

# 21장 뜻자와 뜻자역 관련함수

## 차례

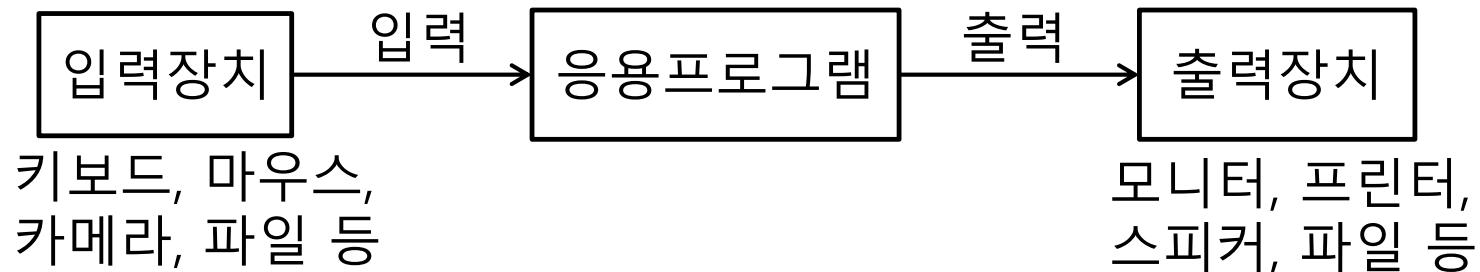
1. 스트림과 데이터의 이동
2. 문자단위 입출력 함수
3. 문자열 단위 입출력 함수
4. 표준 입출력 버퍼
5. 입출력이외의 문자열 관련 함수

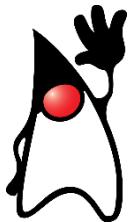


# 입력과 출력의 정의

3

- 응용프로그램 내부로 들어오는 데이터가 입력(input)이고 외부로 나가는 데이터가 출력(output)임



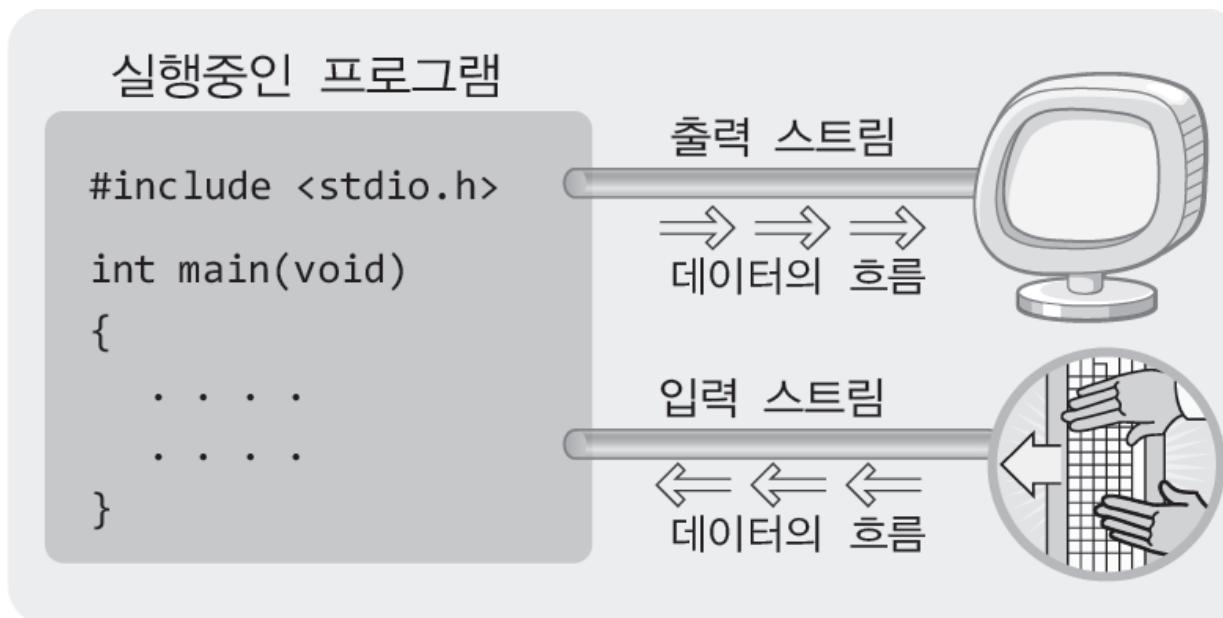


# 데이터의 이동 경로인 스트림

4

## □ 스트림(stream)

- 운영체제(OS)가 데이터의 입출력을 위해 놓아주는 소프트웨어적인 형태의 다리 또는 데이터의 경로
- 콘솔(키보드, 모니터) 입출력을 위한 스트림은 프로그램이 시작되면 OS에 의해서 자동으로 생성

