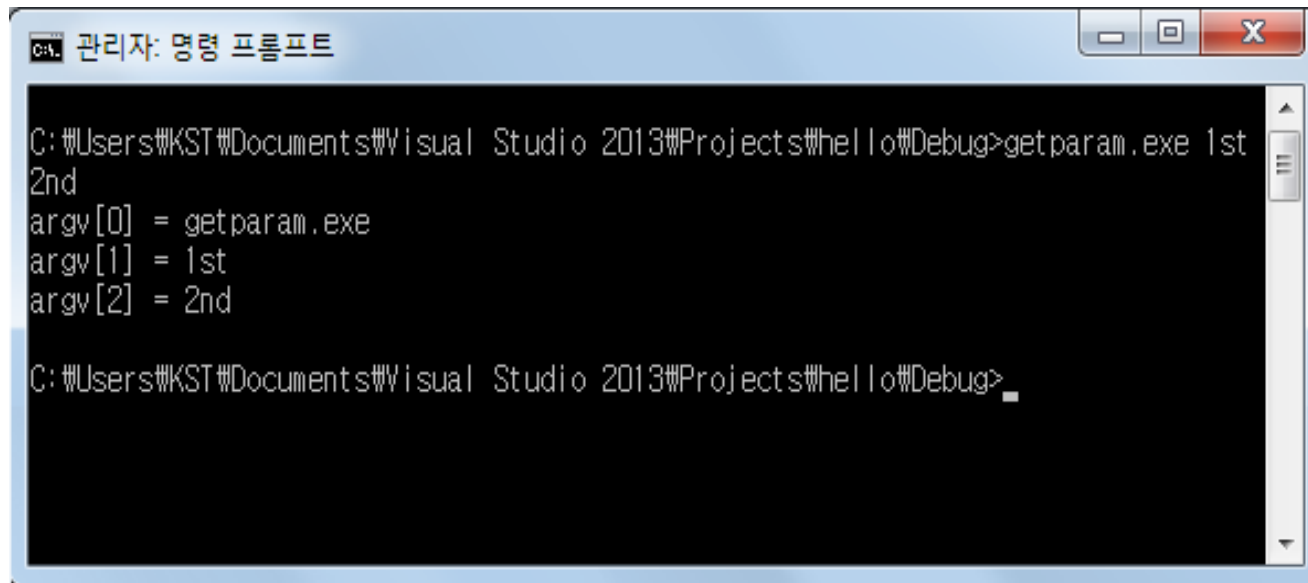
# main 함수의 형식

- ◆ `int main(int argc, char *argv[ ]) // =char **argv`
  - `argc(argument count)` : 인수의 개수
  - `argv(argument vector)` : 인수의 내용 (문자열이 여러 개 올 수 있다)
- ◆ `C:\> Program.exe myname 3 ↵`
  - `argc = 3`
  - `argv[0] = "Program.exe"`
  - `argv[1] = "myname"`
  - `argv[2] = "3" // 모두 문자열이다.`

# main 함수의 형식

```
int main(int argc, char *argv[])
{
    int i;

    for ( i = 0 ; i < argc ; i++ )
        printf("argv[%d] = %s\n" , i , argv [ i ] );
}
```



The screenshot shows a Windows Command Prompt window with the title '관리자: 명령 프롬프트'. The command prompt shows the following text:

```
C:\Users\KST\Documents\Visual Studio 2013\Projects\hello\Debug>getparam.exe 1st
2nd
argv[0] = getparam.exe
argv[1] = 1st
argv[2] = 2nd

C:\Users\KST\Documents\Visual Studio 2013\Projects\hello\Debug>
```

# 명령 인수의 사용 예

```
int main(int argc, char *argv[ ])
{
    int select;
    select = atoi(argv[1]);
    switch(select) {
        case 1 : // 1번 방식으로 실행됨
        case 2 : // 2번 방식으로 실행됨
    }
}
```

```
C:\> cardselect.exe 1
// 1번 방식으로 실행
```

```
C:\> cardselect.exe 2
// 2번 방식으로 실행
```

```
C:\> examresult.exe 1-1.dat
1학년 1반 성적을 처리하였습니다.
// 1-1.dat 파일을 가지고 실행
```

```
C:\> examresult.exe 1-2.dat
1학년 2반 성적을 처리하였습니다.
// 1-2.dat 파일을 가지고 실행
```

