



C언어 (CSE2035) (Chap10. Strings) (2-2)

Sungwon Jung, Ph.D.

Dept. of Computer Science and Engineering

Sogang University

Seoul, Korea

Tel: +82-2-705-8930

Email : jungsung@sogang.ac.kr

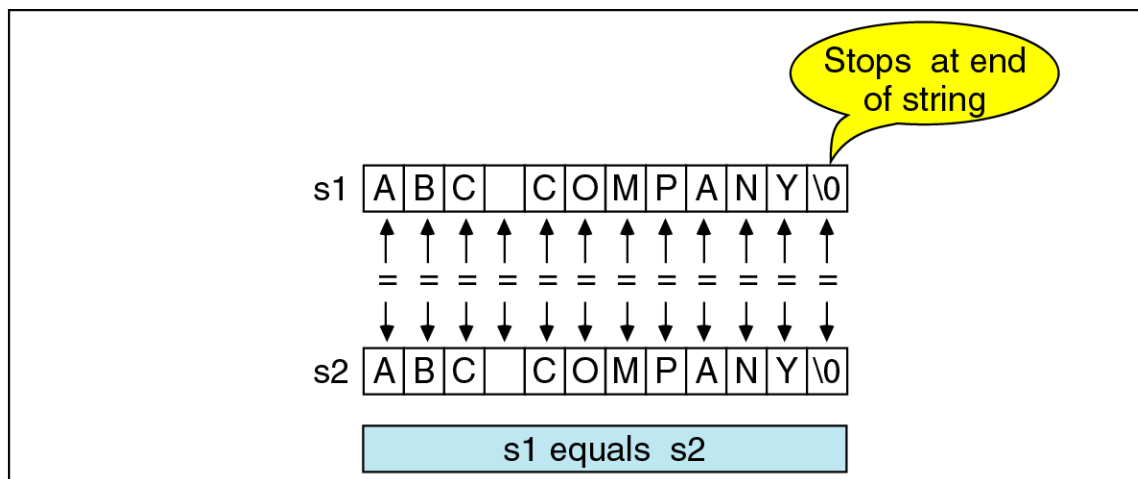
String manipulation function

■ strcmp()

- strcmp는 string compare의 약자로서, 이 함수는 두 문자열을 비교한다

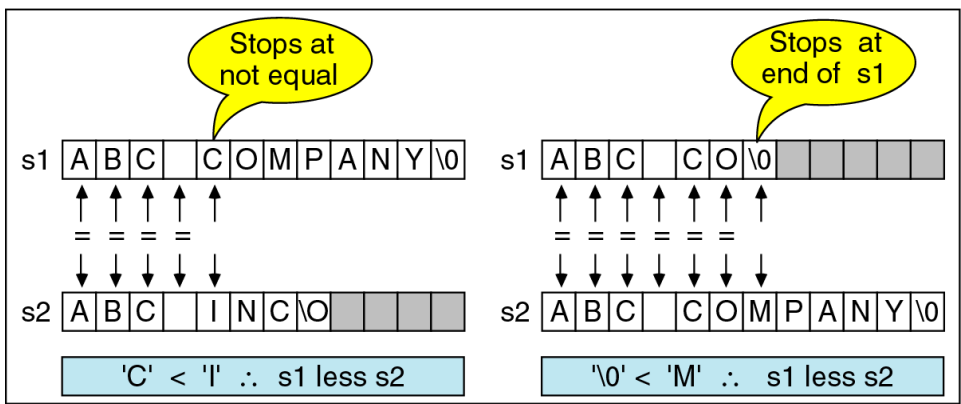
```
int strcmp (const char *string1, const char *string2);
```

- 두 문자열을 순차적으로 비교하여 string1 이 string2보다 작으면 음수, 크면 양수, 같으면 0을 리턴한다.
- strcmp(s1, s2)와 같이 호출했을 때, 다음과 같은 경우에는 두 문자열이 서로 같으므로 0을 리턴한다.



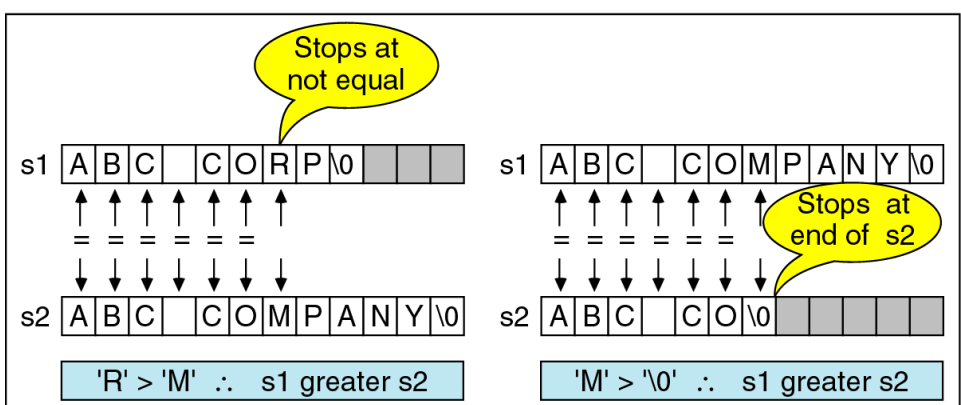
String manipulation function

- 다음은 음수가 리턴되는 경우이다.



