



# C언어 (CSE2035) (Chap10. Strings) (1-2)

---

**Sungwon Jung, Ph.D.**

**Dept. of Computer Science and Engineering**

**Sogang University**

**Seoul, Korea**

**Tel: +82-2-705-8930**

**Email : [jungsung@sogang.ac.kr](mailto:jungsung@sogang.ac.kr)**

# String input/output function

- scanf와 printf 함수를 이용하여 문자열의 입출력이 가능하다.
- scanf() 를 이용한 문자열 입력

```
scanf("%s", month);
```

month가 입력 받을 데이터를 저장하기에 충분한 크기인지를 고려할 필요가 있다.

- [경우1] 길이 9의 문자열을 입력 받아 배열에 저장

```
char month[10];  
scanf("%9s", month);
```

null까지 저장을 해야 하므로 길이 10인 배열을 만든다.

9글자 이상의 문자를 입력 받더라도 최대 9글자까지만을 month에 저장

## Note

%s로 받을 경우, blank(' '), tab('\t')등을 입력 받을 수 없다.

# String input/output function

## ■ scanf() 를 이용한 문자열 입력(cont.)

- [경우2] edit set을 이용한 매치되는 문자만 저장

```
scanf("%10[0123456789.,-,$]", str);
```

%[...] 은 edit set이라 불리며 scanf()를 통해 전달 받은 문자열이 [...]안의 문자와 매치되는 경우에만 str[] 변수에 저장된다.

- “-430,743.33\$”와 같은 문자열을 입력 받는다

- [경우3] edit set과 반대의 기능을 이용하여 저장

```
scanf("%81[^\n]", line);
```

%[ ]안의 ^는 scanf()를 통해 전달 받은 문자열이 ^ 뒤에 나오는 문자와 매치되지 않는 경우에만 str[] 변수에 저장된다.

- ‘\n’를 입력 받기 전까지 모든 문자열을 입력 받는다.

### **Note**

조건을 만족하지 못하는 곳과 그 뒷부분은 모두 무시당한다.

# String input/output function

## ■ scanf 함수를 이용한 문자열 입력의 예제 프로그램

```

1 #include <stdio.h>
2
3 int main(void)
4 {
5     char str[10];
6     char *p;
7
8     printf("Enter a string: ");
9     scanf("%9s", str);
10    p=str;
11
12    while(*p != '\0') {
13        printf("%c", *p);
14        p++;
15    }
16
17    printf("\n");
18 }

```

```

vore@nlpsag:~$ ./a.out
Enter a string: 123456789012345
123456789
vore@nlpsag:~$ ./a.out
Enter a string: 123 456789
123
vore@nlpsag:~$ █

```

최대 9글자를 입력 받는다.

%s로 받을 경우, Blank character ' '가 입력되면 그 앞까지만 받는 것을 확인할 수 있다.

(while 루프를 이용하여) 한 문자씩 출력한다.

- `printf()` 를 이용한 문자열의 출력

- scanf에서와 같이 **%s**를 이용하여 문자열을 출력한다.
- Null character(‘\0’)앞 까지만 출력하므로, 문자열의 중간에 null이 있으면 뒷부분은 출력되지 않는다.
- 30칸에 맞춰서 출력. 오른쪽 정렬

```
printf("|%30s|\n", "This is the string");
```

This is the string

- 30칸에 맞춰서 출력. 왼쪽 정렬

```
printf("|%-30s|\n", "This is the string");
```

This is the string

- 15칸에 맞춰서 출력. 14글자만 출력. 왼쪽 정렬

```
printf("%-15.14s\n", "12345678901234567890");
```

1	2	3	4	5	6	7	8	9	0	1	2	3	4
---	---	---	---	---	---	---	---	---	---	---	---	---	---

