문자 분류 데이터 변환 관련 라이브러리

한국기술교육대학교
온라인평생교육원

학습내용

- 라이브러리 함수 이해
- 라이브러리 함수 활용

학습목표

- 문자열 처리 관련 라이브러리의 종류를 설명할 수 있다.
- 문자열을 복사·연결하는 라이브러리 함수의 용도를 알고 구현할 수 있다.

라이브러리 함수 이해

1 종류

1 헤더 파일 : string.h

1 길이함수

strlen	문자열의 길이를 반환

```
int count=0;
while(str[i]!=NULL)
{
   count++;
   i++;
}
```

2 복사함수

memcpy	메모리 블록을 복사
memmove	메모리 블록을 이동
strcpy	문자열을 복사
strncpy	문자열 개수를 지정하여 복사

라이브러리 함수 이해

1 종류

1 헤더 파일 : string.h

3 연결함수

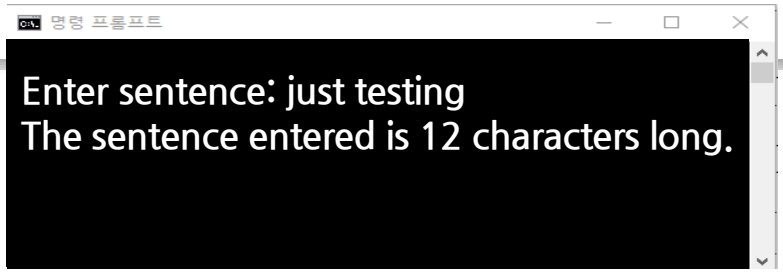
strcat	문자열을 연결
strncat	문자열 개수를 지정하여 연결

라이브러리 함수 활용

1 길이함수

항목	내용
함수원형	size_t strlen(const char *str)
헤더	string.h
기능	문자열의 길이를 구함
매개변수	char *str ➡ 길이를 구할 문자열
반환값	문자열 길이를 바이트 단위로 반환

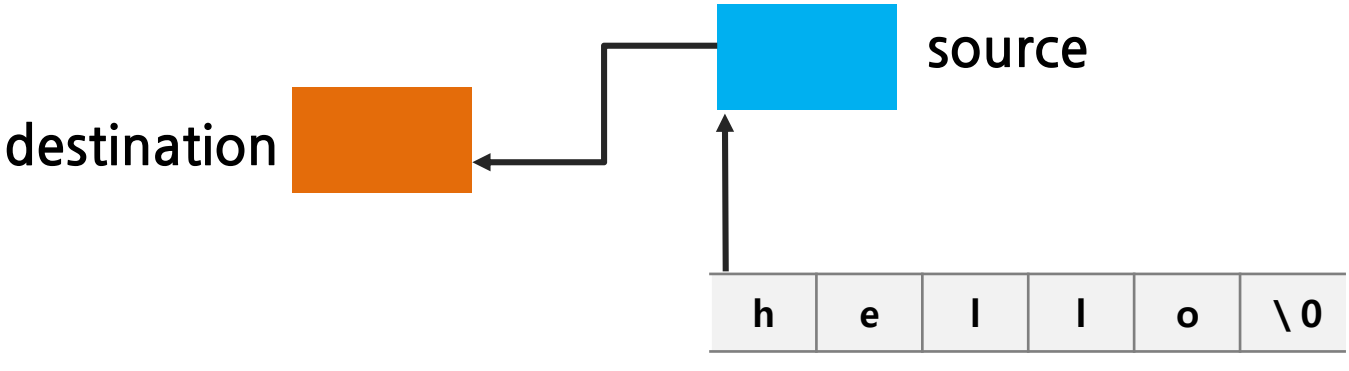
```
#include <stdio.h>
#include <string.h>

int main ()
{
  char szInput[256];
  printf ("Enter a sentence: ");
  gets (szInput);
  printf ("The sentence entered is %u characters long.₩n",(unsigned)strlen(szInput));
  return 0;
}
```



라이브러리 함수 활용

2 복사함수

항목	내용
함수원형	void *memcpy (void *destination, const void *source, size_t num);
헤더	string.h
기능	메모리 영역을 복사(자기 자신은 복사 불가)
매개변수	void *destination ➡ 복사될 메모리의 포인터 void *source ➡ 복사할 메모리의 포인터 size_t num ➡ 복사할 바이트 개수
반환값	void *destination 포인터를 반환, 실패하면 NULL을 반환



