Assembly Language and System Programming Lab9 Report 2

Group: 2

Names: 林柏廷(Brian)、蔡淵丞(Vincent)

Student ID: 110502531、110502567

**Objective:**

Create a procedure that delete n characters from the beginning of a given string.

**Data:**

These strings are the target strings from which n characters will be removed.

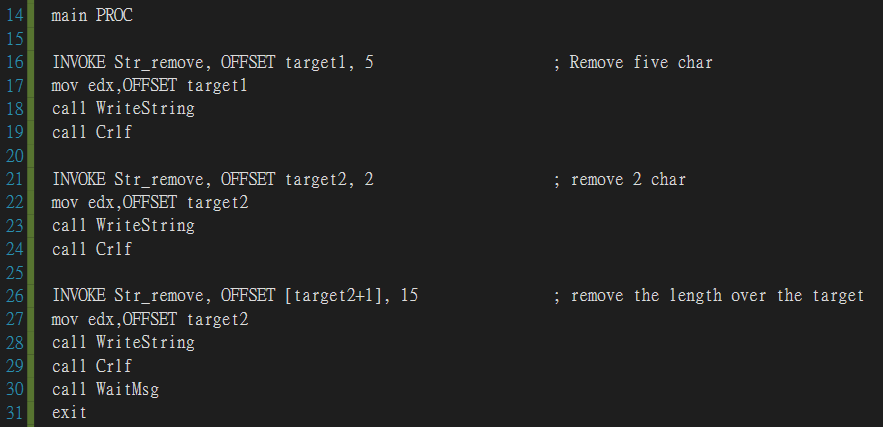


**main PROC:**

1. Uses the procedure Str\_remove to remove first 5 characters starting from the first character of target1. Store the location of the string target1 in ***edx***, then print it out. Finally, print an empty line.

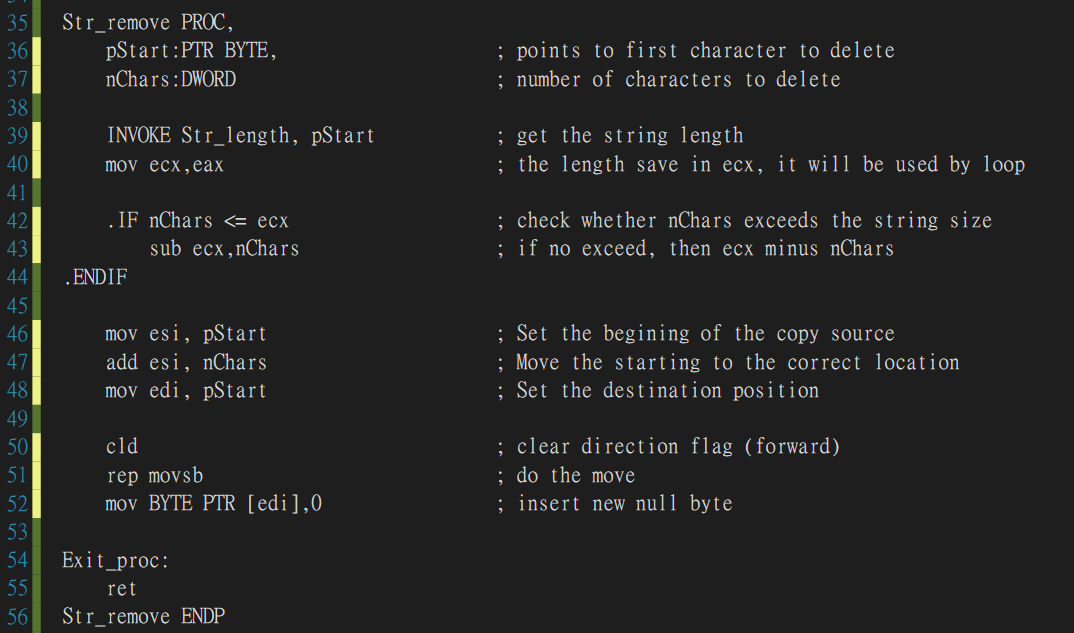
2. Uses the procedure Str\_remove to remove first 2 characters starting from the first character of target2. Store the location of the string target1 in ***edx***, then print it out. Finally, print an empty line.

