Микропроект №1

Выполнил Х. Дилавар Ахмад Шаир Хамаюнович студент БПИ 196 ФКН ВШЭ

Задание

Разработать программу, находящую в заданной ASCII-строке первую справа налево последовательность N символов, каждый элемент которой определяется по условию "больше предшествующего" (N=5)

Источники

- http://www.int80h.org/strlen/ Использование цепочечной функции для нахождения длины строки

Ход Работы

Первым шагом при работе с программой является ввод строки, после того как пользователь выполнит данный шаг, программа из консоли считает введенное значение и начнет его обрабатывать. Сначала она найдет длину строки с помощью функции GetStringLength, которая после своего выполнения сохранит длину строки в регистр eax, после возвращение в основную часть программы, вызывается процедура для поиска подстроки, которая в случае удачного выполнения будет запишет значение 5 в регистр edx и в память запишет адрес символа, с которого начинается подстрока, а в случае, если подстроки нет, то в консоль выведется строка (could not find the necessary substring),

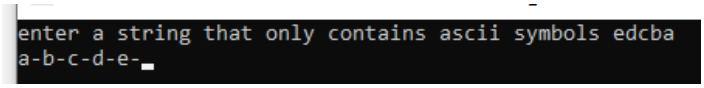
```
format PE console
entry start

; Разработать программу, находящую в заданной ASCII-строке
; первую справа налево последовательность из 5 символов,
; каждый элемент которой определяется по условию "больше предшествующего"

include 'win32ax.inc'

section '.data' data readable writable
        enter_string          db 'enter a string that only contains ascii symbols ', 0
        didNotFindStr         db 'could not find the necessary substring ', 0
        printChar             db '%c-', 0
        printInt              db '%d ', 0
        read_string           db '%s', 0
        notAsciiOutput        db 'You have entered a string that contains symbols outside the ascii table', 0
        stroka                rd 64
        firstElement          dd 0
        A                     dd 0
        B                     dd 0
        stringLength          dd 0

section '.code' code readable executable
start:
        mov     [A], 5
        mov     [B], 0
        push    enter_string
        call    [printf]
        add     esp, 4

        push    stroka
        push    read_string
        call    [scanf]
        add     esp, 8

        lea     edi, [stroka]       ; заносим абсолютный адрес в edi
        call    GetStringLength     ; заносит длину строки в регистр eax
        mov     [stringLength], eax ; сохраняем длину строки
```

```
        call    FindSubstring             ;abcdefgfedcba

        cmp     edx, 0
        je      didNotFindSusbtring

        cmp     edx, 0
        jne     substringFoundPrint

GetStringLength:
        push    ecx       ; сохраняем значение ecx
        push    edi       ; сохраняем значение edi

        sub     ecx, ecx  ; зануляем счетчик
        mov     ecx, -1   ; задаем макс значение (-1 -> 11111111..)
        sub     al, al
        cld
        repne   scasb     ; сраваниваем текущий символ в строке с 0 (al),
                          ; повторяем, пока <>
        neg     ecx       ; меняем все 0 на 1 и наобророт
        sub     ecx, 2    ; вычитаем 1, тк мы начали с -1
        mov     eax, ecx  ; реузльтат функции запишем в eax

        pop     ecx       ; восстанавливаем значение eax
        pop     edi       ; восстанавливаем значение edi

        ret

FindSubstring:
        sub     ebx, ebx
        sub     ecx, ecx
        sub     edx, edx
        mov     ebx, [stringLength]
        ;mov     ecx, [stroka+ebx-1] ; last char

        mov     ecx, stroka
        add     ecx, ebx
        dec     ecx

findSubstringLoop:
        mov     [A], ecx
        mov     [B], edx
        cmp     edx, 5
        je      loopBreak

        cmp     ecx, stroka-1
        je      loopBreak

        mov     eax, [ecx-1]
        mov     ebx, [ecx]

        cmp     eax, ebx
        jg      greater

        cmp     eax, ebx
        jng     notGreater

notAscii:
        push notAsciiOutput
        call [printf]
        add esp, 4
        jmp exit

greater:
        cmp     edx, 0
        je      firstFound
        mov     edx, [B]
        inc     edx

        mov     ecx, [A]
        dec     ecx
        jmp     findSubstringLoop

notGreater:
        sub     edx, edx
        mov     edx, 0
```

```
        mov     ecx, [A]
        mov     [firstElement],0
        dec     ecx
        jmp     findSubstringLoop

firstFound:
        mov     edx, [B]
        mov     edx, 1
        mov     ecx, [A]
        mov     [firstElement],ecx
        jmp     findSubstringLoop

loopBreak:
        ret

didNotFindSusbtring:
        push    didNotFindStr
        call    [printf]
        add     esp, 4

        jmp     exit

substringFoundPrint:
        sub     ecx, ecx
        mov     ecx, 0
        mov     ebx, [firstElement]
printLoop:
        cmp     ecx, 5
        je      exit

        mov     [firstElement], ebx
        mov     [A], ecx

        sub     ebx, ecx
        mov     ecx, [ebx]
        push    ecx
        push    printChar
        call    [printf]
        add     esp, 8

        mov     ecx, [A]
        inc     ecx

        mov     ebx, [firstElement]
        jmp     printLoop

exit:
        call    [getch]
        push    0
        call    [ExitProcess]

section '.idata' import data readable
    library kernel, 'kernel32.dll',\
            msvcrt, 'msvcrt.dll',\
            user32,'USER32.DLL'

include 'api\user32.inc'
include 'api\kernel32.inc'
    import kernel,\
           ExitProcess, 'ExitProcess',\
           HeapCreate,'HeapCreate',\
           HeapAlloc,'HeapAlloc'
  include 'api\kernel32.inc'
    import msvcrt,\
           printf, 'printf',\
           scanf, 'scanf',\
           getch, '_getch'
```

Тест 1

Ввод простой строки с нужной подстрокой (abcde)



Тест 2

Ввод строки без нужной подстроки