- 첫 번째 경우는 'C'가 'I'보다 작다.
- 두 번째 경우는 내용은 서로 같지만 s1이 s2보다 짧다.

- 다음은 양수가 리턴되는 경우이다.



- 첫 번째 경우는 'R'이 'M'보다 작다.
- 두 번째 경우는 내용은 서로 같지만 s2가 s1보다 짧다.

String manipulation function

■ strcmp()

- strcmp도 strcmp처럼 두 문자열을 비교하는 함수이다.

```
int strcmp (const char *string1,  
            const char *string2,  
            int    size);
```

- strcmp는 두 문자열을 순차적으로 size의 길이 만큼 비교한다.
- 비교 결과에 따른 리턴 값은 strcmp와 같다.
(string1 이 string2보다 작으면 음수, 크면 양수, 같으면 0을 리턴 한다.)

String manipulation function

- `strncmp(string1, string2, size);` 와 같이 호출했을 때의 리턴 값

string1	string2	Size	Results	Returns
"ABC123"	"ABC123"	8	equal	0
"ABC123"	"ABC456"	3	equal	0
"ABC123"	"ABC456"	4	string1 < string2	< 0
"ABC123"	"ABC"	3	equal	0
"ABC123"	"ABC"	4	string1 > string2	> 0
"ABC"	"ABC123"	3	equal	0
"ABC123"	"123ABC"	-1	equal	0

string1과 string2는 서로 다른 문자열이지만 맨 앞의 3글자만 비교했을 때는 서로 같으므로 0을 리턴한다.

String manipulation function

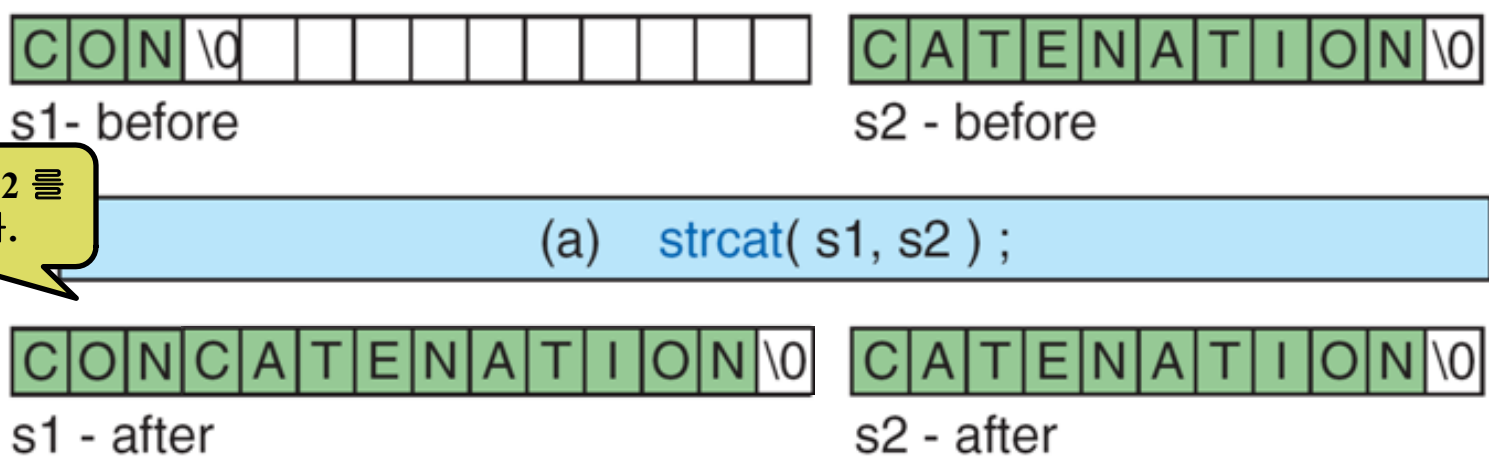
■ strcat()

- strcat는 string concatenation의 약자로, 이 함수는 두 문자열을 결합한다.

```
char *strcat (char *string1, const char *string2);
```

Return : string1

- string1 의 마지막 '\0' 자리부터 string2의 값을 결합한다.



S1 다음에 S2 를 결합한다.