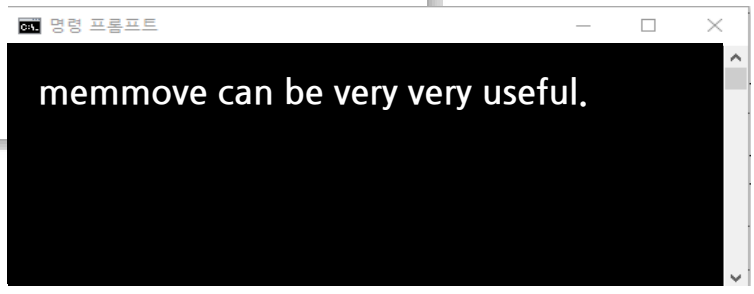
라이브러리 함수 활용

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

int main(void)
{
  char  *ptr_sour="Good Morning!";
  char  *ptr_dest;


  ptr_dest=(char*)malloc(20);
  memcpy(ptr_dest, ptr_sour, strlen(ptr_sour)+1);  //NULL까지 포함하기 위해 +1

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

  free(ptr_dest);

  return 0;
}
```

항목	내용
함수원형	void * memmove (void * destination, const void * source, size_t num);
헤더	string.h
기능	메모리 영역을 복사(자기 자신 복사 가능)
매개변수	void *destination ➧ 복사될 메모리의 포인터 void *source ➧ 복사할 메모리의 포인터 size_t num ➧ 복사할 바이트 개수
반환값	void * destination 포인터를 반환, 실패하면 NULL을 반환

라이브러리 함수 활용

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

int main ()
{
  char str[] = "memmove can be very useful.....";
  memmove (str+20,str+15,11);
  puts (str);
  return 0;
}
```

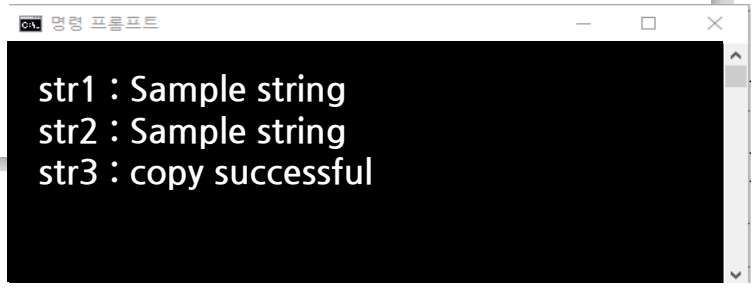


0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
m	e	m	m	o	v	e		c	a	n		b	e		v	e	r	y		u	s	e	f	u	l	.	.	.	.

항목	내용
함수원형	char * strcpy (char * destination, const char * source);
헤더	string.h
기능	문자열을 복사
매개변수	char *destination ➧복사할 위치 char *source ➧원문 문자열
반환값	복사된 문자열을 반환

라이브러리 함수 활용

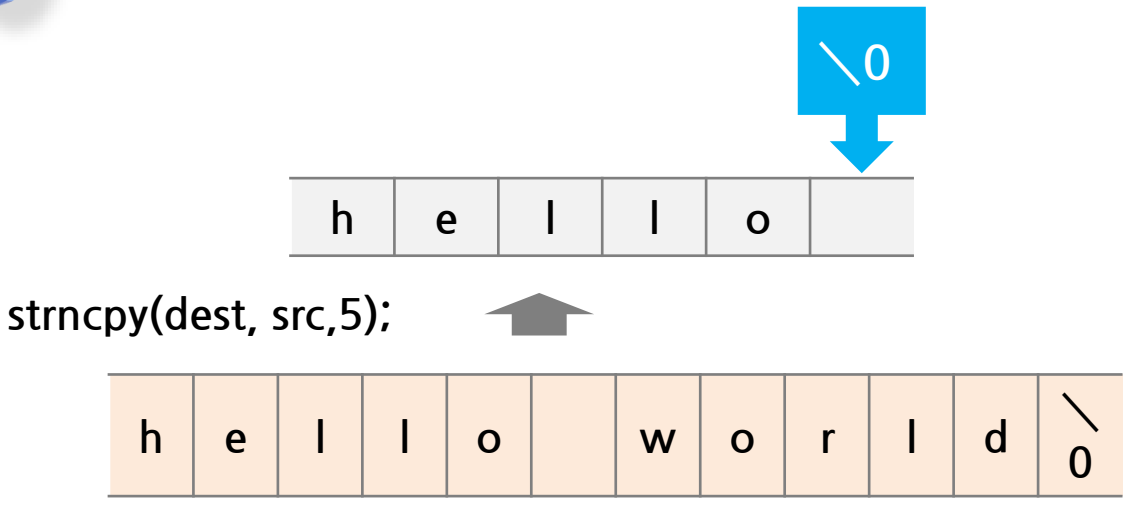
```
#include <stdio.h>
#include <string.h>

int main ()
{
  char str1[]="Sample string";
  char str2[40];
  char str3[40];
  strcpy (str2,str1);
  strcpy (str3,"copy successful");
  printf ("str1: %s₩nstr2: %s₩nstr3: %s₩n",str1,str2,str3);
  return 0;
}
```