3. Uses the procedure Str\_remove to remove first 15 characters starting from the second character of target2. Store the location of the string target1 in ***edx***, then print it out. Finally, print an empty line.



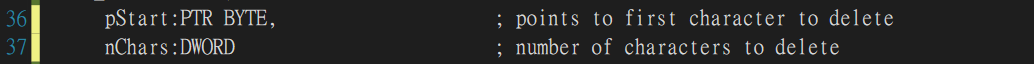
**Str\_remove:**

Remove first n character of the input string by copying the (n+1)st character to the first character, the (n+2)st to the second...etc.



**Line 36-37:**

Define two parameters. pStart points to the first character to be deleted, nChars stores the number of characters to be deleted.



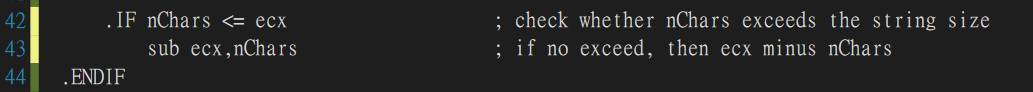
**Line 39-40:**

Get the string length using the procedure Str\_length, then store it in ***ecx***, which will be used by the instruction rep.



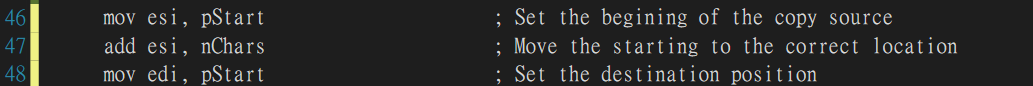
**Line 42-44:**

If nChars is smaller than ecx, which is the length of the string, subtract ***ecx*** by nChars. Since the number of repetitions should be length – n.



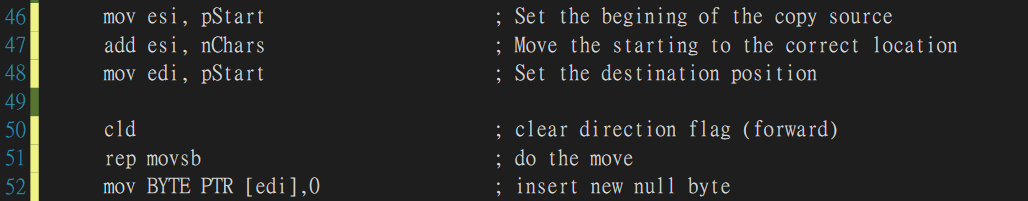
**Line 46-48:**

Store pStart in ***esi***, then add by nChars. Since we want the position of the source of copying to be length + n. Move pStart to ***edi***, since the position of the destination should be the first character.



**Line 50-52:**

Use the instruction cld to clear the direction flag since we will move forward. The instruction “rep movsb” would move the byte pointed to by ***esi*** to the byte pointed to by ***edi***, then repeat by ***ecx*** times. After these two instructions, we have removed first n characters of the target string. The last instruction inserts a null byte to the end of the string.



**Result:**

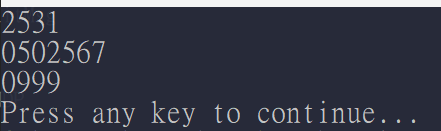
110502531 remove 5 characters-> 2531

110502567 remove 2 characters-> 0502567

The result of removing 15 characters from [target2 + 1] needs further explanations.

Target2 after first two invocations is.

As you can see, the relative position of target2 and target3 is same as it was before removing 2 characters from target2. This is because we didn’t actually create a new string with first n characters removed, we simply copy the last (length – n) characters to the beginning of the string. As a result, remove 15 characters from [target2 + 1] would remove “502567.7.999999” which would be 0999.



**Review:**

In this lab and lecture, we’ve learned how to use movsb instruction, which copies data from the memory location pointed to by ***esi*** to which pointed by ***edi.*** This instruction combined with the instruction rep, which can be inserted before movsb, movsw, ore movsd, can be really useful when manipulating strings. In the third invocation in the code of this lab, we can see that sometimes the result isn’t that obvious, we need to know how the programs work in order to fully understand the results. We can also know it better by examining the memory and registers.

**Full code:**