String Concatenate

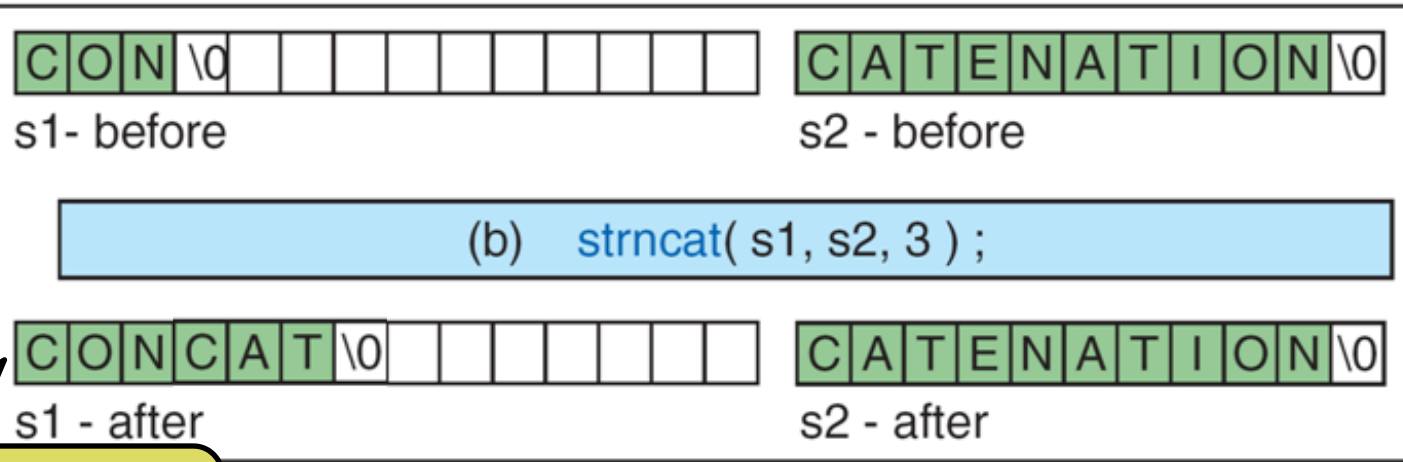
String manipulation function

■ strncat()

- strncat 함수도 strcat 함수처럼 두 문자열을 결합하는 함수이다.
- strncat는 string1 의 마지막 '\0' 자리부터 size 길이 만큼의 string2의 값을 결합한다.

```
char *strncat (      char    *string1,
                    const   char    *string2,
                        int    size);
```

Return : string1



S1 다음에 S2의 3번째까지의 문자를 결합한다.

String N Concatenate

Memory formatting

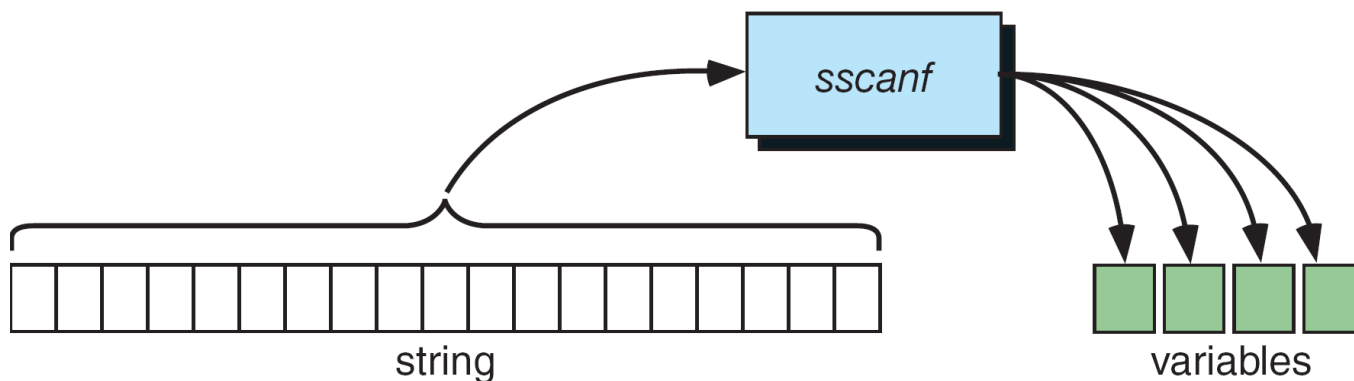
■ sscanf()

- 메모리에 저장된 문자열을 입력으로 받아들여 이를 각 변수에 저장한다.
- fscanf가 scanf의 파일 버전인 것처럼 sscanf는 scanf()의 메모리 버전이라고 생각할 수 있다.

```
int sscanf (char *str, const char *format_string, ...);
```

Return :
성공적으로 읽어
들인 데이터 수

- 실제 사용 방법도 fscanf()와 거의 같다.



