

url : [https://www.youtube.com/watch?v=LmwycZ393r4&list=PLz-ENLG\\_8TMdMJlwyqDlpcEOysvNoonf&index=14](https://www.youtube.com/watch?v=LmwycZ393r4&list=PLz-ENLG_8TMdMJlwyqDlpcEOysvNoonf&index=14)

## ● 표준함수 만들기

리눅스에서는 비주얼 스튜디오와 다르게 sdi 경고를 해제하고, 보안경고가 뜨는 함수를 쓸 필요가 없다.  
예를 들어서, scanf, strcpy, strcat 등등 보안 오류를 발생하는 함수들을 마음껏 사용할 수 있다.

그래서, 간단하게 string.h 에 있는 표준함수 3개를 만들어 보았다

### 1. strcpy

먼저, strcpy의 함수원형이 리눅스 매뉴얼에 이렇게 적혀있다.

```
#include <string.h>

char *strcpy(char *dest, const char *src);
```

이 함수는 src의 문자열을 dst로 통째로 복사하는 함수이다.

보안문제는 dst 문자열을 담는 메모리공간이 src보다 작으면, 잘못된 접근으로 OS의 철퇴를 맞는다.  
그러나, 리눅스에선 이러한 점까지 생각하지 않고 순수 기능만 구현하였다.

```
char* string_copy(char * dst, char * src)
{
    while(*src) // ascii에서 0 은 null
    {
        *dst++ = *src++;
    }

    return dst;
}
```

이렇게 간단하게 구현하였다.

### 2. strlen

먼저, strlen의 함수원형을 리눅스 매뉴얼에서 확인하였다.

```
#include <string.h>

size_t strlen(const char *s);
```

이 함수는 문자열의 길이를 출력하는 함수이다.

그러나, NULL 포인터 변수를 받으면 -1을 출력한다.

이 함수는 딱히 보안문제를 발생시키지 않는다. 그래서, 이 함수는 대체함수로 strlen\_s 같은 함수가 없다.

```

size_t string_length(const char *str)
{
    int cnt = 0;
    if(str != NULL)
    {
        while(1)
        {
            if(str[cnt++] == 0)
                break;
            else
                cnt++;
        }
    }
    else
        return -1;
    return cnt;
}

```

이렇게 간단하게 구현하였다.

### 3. strcmp

먼저, strcmp의 함수원형을 리눅스 매뉴얼에서 확인하였다.

```

#include <string.h>

int strcmp(const char *s1, const char *s2);

```

이 함수는 문자열이 서로 똑같으면 0을 출력하고, 그렇지 않으면, 틀린부분에서 아스키코드값을 뺀 정수값을 출력한다.  
이 역시 strcmp\_s 같은 대체함수가 없어서 안전하다.

```

int string_compare(const char *str1, const char *str2)
{
    // 두 문자열의 각 문자들을 빼서, 0이 아니면 그 차이를 함수로 출력

    int idx = 0;

    while(str1[idx] != 0 && str2[idx] != 0)
    {
        if(str1[idx] - str2[idx])
            return str1[idx] - str2[idx]; // 두 문자가 서로 다를때, 그 차이를 출력
        idx++;
    }

    return 0; // 두 문자열이 완벽히 똑같은
}

```

이렇게 간단하게 구현하였다.