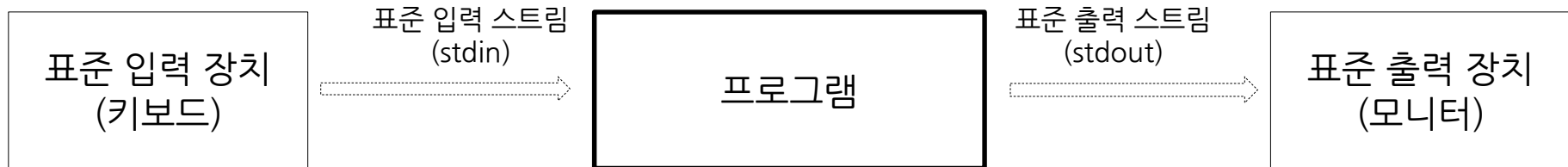
# 미니 과제

## ◆ 다음의 계산 프로그램을 작성하라

- 숫자와 부호 사이에는 공백이 있다. 숫자는 세 자리 이하이다.
- 사칙 연산 모두 가능하다.

```
C:\> calc.exe 2 + 3  
2 + 3 = 5
```

# 스트림과 표준 입출력



## ◆ 용어

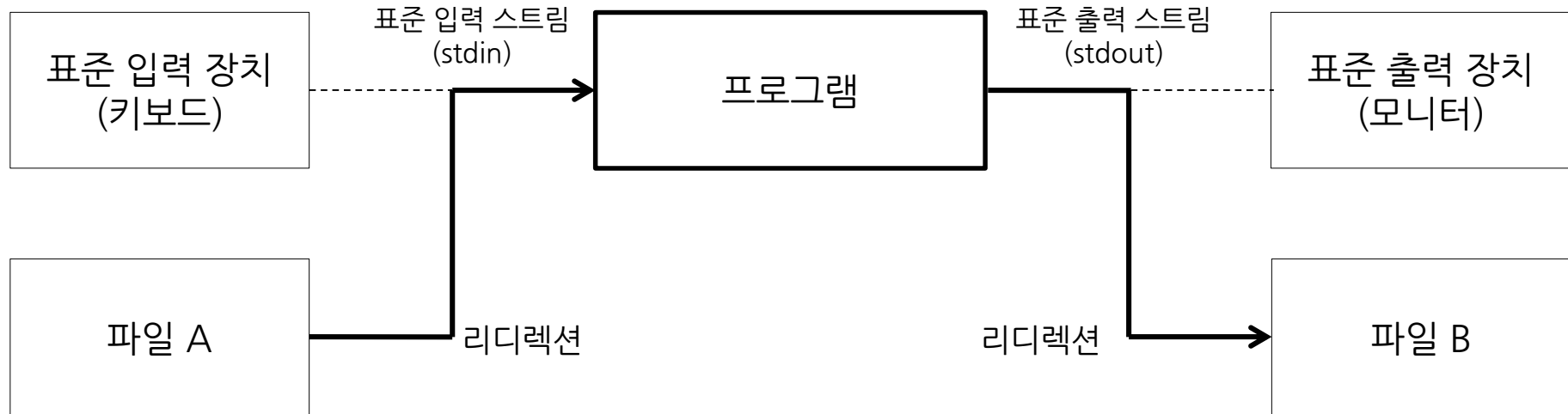
- 스트림(stream)
- 표준 입출력, 표준 입출력 장치
- 표준 입력 스트림(stdin), 표준 출력 스트림(stdout)

## ◆ 다른 장치와의 입출력

- 먼저 입출력 스트림을 생성한다.



# 리디렉션 (redirection)



## ◆ 표준 입출력을 파일로 연결하는 것

- C:\> program.exe < input.dat
- C:\> program.exe > output.txt
- C:\> program.exe < input.dat > output.txt