Memory formatting

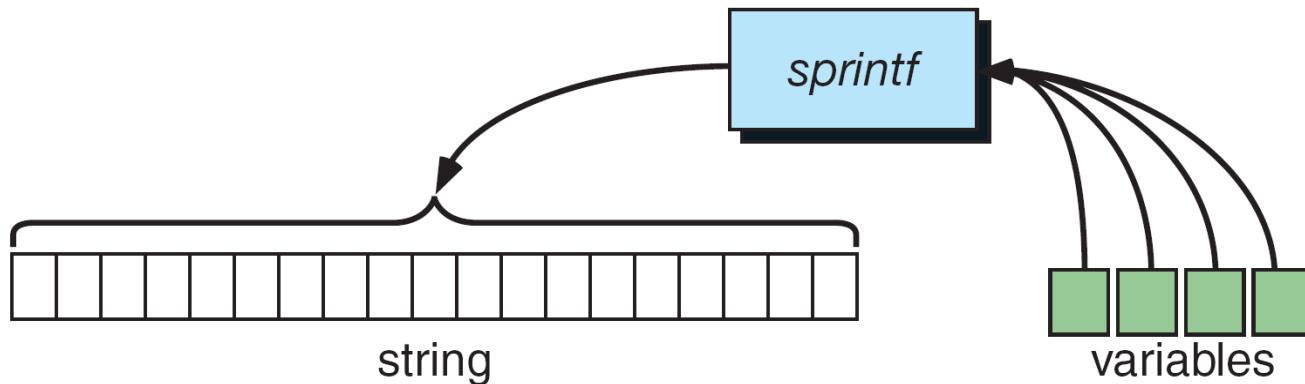
■ sprintf()

- 각 문자열 (또는, 변수들)을 파라미터로 입력 받아 이를 서식에 따라 하나의 문자열로 저장한다 .
- fprintf가 printf의 파일 버전인 것 처럼, sprintf는 printf의 배열 버전이라고 생각할 수 있다.

```
int sprintf (      char *out_string,
                  const char *format_string, ...);
```

Return :
NULL을 제외한
출력한 문자 수
Error : EOF

- 실제 사용 방법도 fprintf와 거의 같다.



Memory formatting

■ 예제 프로그램 – sscanf() , sprintf()

```

1 #include <stdio.h>
2
3 int main(void){
4
5     char str[80] = "Einstein, Albert; 1234 97 A";
6     char strOut[80];
7     char name[50];
8     char id[5];
9     int score;
10    char grade;
11    int n1, n2;
12
13    printf("String contains : \"%s\\n\"", str);
14    n1 = sscanf(str, "%49[^\n] %c %4s %d %c", name, id, &score, &grade);
15
16    printf("Reformatted data : \\n");
17    printf("\\tName : \\t[%s]\\n", name);
18    printf("\\tID : \\t[%s]\\n", id);
19    printf("\\tScore: \\t%d\\n", score);
20    printf("\\tGrade: \\t%c\\n", grade);
21
22    n2 = sprintf(strOut, "[%s] / %4s / %3d / %c", name, id, score, grade);
23
24    printf("New String : \\t\"%s\\n\"", strOut);
25
26    printf("n1 = %d, n2 = %d\\n", n1, n2);
27 }
    
```

str에 저장되어 있는 내용을 형식에 맞게 입력 받는다.

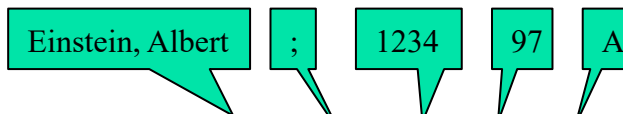
입력 받은 내용을 원하는 형식으로 바꾸어 출력한다.

```

./a.out
String contains : "Einstein, Albert; 1234 97 A"
Reformatted data :
    Name : [Einstein, Albert]
    ID : [1234]
    Score: 97
    Grade: A
New String : "[Einstein, Albert] / 1234 / 97 / A"
n1 = 4, n2 = 35
    
```

Memory formatting

- `str[80] = "Einstein, Albert; 1234 97 A";`



- `n1 = sscanf(str, "%49[^\n] %*c %4s %d %c", name, id, &score, &grade);`

array `str`의 내용으로 부터 다음과 같이 형식에 맞게 읽는다

`%49[^\n]` – ‘;’ 가 입력되기 전까지 최대 49자를 읽어 `name`에 저장

`%*c` – 한 글자를 읽고 **그 글자를 무시**

`%4s` – 4글자를 읽어 `id`에 저장

`%d` – 숫자 하나를 읽고 `score`에 저장

`%c` – 글자 하나를 읽고 `grade`에 저장

- 성공적으로 읽은 개수를 `n1`에 저장하므로 `n1 = 4`가 출력된다.