

# 문자열 관련 함수

# 문자열 처리 함수

- size\_t strlen (const char \*string)
  - 문자열의 길이를 반환하는 함수

```

~/String
1 #include <stdio.h>
2 #include <string.h>
3
4 int main(void){
5     char str1[]="Programming";
6     char str2[]="Project";
7     char str3[]="201111000";
8
9     printf("str1 length : %d \n", strlen(str1));
10    printf("str2 length : %d \n", strlen(str2));
11    printf("str3 length : %d \n", strlen(str3));
12
13    return 0;
14 }
~
~
~
"strlen.c" 14L, 287C          1,5
    
```

```

~/String
DSLab@DESKTOP-10J0DED ~/String
$ vi strlen.c

DSLab@DESKTOP-10J0DED ~/String
$ vi strlen.c

DSLab@DESKTOP-10J0DED ~/String
$ gcc strlen.c

DSLab@DESKTOP-10J0DED ~/String
$ ./a.exe
str1 length : 11
str2 length : 7
str3 length : 9

DSLab@DESKTOP-10J0DED ~/String
$ |
    
```

# 문자열 처리 함수

- char\* strcpy (char \*dest, const char \*src)
  - src의 문자열을 dest에 복사하는 함수
  - dest가 가르키는 배열의 크기가 scr 보다 크거나 같아야 함.

```

~/String
1 #include <stdio.h>
2 #include <string.h>
3
4 int main(void){
5     char str1[] = "programming";
6     char str2[20];
7
8     strcpy(str2, str1);
9     printf("%s\n", str2);
10
11     return 0;
12 }
"strcpy.c" 12L, 161C      6,2-5

```

```

~/String
DSLab@DESKTOP-10J0DED ~/String
$ ls
a.exe  strcpy.c  strlen.c

DSLab@DESKTOP-10J0DED ~/String
$ gcc strcpy.c

DSLab@DESKTOP-10J0DED ~/String
$ ./a.exe
programming

DSLab@DESKTOP-10J0DED ~/String
$ |

```

# 문자열 처리 함수

- char\* strncpy (char \*dest, const char \*src, size\_t n)
  - src의 문자열에서 n 바이트 만큼 dest에 복사하는 함수

```

~/String
1 #include <stdio.h>
2 #include <string.h>
3
4 int main(void){
5     char source[40] = "1234567890";
6     char dest[40] = "programming";
7
8     printf("1. dest = %s \n", dest);
9     strncpy(dest, source, 5);
10    printf("2. dest = %s \n", dest);
11    strncpy(dest, source, sizeof(char)*strlen(source));
12    printf("3. dest = %s \n", dest);
13
14    return 0;
15 }
    
```

```

~/String
DSLab@DESKTOP-10J0DED ~/String
$ ls
a.exe strcpy.c strlen.c strncpy.c

DSLab@DESKTOP-10J0DED ~/String
$ gcc strncpy.c

DSLab@DESKTOP-10J0DED ~/String
$ ./a.exe
1. dest = programming
2. dest = 12345amming
3. dest = 1234567890g

DSLab@DESKTOP-10J0DED ~/String
$ |
    
```

# 문자열 처리 함수

- char\* strcmp (const char \*s1, const char \*s2)
  - 두 문자열이 동일하면 0을 반환
  - 사전편찬 순서상 s1이 s2 보다 앞부분에 위치하면 음수 그렇지 않은 경우 양수 반환

```
~/String
1 #include <stdio.h>
2 #include <string.h>
3
4 int main(void){
5     int result;
6     char str1[]="abcdefg";
7     char str2[]="bqwetas";
8
9     printf("두 문자열 비교 : %d\n", strcmp(str1, str2));
10    strcpy(str2, "abcdefg");
11    printf("두 문자열 비교 : %d\n", strcmp(str1, str2));
12
13    return 0;
14 }
15
~
~
"strcmp.c" 15L, 279C          7,2-5
```

```
~/String
DSLab@DESKTOP-10J0DED ~/String
$ ls
a.exe strcmp.c strcpy.c strlen.c strncpy.c

DSLab@DESKTOP-10J0DED ~/String
$ gcc strcmp.c

DSLab@DESKTOP-10J0DED ~/String
$ ./a.exe
두 문자열 비교 : -1
두 문자열 비교 : 0

DSLab@DESKTOP-10J0DED ~/String
$
```

# 문자열 처리 함수

- char\* strcmp (const char \*s1, const char \*s2, size\_t n)
  - n까지만 문자열을 비교해서 두 문자열이 동일하면 0을 반환
  - 사전편찬 순서상 s1이 s2 보다 앞부분에 위치하면 음수 그렇지 않은 경우 양수 반환

```

~/String
1 #include <stdio.h>
2 #include <string.h>
3
4 int main(void){
5     int result;
6     char str1[]="abcdefg";
7     char str2[]="abcetas";
8
9     printf("strcmp : %d\n", strcmp(str1, str2));
10    printf("strncmp : %d\n", strncmp(str1, str2, 3));
11
12    return 0;
13 }
14
"strncmp.c" 14L, 230C
    
```

```

~/String
DSLab@DESKTOP-10J0DED ~/String
$ ls
a.exe  strcmp.c  strcpy.c  strlen.c  strncmp.c  strncopy.c

DSLab@DESKTOP-10J0DED ~/String
$ gcc strncmp.c

DSLab@DESKTOP-10J0DED ~/String
$ ./a.exe
strcmp : -1
strncmp : 0

DSLab@DESKTOP-10J0DED ~/String
$
    
```

# 문자열 처리 함수

- char\* strtok (char \*str, const char \*set)
  - 문자열을 문자로 자르는 함수
  - str : 대상 문자열, set : 잘라 내기 위한 문자 셋
  - 반환 : 잘라 내기한 문자열의 첫번째 포인터

```

~/String
1 #include <stdio.h>
2 #include <string.h>
3
4 int main(void){
5     int i = 1;
6     char str[]="Programming Project 2017";
7     char *dest;
8
9     dest = strtok(str, " ");
10    printf("%d. %s \n", i, dest);
11    i++;
12    while(dest = strtok(NULL, " ")){
13        printf("%d. %s \n", i, dest);
14        i++;
15    }
16    return 0;
17 }
"strtok.c" 17L, 274C          14,3-9
    
```

```

~/String
DSLab@DESKTOP-10J0DED ~/String
$ ls
a.exe      strcpy.c  strcmp.c  strtok.c
          strlength.c  strncpy.c

DSLab@DESKTOP-10J0DED ~/String
$ gcc strtok.c

DSLab@DESKTOP-10J0DED ~/String
$ ./a.exe
1. Programming
2. Project
3. 2017

DSLab@DESKTOP-10J0DED ~/String
$ |
    
```