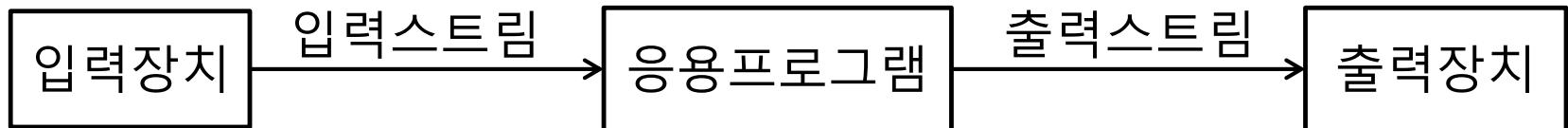




# 데이터의 이동 경로인 스트림

5

- 스트림(stream)의 종류
  - ▣ 입력 스트림 : 입력장치와 프로그램을 이어주는 스트림
  - ▣ 출력 스트림 : 출력장치와 프로그램을 이어주는 스트림
- 프로그램에서 데이터의 입출력이 가능한 이유
  - ▣ 출력장치와 경로가 되는 출력 스트림과 입력장치와 경로가 되는 입력 스트림이 존재하기 때문 -> 누가 만들어주나?





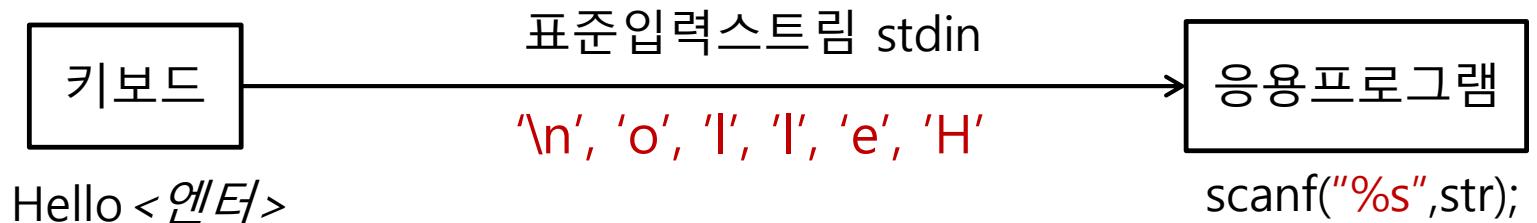
- 표준 입력 스트림(stdin) : 키보드와 연결을 위한 입력 스트림
- 표준 출력 스트림(stdout) : 모니터와 연결을 위한 출력 스트림
- 표준 에러 스트림(stderr) : 에러메시지 출력을 위한 스트림, 기본적으로 모니터와 연결됨
- stdin, stdout, stderr은 모두 **운영체제에 의하여 프로그램 시작과 동시에 자동으로 생성되고 프로그램 종료시 자동으로 소멸됨**
- 이외의 스트림들은 프로그래머가 직접 생성해야 함, 예를 들어 파일 입출력을 위한 스트림은 직접 생성함
- 스트림이라 불리는 이유는 데이터의 이동을 한 방향으로만 형성하기 때문, 물이 한 방향으로 흐르듯 스트림도(스트림은 물의 흐름을 의미함) **한 방향으로만 데이터가 이동함**