# String input/output function

- printf를 이용한 string의 출력 예제 프로그램

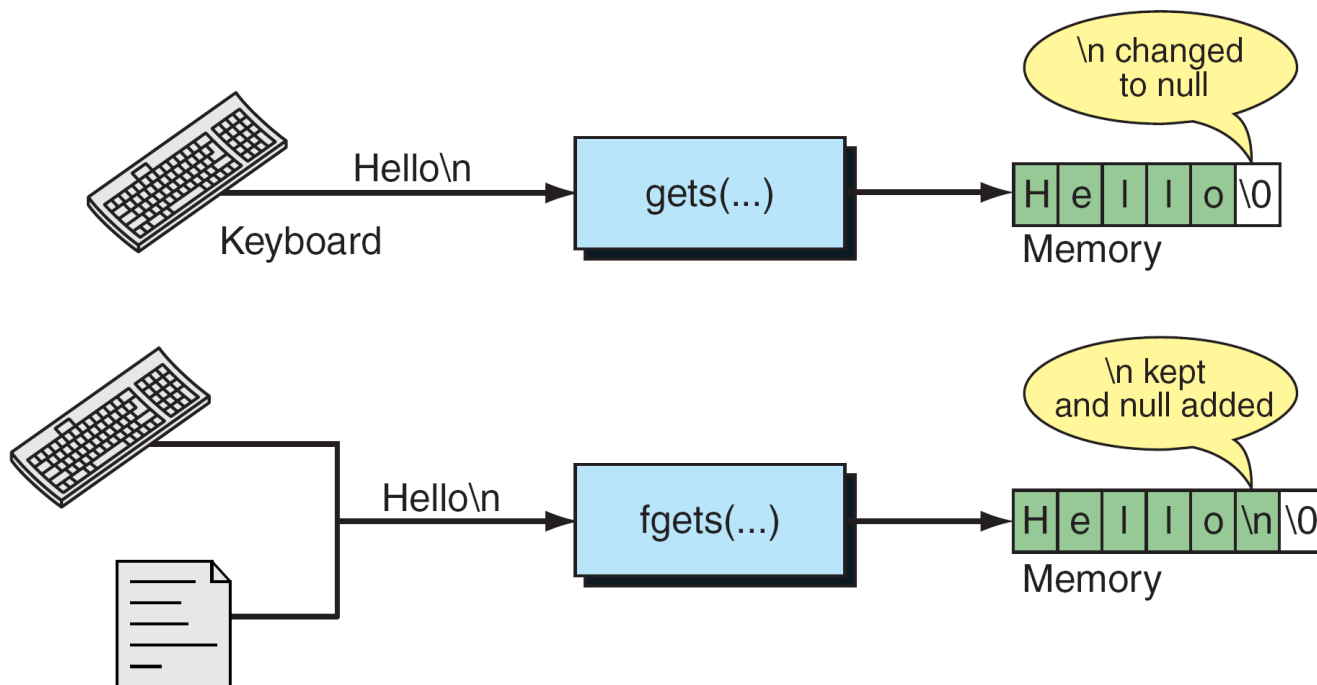
```
1 #include <stdio.h>
2
3 int main(void) {
4
5     printf("|%30s|\n", "This is the string");
6     printf("|%-30s|\n", "This is the string");
7     printf("|%-15.14s|\n", "12345678901234567890");
8     printf("|%15.14s|\n", "12345678901234567890");
9
10 }
```

```
./a.out
|                This is the string|
|This is the string                |
|12345678901234 |
| 12345678901234|
vore@nlpsag:~$
```

# String input/output function

## ■ gets / fgets

- format에 따라 입력을 받는 scanf, fscanf 등과 달리 formatting 없이 한 line을 읽어 들이는 함수.



# String input/output function

## ■ gets / fgets 함수 proto-type

### ■ char\* gets (char\* strPtr);

- 키보드로부터 한 line을 입력받아 strPtr에 저장
- 한 line은 '\n'을 입력 받을 때까지를 의미, '\n'은 '\0'으로 치환되어 저장된다.
- 한 line의 길이가 strPtr의 길이보다 길면 strPtr 뒷부분의 메모리가 침범되어 Segmentation fault가 발생하므로 주의해야 한다.

성공 : strPtr의 주소를 return  
실패 : NULL

### ■ char\* fgets (char\* strPtr, int size, FILE \*fp);

- file 포인터 fp로부터 한 line을 읽어 들여 strPtr에 저장
- ➔ 읽어 들이는 문자의 최대 개수는 size-1개이다.
- 한 line은 '\n'을 입력 받을 때까지를 의미, '\n'까지 입력 받고 뒤에 '\0'이 추가로 저장됨.
- File 포인터에 stdin을 주면 키보드로부터 입력을 받을 수 있다.
- ➔ 일반적으로 gets(strPtr)보다 fgets(strPtr, sizeof(strPtr), stdin); 의 사용이 권장된다.

