Strcpy, strncpy

**헤더** 🡺 #include <string.h>

**인터페이스(규격)**

Char\* strcpy (char\* restrict dest, const char\* src);

Char\* strncpy (char\* restrict dest, const char\* restrict src, size\_t n);

**설명**

1. Strcpy는 NULL 바이트(‘\0’) 종료 바이트를 포함하여 src가 가리키는 문자열을 dest가 가리키는 버퍼로 복사한다.
2. 문자열은 겹치지 않을 수도 있으며, dest는 복사본을 가져갈 만큼의 충분한 크기를 지녀야한다.
3. 버퍼 오버런을 조심해야한다.
4. Strncpy는 strcpy와 유사하지만 최대 n 바이트의 src가 복사된다는 점이 다르다.
5. Src에 처음 n 바이트에 null 바이트가 없다면 dest에 들어간 문자열이 null로 종료되지 않게 한다
6. Src의 길이가 n보다 작으면 dest에 null바이트들을 추가로 채워 총 n 바이트를 복사한다.

🡺 여기서 나오는 null 바이트는 NULL이 아니라 NUL 이다.

🡺 NUL은 문자상수 0으로 ASCII 값이 0 이다

* 숫자 0 은 ASCII 48이다.

🡺 NUL은 char형이고 고로 1byte이다. 즉 1바이트씩 빈공간을 채워 n바이트를 만든다.

**RETRUN**

Strcpy() 및 strncpy() 함수는 대상 문자열 dest에 대한 포인터를 반환한다.

Size\_t형

Unsigned int 🡺 0이상 양수만 표현 가능한 정수이다.

32비트 운영체제 🡪 unsigned 32비트 정수

64비트 운영체제 🡪 unsigned 64비트 정수로 나타낸다.

**But.** 무조건 위와 같은건 아니다.

64비트os에서도 컴파일러에 따라 32비트의 정수를 나타낼 수 있다.

그렇다면 왜 Size\_t 형으로 매개변수를 선언할까?

🡺 이유는 두가지인데 첫번째는 size\_t 형은 typedef로 unsigned int 자료형을 재정의한것이다.

* Typedef = 특정한 명칭으로 자료형을 지정한다.

🡺 두번째 이유는 현재는 int가 4byte지만 과거에는 2byte였고 이후에는 8byte가 될 수 도 있다. 하지만 헤더에서 size\_t 형으로 고정해놓는다면 size\_t형의 크기는 고정이된다. 즉, 운영체제에 따라 해당 변수의 메모리 할당 공간이 달라질 수 있는 것을 size\_t로 고정시킬 수 있다.

🡺 고로 매개변수에 size\_t 형의 변수가 선언되는 함수를 보게 된다면 이 변수에는 양의 정수를 넣어야 한다는 것을 알려주기 위한 형태이다. 만약 내가 넣어야 하는 정수 값이 양수임을 이미 알고 있다면 size\_t로 표현하지 않아도 된다. 무조건 size\_t로 선언하지 않아도 된다는 것이다.