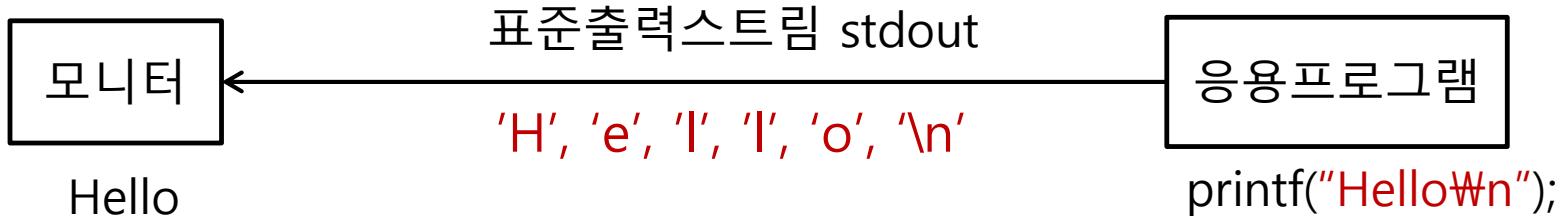
# 스트림의 종류

7

## □ 표준 입력 스트림 stdin



## □ 표준 출력 스트림 stdout

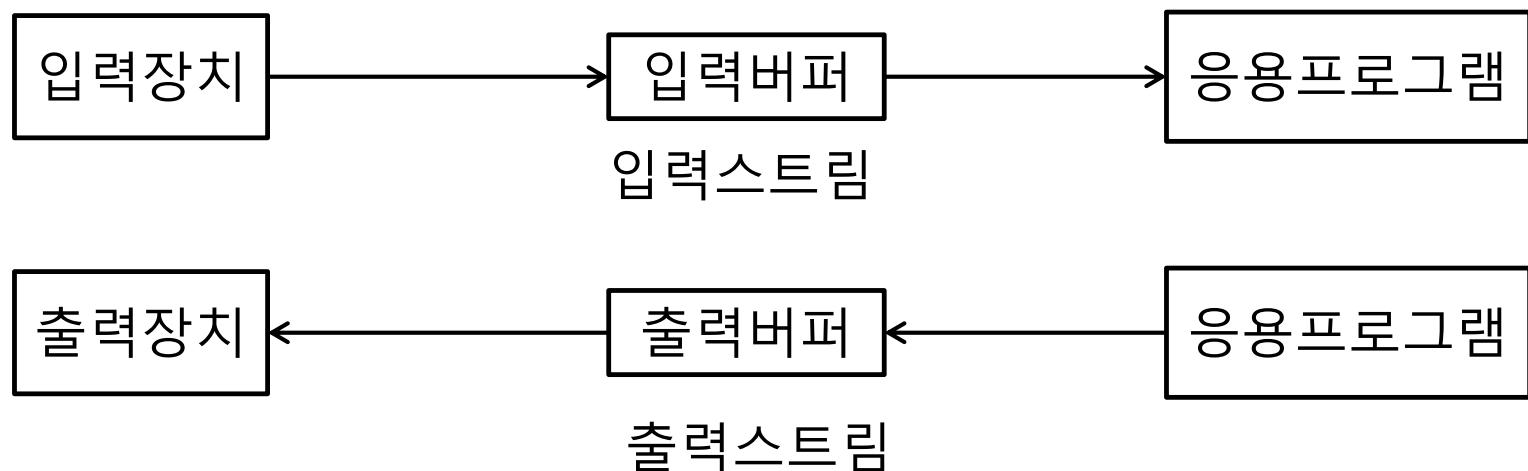




# 버퍼(buffer)

8

- 입출력 장치와 프로그램사이에 존재하여 데이터를 임시로 저장하는 메모리공간 -> 스트림 내부에 포함되어 있고 운영체제가 관리
  - 데이터 입출력시 운영체제의 설정에 따라 버퍼를 사용하지 않거나 버퍼를 사용할 수 있음