항목	내용
함수원형	char * strncpy (char * destination, const char * source, size_t num);
헤더	string.h
기능	source의 첫 번째 문자부터 지정한 개수만큼 문자열 복사
매개변수	char *destination ➧ 문자열 복사 대상 char *source ➧원본 문자열 size_t num ➧복사할 문자 개수
반환값	복사된 문자열을 반환

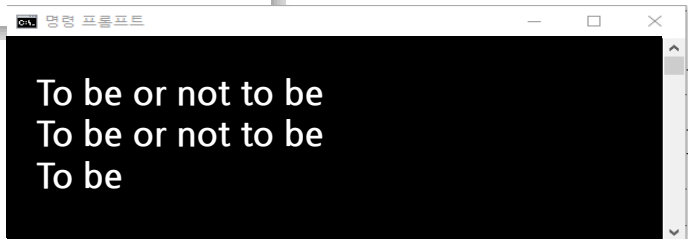
라이브러리 함수 활용

2 복사함수




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

int main ()
{
  char str1[]= "To be or not to be";
  char str2[40];
  char str3[40];

  /* copy to sized buffer (overflow safe): */
  strncpy ( str2, str1, sizeof(str2) );

  /* partial copy (only 5 chars): */
  strncpy ( str3, str2, 5 );
 str3[5] = '₩0';   /* null character manually added */
```

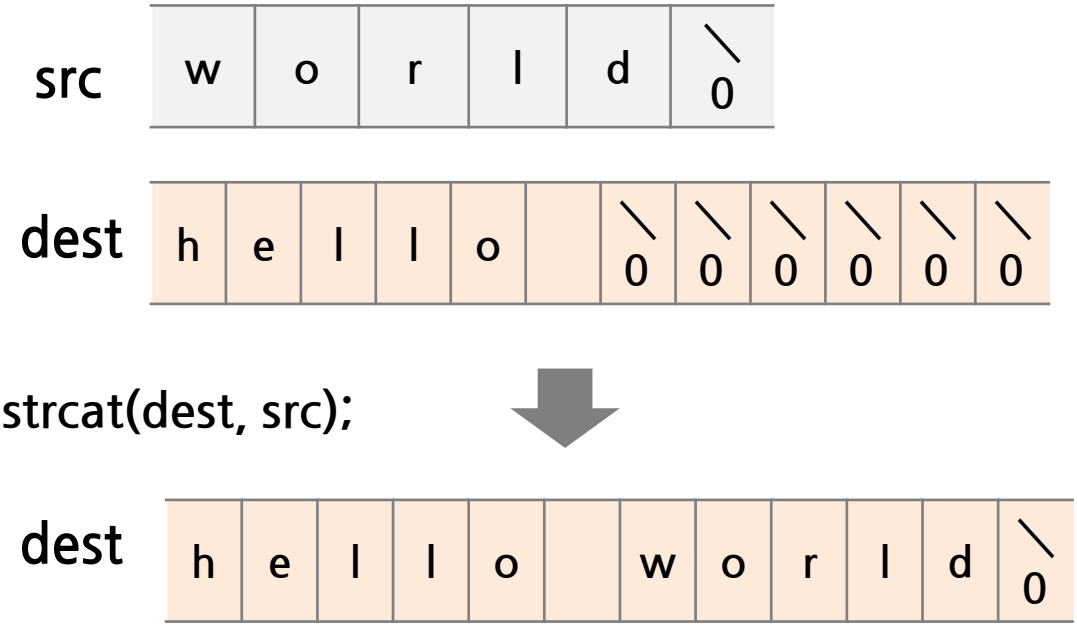
```
puts (str1);
  puts (str2);
  puts (str3);

