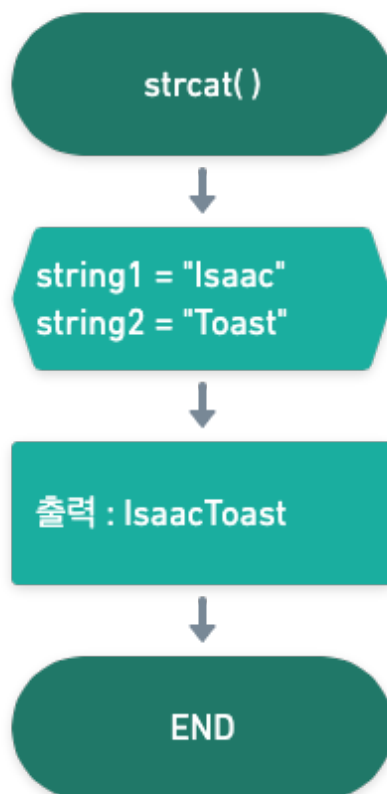


# strcat( ) 구현



입력 : 두 줄의 문자열  
출력 : 두 문자열을 연결



프로그램의 순서도

```

1 strcat.asm
2 ;file: strcat.asm
3 ;입력받은 문자열 2개를 이어 출력하는 프로그램
4
5 section .text
6 global _start
7
8 _start:
9     xor rax, rax           ;rax, rbx, rcx, rdx 초기화
10    mov rbx, rax
11    mov rcx, rax
12    mov rdx, rax
13
14    sub rsp, 128           ;첫번째 문자열 입력
15    mov rdi, 0
16    mov rsi, rsp
17    mov rdx, 63
18    syscall
19
20    mov al, 0
21    add rsp, 64           ;키보드 버퍼 플러쉬 후 두번째 문자열 입력
22    mov rdi, 0           ;rsp를 이동시켜 후에 순서대로 출력될 수 있게 함
23    mov rsi, rsp
24    mov rdx, 63
25    syscall
26
27    mov al, 0
28    sub rsp, 64           ;다시 첫번째 문자열이 있던 위치로 rsp 이동
29    xor r9, r9           ;r9 초기화
30
31    while:
32        mov cl, [rsp + 1 + r9] ;cl register로 문자열 처음부터 읽어옴
33        cmp cl, 0x0A         ;줄간에 들어간 개행문자 확인
34        jz break           ;개행문자라면, 반복문 탈출
35        inc r9
36        jmp while
37
38    break:
39        mov BYTE[rsp + 1 + r9], 0x00 ;개행문자 NULL로 변경
40
41        mov rax, 1
42        mov rdi, 1
43        mov rsi, rsp
44        mov rdx, 127
45        syscall
46
47        mov rax, 60
48        syscall
49
50    ;프로그램 종료
51

```

sys\_read는 치고 있는 한 문자만 입력 버퍼에 들어간다. 따라서 첫 번째 sys\_read가 끝나도, \n은 입력 버퍼에 남아 두 번째 sys\_read까지도 영향을 미치게 된다.

```

;line 18
mov al, 0

```

이를 해결하기 위해 OS의 키보드 버퍼를 플러쉬, 즉 al을 0으로 초기화하였다.

(?) 왜 al에 0을 대입했을 때 입력 버퍼를 비울 수 있나요?

```

;line 26 - 36
sub rsp, 64
xor r9, r9
while:
    mov cl, [rsp + 1 + r9]
    cmp cl, 0x0A
    jz break
    inc r9
    jmp while

break:
    mov BYTE[rsp + 1 + r9], 0x00

```

두 문자열을 하나로 합치기 위해 문자열 사이에 입력된 **개행 문자를 지울 필요**가 있었다. 개행 문자의 아스키 코드는 0x0A 이므로, strlen( ) 에서 NULL 문자를 찾을 때처럼 개행 문자를 찾아 NULL로 바꿔 두 문자열을 하나로 합칠 수 있었다.

```
~/Desktop/Assembly
> ./strcat
Isaac
Toast
IsaacToast

~/Desktop/Assembly 7s
> ./strcat
One
Top
One Top

~/Desktop/Assembly
> █
```

공백 문자까지 빠짐없이 들어가 두 문자열을 이어주는 것을 확인할 수 있다.