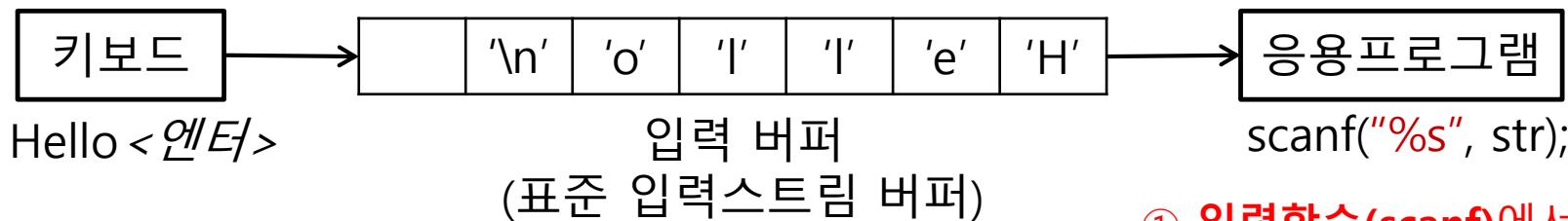


# 버퍼(buffer)

9

## □ 버퍼(buffer)를 이용한 입력

② 운영체제의 함수가 키보드를 누를 때마다 한 문자씩 버퍼로 전송하고 엔터키가 입력되면 대기중인 입력함수(`scanf`)로 복귀



③ 입력함수(`scanf`)가 버퍼에 저장된 데이터를 가져와 메모리에 저장하고 버퍼에서는 삭제됨

① 입력함수(`scanf`)에서 운영체제(디바이스 드라이버)의 함수를 호출하고 대기

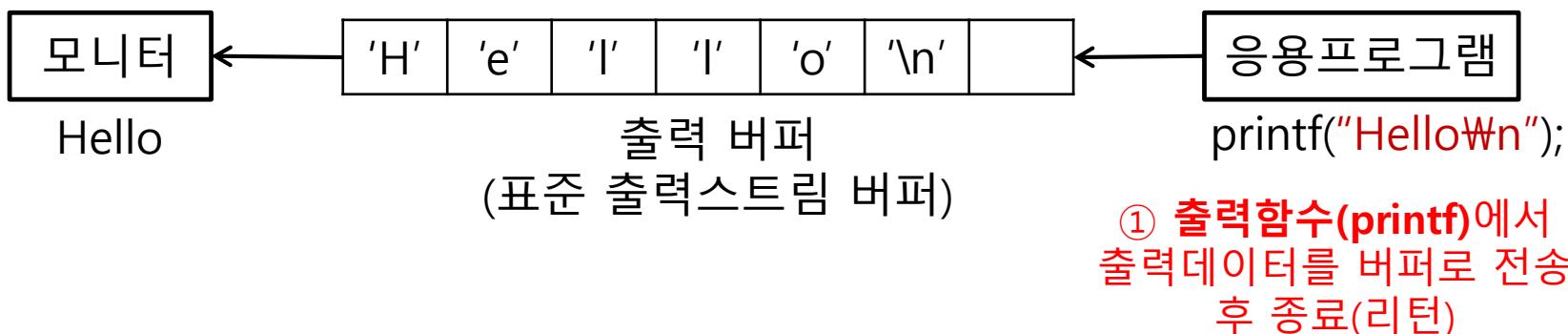


# 버퍼(buffer)

10

- #### □ 버퍼(buffer)를 이용한 출력

② 운영체제가 특정조건(버퍼가 가득 차거나, 문장입력이 완료될 때) 발생시 출력장치로 전송하고 버퍼를 비움

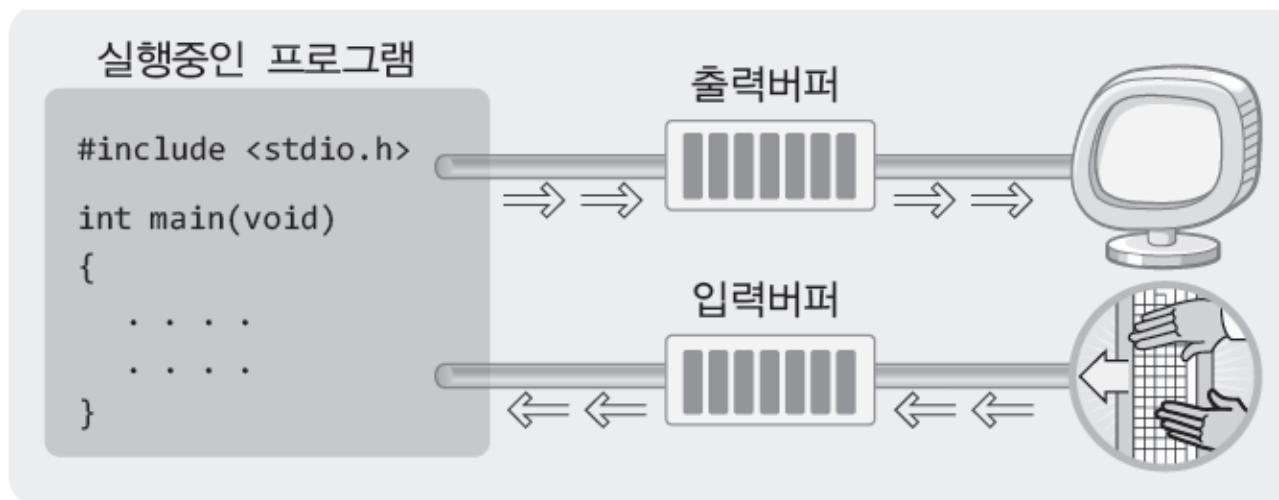




# 버퍼를 사용하는 이유

11

- 창고의 물건을 나르는 경우 하나씩 손으로 나르는 것보다 손수레에 가득 채워서 한번에 나르는 것이 보다 빠르고 효율적임
- 데이터를 버퍼에 모아서 한번에 전송하면, 하나씩 전송하는 것보다 처리시간을 단축
- 버퍼를 사용하면 엔터키가 입력되기전에 데이터의 수정이 가능함