  return 0;     }
```



라이브러리 함수 활용

3 연결함수

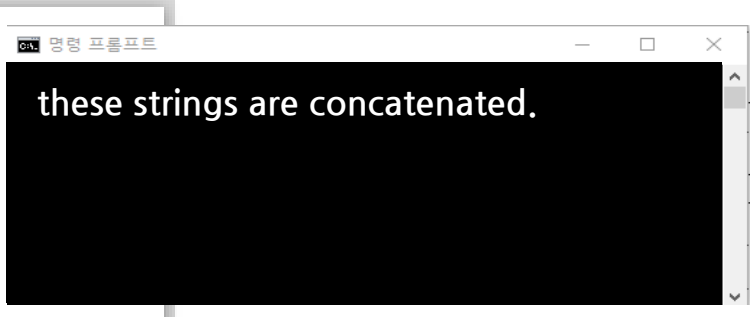
항목	내용
함수원형	char * strcat (char * destination, const char * source);
헤더	string.h
기능	source의 문자열을 destination에 연결
매개변수	char *destination ➡ 연결될 문자열 char *source ➡ 연결할 문자열
반환값	연결된 문자열 반환



라이브러리 함수 활용

3 연결함수

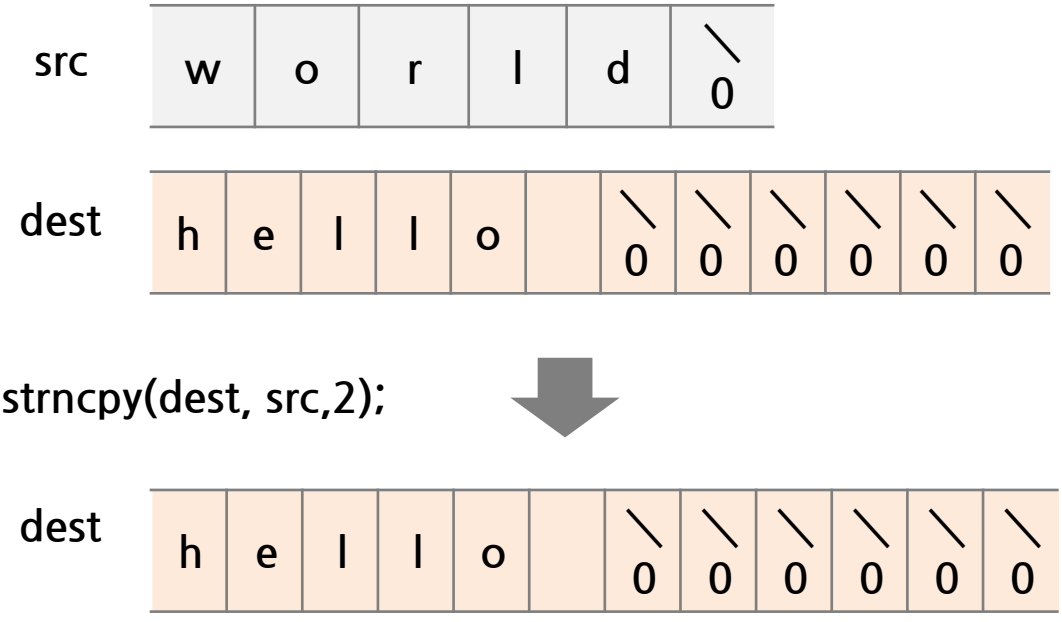
```
#include <stdio.h>
#include <string.h>

int main ()
{
  char str[80];
  strcpy (str,"these ");
  strcat (str,"strings ");
  strcat (str,"are ");
  strcat (str,"concatenated.");
  puts (str);
  return 0;
}
```



항목	내용
함수원형	char * strncat (char * destination, const char * source, size_t num);
헤더	string.h
기능	source의 첫 번째 문자부터 지정한 개수만큼 문자열을 destination에 연결
매개변수	char *destination ➧ 연결될 문자열 char *source ➧ 연결할 문자열 size_t num ➧ 연결할 문자 개수
반환값	연결된 문자열 반환

라이브러리 함수 활용

3 연결함수




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

int main ()
{
  char str1[20];
  char str2[20];
  strcpy (str1,"To be ");
  strcpy (str2,"or not to be");
  strncat (str1, str2, 6);
  puts (str1);
  return 0;
}
```

학습정리

1. 라이브러리 함수 이해

- 문자열 처리 관련 함수는 string.h를 include함
- 문자열의 길이를 반환하는 함수 : strlen()
- 문자열을 복사하는 함수 : memcpy, memmove, strcpy, strncpy
- 문자열을 연결하는 함수 : strcat, strncat

2. 라이브러리 함수 활용

- 문자열 포인터를 복사하는 함수 : memcpy, memmove
  - memmove 함수는 자기 자신을 복사하는 것이 가능함
- 문자열을 복사하는 함수 중 개수를 지정할 수 있는 함수 : strncpy
- 문자열을 연결하는 함수 중 개수를 지정할 수 있는 함수 : strncat