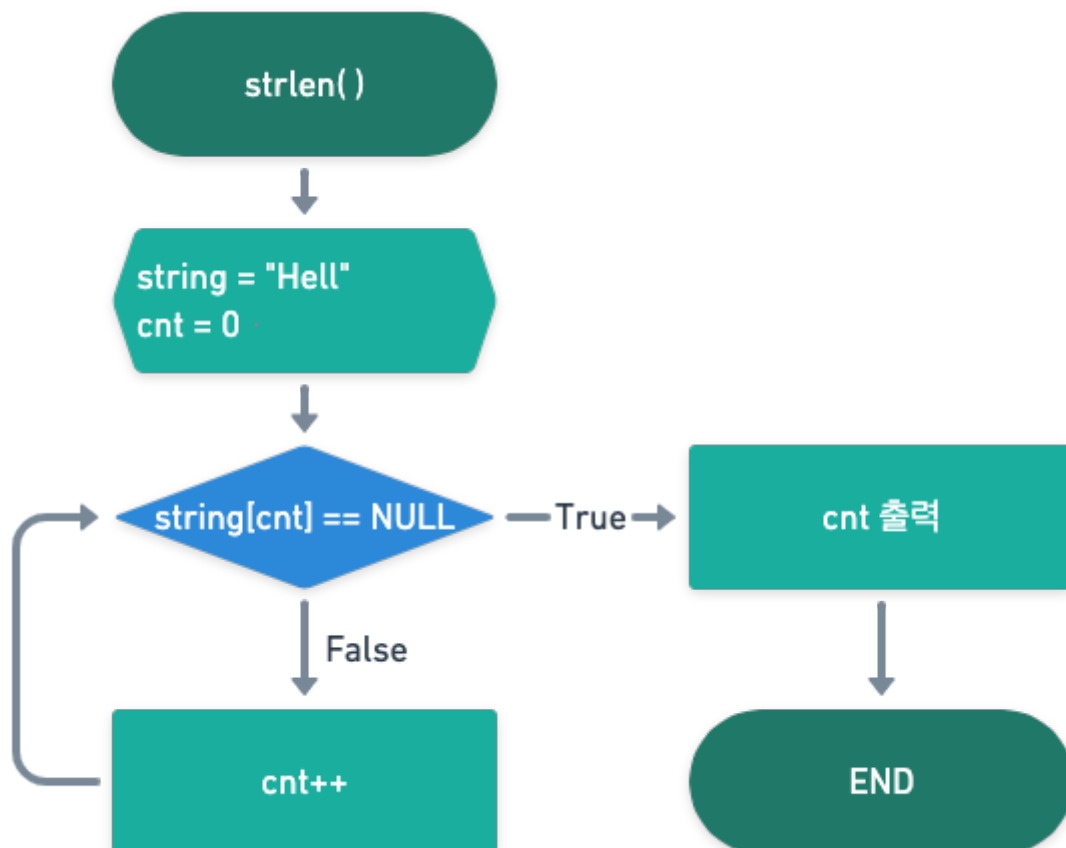


strlen() 구현

>_ 입력 : 한 줄의 문자열
출력 : 문자열의 길이

- CMP (CMP destination source) : dest - src를 계산하여, ZF와 CF 값에 따라서 분기함
 - dest < src : ZF = 0, CF = 1
 - dest = src : ZF = 1, CF = 0
 - dest > src : ZF = 0, CF = 0
- JMP (JMP label) : 미리 정해 둔 라벨로 점프함



프로그램의 순서도

```

1 strlen.asm
;file: strlen.asm
1 ;문자열을 입력받고 길이를 출력하는 프로그램
2
3 section .text
4 global _start
5
6 _start:
7 xor rax, rax
8 mov rbx, rax
9 mov rcx, rax
10 mov rdx, rax
11
12 sub rsp, 64 ;입력 받는 부분
13 mov rdi, 0
14 mov rsi, rsp
15 mov rdx, 63
16
17 syscall
18
19 xor r9, r9
20 while:
21 mov cl, [rsp + 1 + r9] ;문자열을 1 바이트씩 받아옴
22 cmp cl, 0x00 ;NULL인지 확인
23 je break ;NULL 이면 break
24 inc r9 ;cnt++
25 jmp while ;반복
26 break:
27
28 add r9, 0x30 ;숫자로 바꾸기 위해 '0' 추가
29
30 mov rax, 1 ;출력 부분
31 mov rdi, 1
32 mov [rsi], r9
33 mov rdx, 1
34
35 syscall
36
37 mov rax, 60 ;종료 부분
38
39 syscall
40

```

입, 출력부는 동일하므로 생략

;line 19 - 28 : 문자열 길이 측정하기

```

xor r9, r9
while:
    mov cl, [rsp + 1 + r9]
    cmp cl, 0x00
    je break
    inc r9
    jmp while
break:

add r9, 0x30

```

r9 (카운트)를 초기화하고, 문자열을 1byte씩 받아오기 위해 문자열의 시작부터 cl (8-bit register)에 저장한다. 그리고 이 문자가 NULL인지 비교하고 NULL이면 break 라벨로 점프, NULL이 아니라면 카운트를 1 증가시킨다. 반복문이 끝나면, 카운트에 '0'의 아스키 코드값 0x30을 더해 한 자리 숫자로 변환시킨다.

```
~/Desktop/Assembly  
> ./strlen  
Hell  
4  
~/Desktop/Assembly  
> ./strlen  
Isaac  
5  
~/Desktop/Assembly 14s  
> █
```

10 이상 길이의 문자열

→ 카운트 / 10 이 0이 아닌지 확인하고 0이 아니라면 숫자로 바꾼 뒤 출력, 몫 * 10 만큼 빼고 나머지 값을 숫자로 바꾼 뒤 출력.. 을 하면 될 것 같다..