# printf 함수를 이용한 문자 출력

12

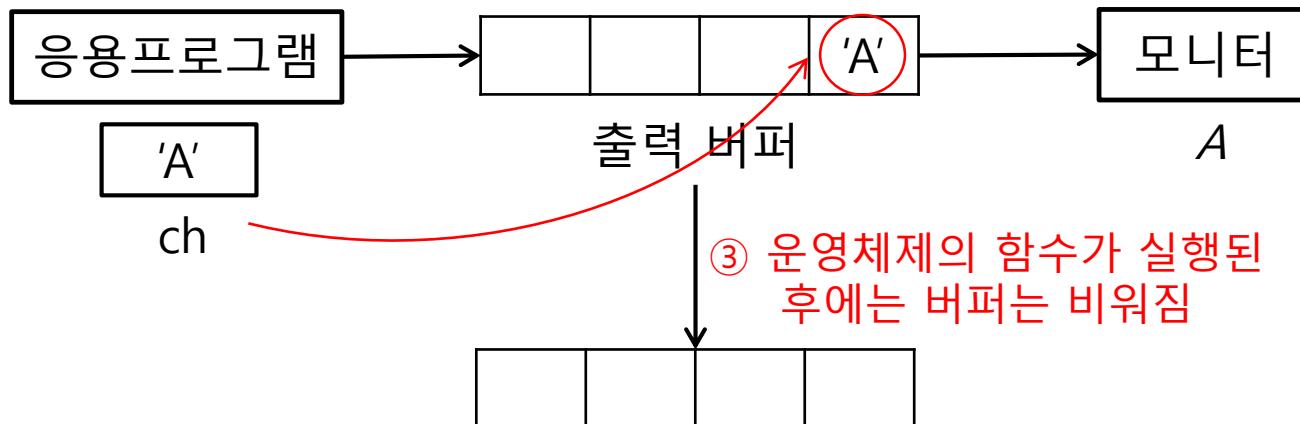
- 버퍼이용, 서식문자 %c 이용, 출력 버퍼에 문자전송

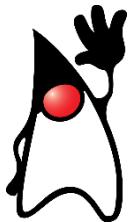
```
char ch = 'A';
printf("%c", ch);
```

A

① printf함수에서 문자를 버퍼가 전송

② 운영체제의 함수가 특정조건이 발생하면 버퍼의 데이터를 모니터로 전송하고 버퍼에서 삭제





# printf 함수를 이용한 문자열 출력

13

- 버퍼이용, 서식문자 %s 이용, 출력 버퍼에 문자전송

```
char str[10] = "Hello";
printf("%s", str);
```

Hello

① printf함수에서 문자열을 버퍼에 전송

② 운영체제의 함수가 특정조건이 발생하면 버퍼의 데이터를 모니터로 전송하고 버퍼에서 삭제





# scanf 함수를 이용한 문자 입력

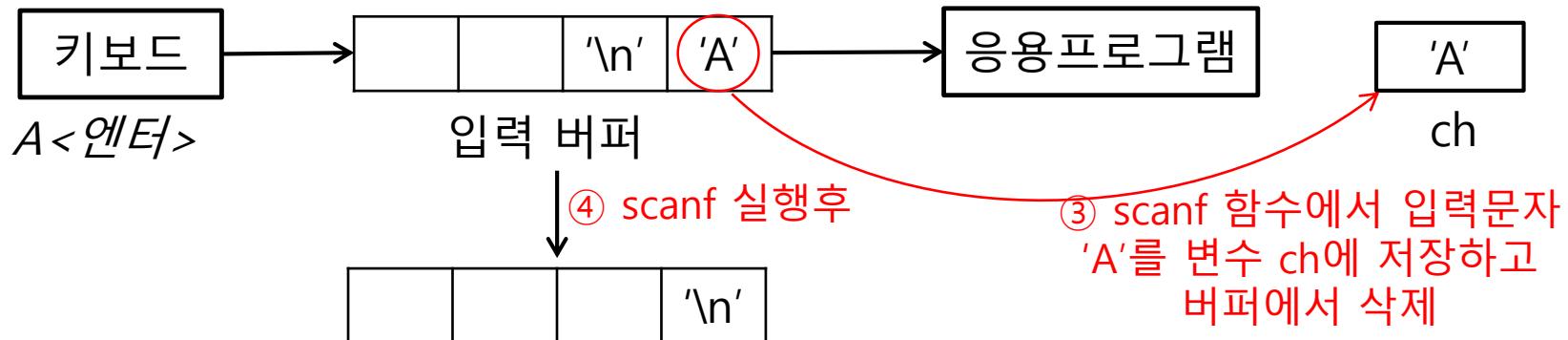
14

- 버퍼이용, 서식문자 %c 이용, 엔터키(입력 완료)가 입력되면 리턴

```
char ch;  
printf("문자를 입력하시오:");  
scanf("%c", &ch);  
printf("%c", ch);
```

