④ string.h :

문자열 (String) 관련 명령

메모리 블록이나 문자열을 다룰 수 있는 함수들을 포함하는 헤더 파일

- 메모리 / 문자열 복사

- 문자열 병합

- 메모리 / 문자열 비교

- 문자열 탐색

등의 기능을 수행 합니다.

[메모리 / 문자열 복사]

**-void \* memcpy ( void \* destination, const void \* source, size\_t num );**

지정한 크기 만큼의 메모리 영역 복사

source가 가리키는 곳 부터 num바이트 만큼을 destination이 가리키는 곳에 복사한다.

**-void \* memmove ( void \* destination, const void \* source, size\_t num );**

메모리 블록을 옮긴다.

source가 가리키는 곳 부터 num바이트 만큼을 destination이 가리키는 곳으로 옮긴다.

**-char \* strcpy ( char \* destination, const char \* source );**

지정한 크기 만큼의 문자열 복사

source를 destination에 복사한다.

**-char \* strncpy ( char \* destination, const char \* source, size\_t num );**

문자열에서 일부 문자들 만을 복사한다.

source에서 destination으로 처음 num개의 문자들을 복사한다.

[문자열 병합]

**-char \* strcat ( char \* destination, const char \* source );**

문자열을 덧붙인다.

source를 destination뒤에 붙인다.

**-char \* strncat ( char \* destination, char \* source, size\_t num );**

문자열에 일부 문자들을 덧붙인다.

source에서 destination뒤에 처음 num개의 문자들을 붙인다.

[메모리 / 문자열 비교]

**int memcmp ( const void \* ptr1, const void \* ptr2, size\_t num );**

ptr1이 가리키는 처음 num바이트의 데이터와

ptr2가 가리키는 처음 num바이트의 데이터를 비교한다.

**int strcmp ( const char \* str1, const char \* str2 );**

str1과 str2를 비교한다.

**int strcoll ( const char \* str1, const char \* str2 );**

strcmp와 비슷하지만 LC\_COLLATE에 정의되어 있는 방식에 따라 해석 된 후 비교한다.

**int strncmp ( const char \* str1, const char \* str2, size\_t num );**

str1의 처음 num개의 문자를 str2의 처음 num개의 문자와 비교한다.

**size\_t strxfrm ( char \* destination, const char \* source, size\_t num );**

source를 현재 지역 정보에 따라 문자열을 변환한 후

변환한 문자열의 처음 num개 문자를 destination에 복사한다.

**-void \* memchr ( const void \* ptr, int value, size\_t num );**

ptr이 가리키는 메모리의 처음 num바이트 중에서

처음으로 value와 일치하는 값(문자)의 주소를 반환한다.

**-char \* strchr ( const char \* str, int character );**

str에서 처음으로 character와 일치하는 문자의 주소를 반환한다.

**-size\_t strcspn ( const char \* str1, const char \* str2 );**

str2에 들어있는 문자들 중 str1에 들어있는 문자와 일치하는 것이 있다면

첫 번째로 일치하는 문자까지 읽어들인 수를 반환한다.

**-char \* strpbrk ( const char \* str1, const char \* str2 );**

str1에서 str2에 들어 있는 문자들을 찾아 str2의 문자들 중

str1의 문자들과 첫 번째로 일치하는 문자의 주소를 반환한다.

**-char \* strrchr ( const char \* str, int character );**

str에서 마지막으로 character와 일치하는 문자의 주소를 반환한다.

**-size\_t strspn ( const char \* str1, const char \* str2 );**

str2의 문자들만을 포함하고 있는 str1의 처음 부분의 길이를 구한다.

**-char \* strstr ( const char \* str1, const char \* str2 );**

str1에서 str2를 검색하여 가장 먼저 나타나는 곳의 위치를 반환한다.

**-char \* strtok ( char \* str, const char \* delimiters );**

str1을 delimiters의 문자들로 분리한다.

[기타]

**-void \* memset ( void \* ptr, int value, size\_t num );**

메모리 블록을 채운다.

ptr이 가리키는 메모리의 처음 num바이트를 value값(문자)으로 채운다.

**-char \* strerror ( int errnum );**

오류 메세지 문자열을 가리키는 포인터를 얻어온다.

errnum(보통 errno)을 해석한 뒤 그에 해당하는 에러 문자열의 포인터를 반환한다.

**-size\_t strlen ( const char \* str );**

문자열의 길이를 구한다.

str의 길이를 반환한다.

[메모리 복사] 예제

#include <stdio.h>

#include <string.h>

int main() {

char str1[] = "Sample string";

char str2[40];

char str3[40];

memcpy(str2, str1, strlen(str1) + 1);

memcpy(str3, "copy successful", 16);

printf("str1: %s\nstr2: %s\nstr3: %s\n", str1, str2, str3);

memmove(str2,str3+5,7);

printf(str2);

return 0;

}

#include <stdio.h>

#include <string.h>

int main() {

char str1[] = "Sample string";

char str2[40];

char str3[40];

char str4[7];

strcpy(str2, str1);

strcpy(str3, "copy successful");

printf("str1: %s\nstr2: %s\nstr3: %s\n", str1, str2, str3);

strncpy(str4, str1, 6);

str4[6] = '\0';

printf("%s\n",str4);

return 0;

}

[문자열 병합]

#include <stdio.h>

#include <string.h>

int main() {

char str[80];

char str1[20];

char str2[20];

strcpy(str, "Hello ");

strcat(str, "World ");

strcpy(str1, "I want");

strcpy(str2, " go home");

strncat(str1, str2, 8);

puts(str);

puts(str1);

return 0;

}

[메모리 / 문자열 비교]

#include <stdio.h>

#include <string.h>

int main() {

char str1[30];

char str2[30];

printf("문자열 입력 1: ");

scanf("%s",str1);

printf("문자열 입력 2: ");

scanf("%s",str2);

if(!strcmp(str1,str2)){

puts(“문자열 일치");

}

else{

puts("문자열 불일치");

if(strncmp(str1,str2,5))

puts("그러나 앞의 다섯 글자 일치");

}

return 0;

}

[문자열 탐색]

#include <stdio.h>

#include <string.h> // strchr 함수가 선언된 헤더 파일

int main()

{

char s1[30] = "A Garden Diary"; // 크기가 30인 char형 배열을 선언하고 문자열 할당

char \*ptr;

ptr = strchr(s1, 'a'); // a 검색

printf("%d\n",strlen(s1));

printf("%s\n", ptr); // 검색된 문자열 출력

printf("%d\n", ptr-s1+1); // 검색된 문자열 위치

return 0;

}

#include <stdio.h>

#include <string.h>

int main() {

char str[] = "- This, a sample string.";

char\* pch;

printf("Splitting string \"%s\" into tokens:\n", str);

pch = strtok(str, " ,.-");

while (pch != NULL) {

printf("%s\n", pch);

pch = strtok(NULL, " ,.-");

}

return 0;

}

Tip)

NULL 을 전달하며 함수를 호출하게 된다면 이전에 찾았던 토큰 바로 다음 문자 부터 시작하여 토큰들의 위치를 찾게 된다.