

Федеральное государственное автономное образовательное учреждение
высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук

Направление: программная инженерия

Программирование на языке ассемблера. Микропроект №1. Пояснительная записка

Абдельсалам Шади Мазен, БПИ191

Текст задания

Разработать программу, которая "переворачивает" заданную позициями N1-N2 часть ASCII-строки символов

Список источников

- <http://www.softcraft.ru/edu/comparch/> - лекции и семинары на сайте SoftCraft
- <https://flatassembler.net/docs.php?article=manual> – мануал разработчика FASM
- <https://fasmworld.ru/spravochnik-komand/> - электронный справочник команд FASM

Алгоритм решения и расчётные методы

Для переворота подстроки использовался метод двух указателей

Текст программы

Файл с основной программой **main.asm**

```
format PE console
entry start

include 'win32a.inc'
include 'input_macro.inc'

section '.data' data readable writable
    strScanInt    db '%d', 0
    strScanIntMsg db 'Enter string index: ', 0
    strScanString db '%s', 0
    strScanStringMsg db 'Enter a string: ', 0
    strPrintString db '%s', 10, 0
    strPrintInt    db '%d', 10, 0
    strValueFailLZ db 'Indexes must be greater than or equal to zero,
try again', 10, 0
    strValueFailGL db 'Indexes must be less than string length %d, try
again', 10, 0
    strPrintAnswer db 'Answer string: %s', 10, 0
    strProgramFinished db 'The program is finished, press any key to e
xit', 10, 0
    strNewLine    db 10, 0

    inputString   db 256 dup ?
    ; первая позиция для взятия подстроки
    i             dd ?
    ; вторая позиция для взятия подстроки
    j             dd ?
    ; длина вводимой строки
    strLen        dd ?

section '.code' code readable executable
start:
    InputMacro
reverseSubstring:
    stdcall ReverseSubstring, inputString, [i], [j]
    ; вывод искомой строки
    cinvoke printf, strNewLine
    cinvoke printf, strPrintAnswer, inputString
    cinvoke printf, strProgramFinished
finish:
    cinvoke getch
    push 0
    call [ExitProcess]

; процедура для получения длины строки
```

```

proc GetStringLength _, string
    string equ dword [string] ; вместо string в дальнейшем будет подставля
    тся строковой аргумент процедуры
    enter 0, 0 ; установка кадра стека

    cld ; сброс флага направления
    mov edi, string ; edi = &string
    mov esi, edi ; esi = &string
    mov ecx, 256d ; ecx = максимальная длина строки
    mov al, 0 ; al = символ конца строки
    repne scasb ; повторяем, пока не найдем символ конца строки
    sub edi, esi ; вычитаем из текущего адреса адрес начала строки
    dec edi ; вычитаем из получившейся длины терминальный символ
    mov eax, edi ; eax = длина строки

    leave ; восстановление кадра стека
    ret
endp

proc ReverseSubstring _, string, i, j
    string equ dword[string]
    leftPtr equ dword[i]
    rightPtr equ dword[j]
    enter 0, 0

    mov eax, rightPtr
    sub eax, leftPtr ; длина переворачиваемой подстроки

    mov ecx, string
    add ecx, leftPtr ; указатель на левую границу

    add eax, ecx ; указатель на правую границу
reverseLoop:
    ; сравниваем указатели, если левый становится >= правому, то прерываем
    цикл
    cmp eax, ecx
    jle reverseLoopEnd

    ; меняем значения левого и правого указателей
    mov dl, [eax]
    mov dh, [ecx]
    mov [eax], dh
    mov [ecx], dl

    ; уменьшаем правый указатель
    dec eax
    ; увеличиваем левый указатель
    inc ecx
    jmp reverseLoop
reverseLoopEnd:
    leave ; восстановление кадра стека

```

```

    ret
endp

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'

```

Файл с макросом для ввода и проверки данных **input_macros.inc**

```

include 'utils_macro.inc'

macro InputMacro {
stringInput:
    cinvoke printf, strScanStringMsg
    cinvoke scanf, strScanString, inputString
    stdcall GetStringLength, inputString
    mov [strLen], eax ; сохраняем длину строки в strLen
inputValues:
    ; ввод индексов начала и конца построки
    cinvoke printf, strScanIntMsg
    cinvoke scanf, strScanInt, i
    cinvoke printf, strScanIntMsg
    cinvoke scanf, strScanInt, j

    mov eax, [i]
    mov ecx, [j]

    ; если i > j, то меняем их местами
    cmp eax, ecx
    jg swapVars
    jmp checkValues
swapVars:
    SwapVars i, j
checkValues:

```

```

; проверка меньшего индекса
mov eax, [i]
mov ecx, [strLen]

; если число больше длины строки
cmp eax, ecx
jge valueGreaterThanLength

; если число меньше нуля
cmp eax, 0
jlt valueLessThanZero

; проверка большего индекса
mov eax, [j]

; если число больше длины строки
cmp eax, ecx
jge valueGreaterThanLength

; если число меньше нуля
cmp eax, 0
jlt valueLessThanZero

; продолжение программы
jmp reverseSubstring
valueGreaterThanLength:
    cinvoke printf, strValueFailGL, [strLen]
    jmp valueFailed
valueLessThanZero:
    cinvoke printf, strValueFailLZ
    jmp valueFailed
valueFailed:
    jmp inputValues
}

```

Файл с макросом для перестановки переменных местами **utils_macro.inc**

```

; меняет значения переданных переменных местами
macro SwapVars var1, var2 {
    mov eax, [var1]
    mov ecx, [var2]
    mov [var1], ecx
    mov [var2], eax
}

```