성공 : strPtr의 주소를 return  
실패, 파일의 끝(EOF) : NULL



# String Input/Output Functions

- fgets 를 사용한 예제 프로그램

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5
6     char str[81];
7
8     printf("Please enter a string: ");
9     fgets(str, sizeof(str), stdin);
10    printf("Here is your string: \n\t%s", str);
11
12    return 0;
13 }
```

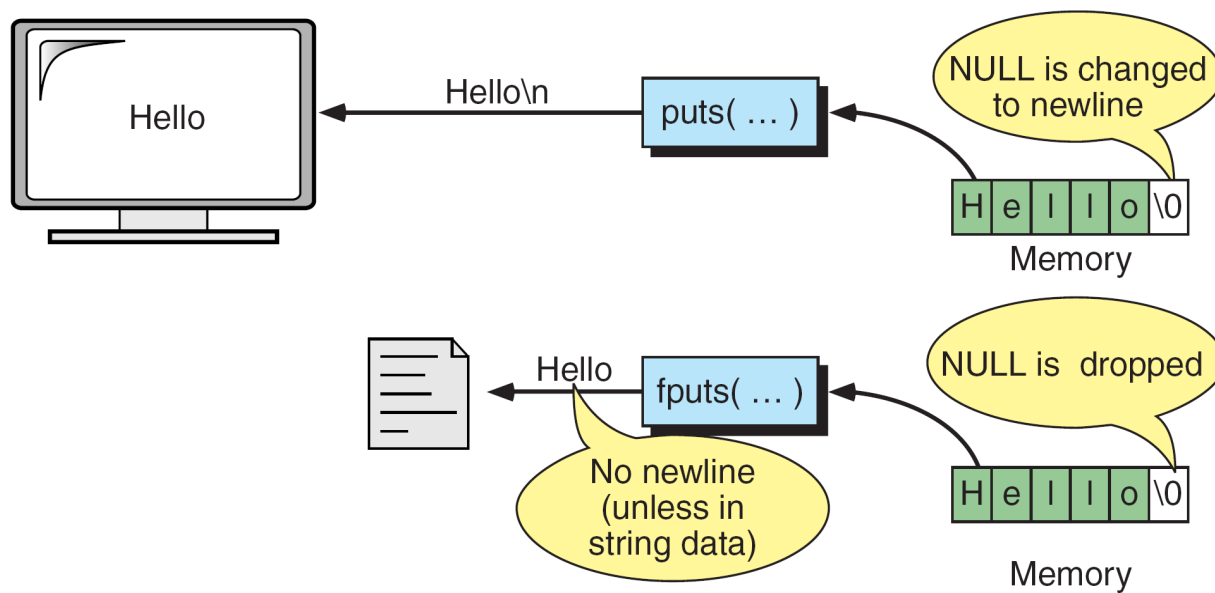
키보드를 통해  
한 줄을 입력받  
는다.

```
[root@mclab chap11]# ./chap11-3
Please enter a string: Now is the time for all students
Here is your string:
        Now is the time for all students
[root@mclab chap11]#
```

# String input/output function

## ■ puts / fputs

- format에 따라 출력을 하는 printf, fprintf 등과 달리 formatting 없이 한 line을 출력하는 함수.



# String input/output function

## ■ puts / fputs 함수 proto-type

■ int puts (const char \*strPtr);

- strPtr의 문자열을 모니터에 한 line으로 출력
- '\0'이 '\n'으로 치환되므로 문자열 끝에 '\n'이 있다면 줄 바꿈이 두 번 일어난다.
- '\0'까지만 출력하므로 문자열 중간에 '\0'이 있으면 뒷부분은 무시된다.

성공 : 출력한 글자 수  
('\n'로 치환된 '\0' 포함)  
실패 : EOF(-1)

■ int fputs (const char \*strPtr, FILE \*fp);

- strPtr의 문자열을 file 포인터 fp에 기록
- strPtr은 '\0'으로 끝나야 하며 NULL문자는 출력되지 않는다.
- '\0'까지만 출력하므로 문자열 중간에 '\0'이 있으면 뒷부분은 무시된다.
- File 포인터 fp에 stdout을 주면 모니터에 출력할 수 있다.

성공 : 1  
실패 : EOF(-1)

➔ gets()와 fgets()의 큰 차이와 달리 puts(strPtr)와 fputs(strPtr, stdout)은 '\n'이 마지막에 출력되는가의 여부, 함수의 return값만 다르다.

# String Input/Output Functions

## ■ fputs 를 사용한 예제 프로그램

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5
6     char str[]="Necessity Is the Mother of Invention.";
7     char* pStr = str;
8
9     fputs(pStr, stdout);
10    fputs("\n", stdout);
11    fputs(pStr+13, stdout);
12    fputs("\n", stdout);
13
14    return 0;
15 }
```

문자열의 시작위치를 지정

문자열의 시작부터 모두 출력

문자열의 13번째 문자부터 출력

```
[root@mclab chap11]# ./chap11-4
Necessity Is the Mother of Invention.
the Mother of Invention.
[root@mclab chap11]#
```

# String Input/Output Functions

## ■ fgets / fputs 사용 예제프로그램

- 다음 프로그램은 키보드로 입력 받아서 대문자로 시작할 경우에만 출력한다.

EOF(Ctrl + z)를  
입력 받을 때 까지  
반복

```
#include <stdio.h>
```

```
main(void){
    char str[81];
```

```
while(fgets(str, sizeof(str), stdin) != NULL)
```

```
if( 'A' <= str[0] && str[0] <= 'Z' )
    fputs(str, stdout);
```

```
return 0;
```

```
11 }
```

```
vore@nlpsag:~$ ./a.out
```

```
Now is the time
```

```
Now is the time
```

```
for all good students
```

```
to come to the aid of their school.
```

```
Amen
```

```
Amen
```

```
^Z
```

```
[5]+ 정지됨
```

```
vore@nlpsag:~$
```

```
./a.out
```

키보드로부터 문자열을  
입력 받는다.

입력 받은 문자열이 대문자로  
시작할 경우에만 출력한다.

```
Now is the time
```

```
Now is the time
```

```
for all good students
```

```
to come to the aid of their school.
```

```
Amen
```

```
Amen
```

```
^Z
```

```
계속하려면 아무 키나 누르십시오 . . .
```

- Linux에서 ^Z(Ctrl+Z, EOF)를 입력하면 프로그램이 강제종료 되지만, Windows에서는 그렇지 않고 fgets에서 NULL이 return되어 while문을 빠져나가는 것을 볼 수 있다.