Str_length Procedure

The Str_length procedure returns the length of a string in the EAX register. It takes one argument: a pointer to the string.

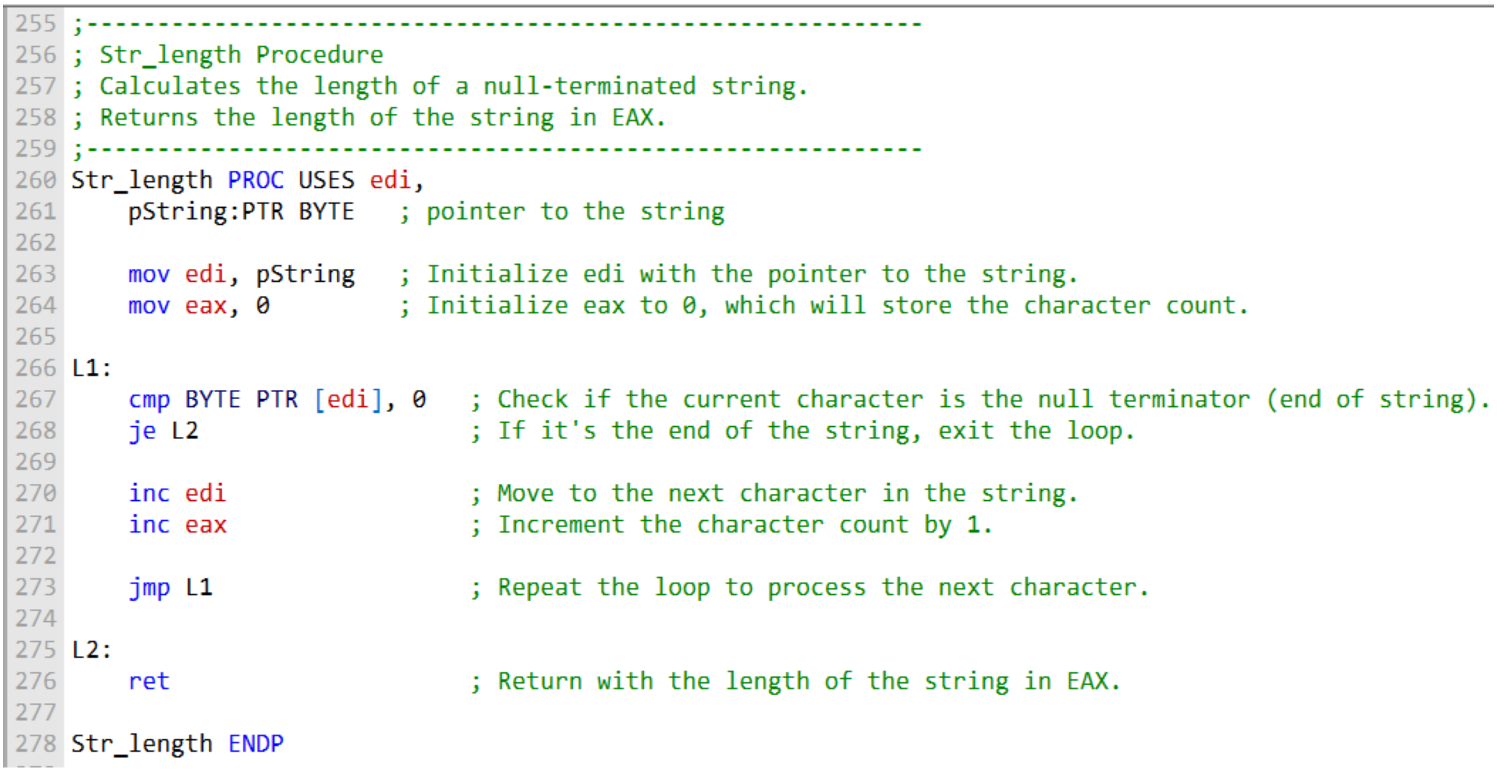
The Str_length procedure works by scanning the string byte by byte until it reaches the null terminator.

The procedure increments the EAX register for each byte in the string. After reaching the null terminator, the procedure returns the value in the EAX register.

The following is a more detailed explanation of the Str_length procedure:

This code works by scanning the string byte by byte until it reaches the null terminator. The code increments the EAX register for each byte in the string. After reaching the null terminator, the code returns the value in the EAX register.

To use the Str_length procedure, you would pass the address of the string to the procedure as an argument.



```
;----------------------------------------------------------
; Str_length Procedure
; Calculates the length of a null-terminated string.
; Returns the length of the string in EAX.
;----------------------------------------------------------
Str_length PROC USES edi,
    pString:PTR BYTE   ; pointer to the string

    mov edi, pString   ; Initialize edi with the pointer to the string.
    mov eax, 0         ; Initialize eax to 0, which will store the character count.

L1:
    cmp BYTE PTR [edi], 0   ; Check if the current character is the null terminator (end of string).
    je L2                   ; If it's the end of the string, exit the loop.

    inc edi                 ; Move to the next character in the string.
    inc eax                 ; Increment the character count by 1.

    jmp L1                  ; Repeat the loop to process the next character.

L2:
    ret                     ; Return with the length of the string in EAX.

Str_length ENDP
```

Line 1: The Str_length procedure pushes the EDI register onto the stack. This is necessary because the procedure uses this register.

Line 2: The Str_length procedure moves the pointer to the string into the EDI register.

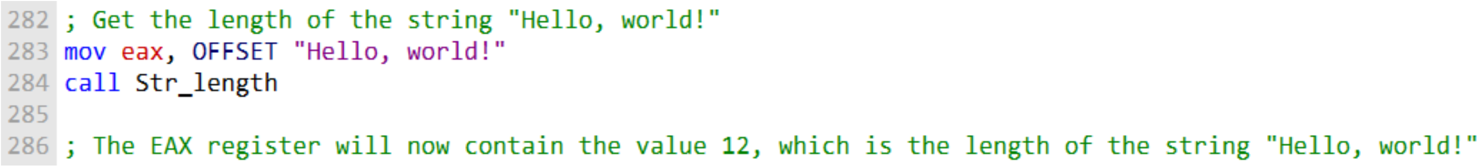
Line 3: The Str_length procedure moves the value 0 into the EAX register. This will be used to store the length of the string.

Line 4: The Str_length procedure enters a loop. On each iteration of the loop, the following steps are performed:

- The Str_length procedure compares the byte at the memory location pointed to by the EDI register to the null terminator (0x00).
- If the byte is equal to the null terminator, the Str_length procedure exits the loop.
- If the byte is not equal to the null terminator, the Str_length procedure increments the EDI register and the EAX register.

Line 10: The Str_length procedure exits the loop and returns to the caller.

The following is an example of how to use the Str_length procedure:



```
; Get the length of the string "Hello, world!"
mov eax, OFFSET "Hello, world!"
call Str_length

; The EAX register will now contain the value 12, which is the length of the string "Hello, world!"
```