문자를 입력하시오: A<엔터>  
A

- ② 운영체제의 함수가 입력된 문자를 버퍼로 전송, 엔터키 입력되면 scanf함수 호출



- ① scanf함수에서 버퍼가 비어 있으면 운영체제의 함수 호출하고 대기

- ③ scanf 함수에서 입력문자 'A'를 변수 ch에 저장하고 버퍼에서 삭제



# scanf 함수를 이용한 문자열 입력

15

- 버퍼이용, 서식문자 %s 이용, 엔터키(입력 완료)가 입력되면 리턴

```
char str[10];
printf("문자열을 입력하시오:");
scanf("%s", str);
printf("%s", str);
```

문자열을 입력하시오: Hello <엔터>  
Hello

- ② 운영체제가 입력된 문자열을 버퍼로 전송, 엔터키 입력되면 scanf함수 호출



- ① scanf함수에서 버퍼가 비어 있으면 운영체제의 함수 호출 후 대기





# printf, scanf 함수의 동작정리

16

- printf 함수는 모니터에 출력하는게 아니라 출력 버퍼에 데이터를 출력함 -> 출력 버퍼에서 모니터로 전송하는 것은 운영체제의 역할임
- scanf 함수는 키보드에서 데이터를 입력 받는게 아니라 입력 버퍼에서 데이터를 읽어 옴 -> 키보드에서 입력 버퍼로 전송하는 것은 운영체제의 역할임
- 입력 버퍼가 비어 있으면 사용자의 키보드 입력을 기다림
- 입력 버퍼에 저장된 문자가 있다면 사용자의 입력을 기다리지 않고 버퍼의 문자를 읽고 바로 리턴함



# scanf 함수로 문자열 입력시 문제점

17

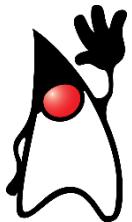
- 문자열에 공백 또는 탭이 포함되면 공백(탭) 이전까지만 저장함
  - 공백을 포함하는 문자열 전체를 입력 받을 때는 fgets 함수 이용

```
char str[10];
printf("문자열을 입력하시오:");
scanf("%s", str);
printf("%s", str);
```

홍길동 <엔터>  
홍길동

군산시 미룡동 558 <엔터>  
군산시





# 예제 1

18

```
#include <stdio.h>
int main(void)
{
    int x;
    char ch;
    printf("정수를 입력하시오:");
    scanf("%d", &x);

    printf("문자를 입력하시오:");
    scanf("%c", &ch);

    printf("정수: %d\n", x);
    printf("문자: %c\n", ch);

    return 0;
}
```

정수를 입력하시오 : 100<엔터>  
문자를 입력하시오 : 정수: 100  
문자:

// 정수 입력

// 바로 리턴함

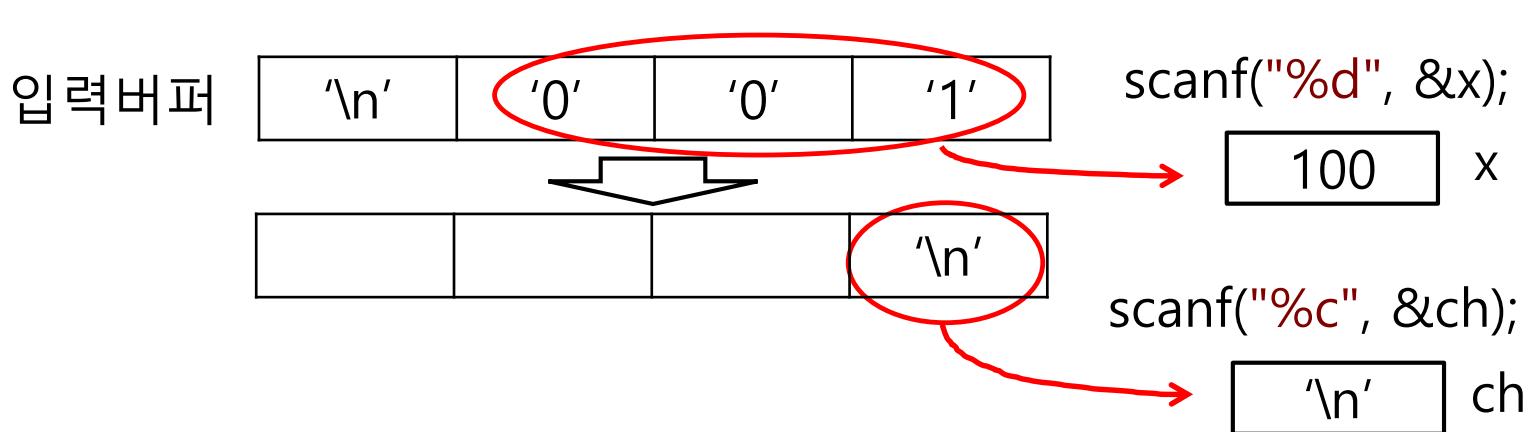
// '\n'을 출력함

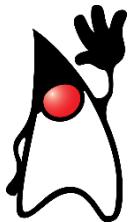


# 예제 1

19

- 키보드로 입력한 데이터는 키보드 버퍼에 아스키코드로 저장됨
- scanf함수의 서식문자에 지정된 자료형으로 변환 후 변수에 저장함
- scanf("%d", &x); -> '1', '0', '0'을 정수 100으로 변환 후 x에 저장하고 버퍼에서는 제거, 그러나 입력완료문자인 엔터키('\n')는 버퍼에 남김
- scanf("%c", &ch); -> 엔터키('\n')를 변수 ch에 저장





## 예제 2

20

```
#include <stdio.h>
int main(void)
{
    int x;
    char ch;
    printf("정수를 입력하시오:");
    scanf("%d", &x);
    scanf("%c", &ch);

    printf("문자를 입력하시오:");
    scanf("%c", &ch);

    printf("정수: %d\n", x);
    printf("문자: %c\n", ch);

    return 0;
}
```

정수를 입력하시오 : 100<엔터>  
문자를 입력하시오 : x<엔터>  
정수: 100  
문자: x

// 정수 입력  
// 버퍼에서 '\n' 제거  
  
// 문자 입력



## 예제 2

21

- scanf("%c", &ch); 문장이 실행되기 전에 엔터키('\n')를 읽어주면 버퍼에서 제거됨
- scanf("%c", &ch); 문장을 문자입력 전에 추가하여 버퍼를 비움
- 버퍼를 비우는 용도로 뒤에서 배울 getchar() 함수를 이용해도 됨

입력버퍼



scanf("%d", &x);

100 x

scanf("%c", &ch);

'\n' ch

'\n'

'x'

scanf("%c", &ch);

'x' ch

'\n'



## 예제 3

22

```
#include <stdio.h>
int main(void)
{
    int x;
    char ch;
    printf("정수를 입력하시오:");
    scanf("%d", &x);

    printf("문자를 입력하시오:");
    scanf(" %c", &ch);

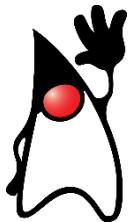
    printf("정수: %d\n", x);
    printf("문자: %c\n", ch);

    return 0;
}
```

정수를 입력하시오 : 100<엔터>  
문자를 입력하시오 : x<엔터>  
정수: 100  
문자: x

// 정수 입력

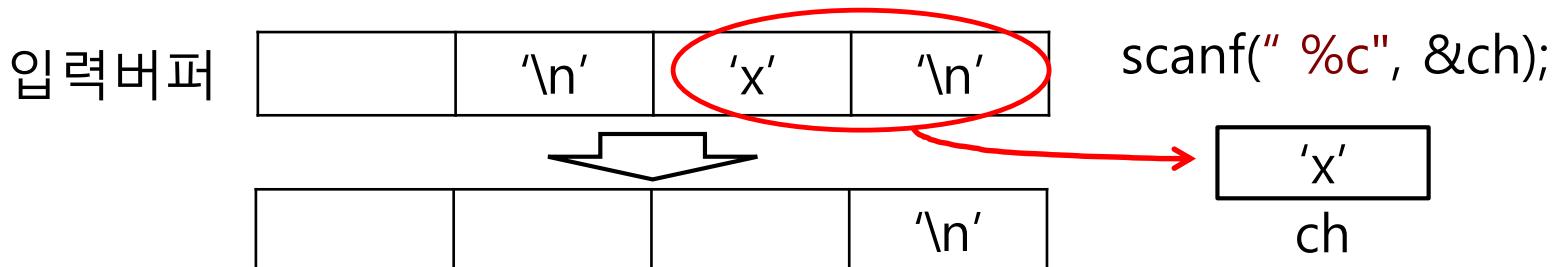
// %c앞에 공백추가->엔터키 무시



## 예제 3

23

- 다른 방법으로는 버퍼를 비울 필요없이 문자를 읽을때 `scanf(" %c", &ch);`; 처럼 `%c`앞에 공백을 추가하여 호출하면 버퍼에 남아있는 '`\n`' 문자를 무시함
- 공백은 whitespace(space, enter, tab) 문자를 의미함,**
- whitespace 문자가 아닌 문자가 나올 때까지 whitespace 문자는 읽어서 버리고 whitespace 문자가 아닌 것만 읽어서 저장하라는 의미**





# 실습과제 1

24

- 버퍼를 사용하여 얻는 장점과 단점을 모두 설명하시오.
- 입력 버퍼에 데이터가 이미 존재할 때 scanf 함수를 호출한다면 어떤 일이 발생하는가?
- 운영체제가 어떤 경우에 출력 버퍼의 데이터를 모니터로 출력하는가?



# 실습과제 2

25

- 키보드로부터 문자를 입력 받아 화면에 출력해주는 프로그램을 작성하시오. 'q' 또는 'Q'를 입력하면 종료한다.
- 힌트: 무한 루프(while(1){...})와 if(종료조건) break문을 이용하라.

문자입력 : a <엔터>

입력된 문자는 a

문자입력: 1 <엔터>

입력된 문자는 1

문자입력: C <엔터>

입력된 문자는 C

문자입력: q <엔터>



# 실습과제 3

26

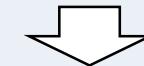
- 위쪽 실행결과처럼 나오는 이유를 설명하고 아래쪽 결과처럼 나오도록 코드를 수정해 보시오. 19, 21페이지처럼 그림을 그려서 설명하라.

```
#include <stdio.h>
int main(void)
{
    int x;
    char name[10];
    char blood_type;

    printf("이름을 입력하시오:");
    scanf("%s", name);
    printf("혈액형을 입력하시오:");
    scanf("%c", &blood_type);

    printf("이름: %s\n", name);
    printf("혈액형: %c\n", blood_type);
    return 0;
}
```

이름을 입력하시오 : 홍길동 <엔터>  
혈액형을 입력하시오 : 이름: 홍길동  
혈액형:



이름을 입력하시오 : 홍길동 <엔터>  
혈액형을 입력하시오 : A <엔터>  
이름: 홍길동  
혈액형: A



# 실습과제 4

27

- 다음 코드의 실행결과를 자세히 설명하라. 19, 21페이지의 그림처럼 반복문 실행단계별로 버퍼의 상태를 그리고 설명하라.

```
#include <stdio.h>
int main(void)
{
    char ch;
    int i;
    for (i = 0; i < 5; i++)
    {
        printf("문자 입력:");
        scanf("%c", &ch);
        printf("문자: %c\n", ch);
    }
    return 0;
}
```

문자 입력:abcde <엔터>  
문자: a  
문자 입력:문자: b  
문자 입력:문자: c  
문자 입력:문자: d  
문자 입력:문자: e



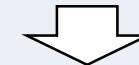
# 실습과제 5

28

- 아래쪽 실행결과처럼 나오도록 문자 입출력(`scanf("%c", ...)`) 방식을 이용하여 코드를 수정하시오. 엔터키('₩n')가 나올 때까지 문자를 하나씩 배열에 복사하고 마지막에 널문자를 대입한다.

```
#include <stdio.h>
int main(void)
{
    char str[20];
    printf("문자열을 입력하시오:");
    scanf("%s", str);
    printf("문자열: %s₩n", str);
    return 0;
}
```

문자열을 입력하시오 : Hello world <엔터>  
문자열: Hello



문자열을 입력하시오 : Hello world <엔터>  
문자열: Hello world



# 과제제출방법

29

- 소스코드, 라인단위의 주석, 실행결과를 포함하는 pdf파일을 작성한 후 eclass 과제 게시판에 업로드, **반드시 하나의 pdf파일로 업로드할 것**
- 기한 : 과제 게시판에 마감시간 참조
- 실행결과를 캡쳐할 때 글자를 알아보기 쉽게 확대해서 캡쳐할 것.
- 소스코드의 첫 부분은 아래처럼 제목, 날짜, 작성자(학번, 이름)를 작성할 것

```
// *****
// 제 목 : 정수 4개의 평균을 구하는 프로그램
// 날 짜 : 2023년 9월 10일
// 작성자 : 15010101 홍길동
// *****

// 소스코드 작성
```