|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования  Российской Федерации | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования | | |
| «Новосибирский государственный технический университет» | | |
|  | | |
| Теоретической и прикладной математики | | |
|  | | |
| Лабораторная работа № 2 | | |
| по дисциплине «Низкоуровневое программирование» | | |
|  | | |
| **ЦЕПОЧЕЧНЫЕ КОМАНДЫ** | | |
|  | | |
|  | Факультет: | ПМИ |
| Группа: | ПМИ-02 |
| Вариант: | 7 |
| Студент: | Сидоров Даниил, |
|  | Дюков Богдан |
| Преподаватель: | Еланцева И. Л., |
|  | Сивак М. А.  . |
|
|  |  |
|  | | |
| Новосибирск | | |
| 2022 | | |

1. **Цель работы**

Изучить и приобрести практические навыки работы с цепочечными командами (командами обработки строк символов).

1. **Содержание работы**

* Изучить цепочечные команды;
* Написать программу, которая по исходной строке формирует инвертированную строку и проверяет, совпадают ли обе строки;
* Отладить программу, убедиться в правильности ее работы на тестовых примерах.

1. **Текст программы**

.386 *;* это ассемблерная директива, предписывающая Ассемблеру использовать набор операций для процессора 80386

.model flat, stdcall ; плоская модель памяти,использующаяся в операционной системе Windows.

; stdcall — используемое соглашение о вызовах процедур

; прототипы внешних процедур описываются директивой extrn,

; после знака @ указывается общая длина передаваемых параметров,

; после двоеточия указывается дистанция «ближняя» – near

extrn GetStdHandle@4:near ; функция получения дескриптора (функция возвращает дескриптор в eax)

extrn WriteConsoleA@20:near

extrn CharToOemA@8:near ; функция перекодировки строковых констант в DOS

extrn ReadConsoleA@20:near

extrn ExitProcess@4:near ; функция выхода из программы

extrn lstrlenA@4:near ; функция определения длины строки

; сегмент данных

.data

dout dd ? ; дескриптор вывода

din dd ? ; дескриптор ввода

; управляющие символы: 13 – возврат каретки, 10 – переход на новую строку, 0 – конец строки

strn1 db "Введите cтроку: ",13,10,0 ; первая выводимая строка,

strn2 db "Инвертированная строка: ",13,10,0 ; вторая выводимая строка,

strn3 db 10, "Не одинаковы",13,10,0 ; третья выводимая строка,

strn4 db 10, "Одинаковы",13,10,0 ; четвертая выводимая строка,

buf dw 400 dup (?); буфер для вводимой строки

invbuf db 400 dup (?); буфер для инвертированной строки

lens dd ? ; количество выведенных и введенных символов

; сегмент кода

.code

start: ; метка точки входа

; перекодируем строку strn1

push offset strn1 ; параметры функции помещаются в стек командой

; offset – операция, возвращающая смещение

push offset strn1

call CharToOemA@8 ; вызов функции

; перекодируем строку strn2

push offset strn2 ; параметры функции помещаются в стек командой

; offset – операция, возвращающая смещение

push offset strn2

call CharToOemA@8 ; вызов функции

; перекодируем строку strn3

push offset strn3 ; параметры функции помещаются в стек командой

; offset – операция, возвращающая смещение

push offset strn3

call CharToOemA@8 ; вызов функции

; перекодируем строку strn4

push offset strn4 ; параметры функции помещаются в стек командой

; offset – операция, возвращающая смещение

push offset strn4

call CharToOemA@8 ; вызов функции

; получим дескриптор ввода

push -10

call GetStdHandle@4

mov din, eax ; переместим результат из регистра eax в ячейку памяти с именем din

; получим дескриптор вывода

push -11

call GetStdHandle@4

mov dout, eax

; вывод строки "Введите cтроку: "

try:push offset strn1 ; в стек помещаем указатель на строку

call lstrlenA@4 ; длина в eax

push 0 ; в стек помещаем 5-й параметр

push offset lens ; 4-й параметр

push eax ; 3-й параметр

push offset strn1 ; 2-й параметр

push dout ; 1-й параметр

call WriteConsoleA@20

; ввод строки, которую нужно инвертировать

push 0 ; в стек помещаем 5-й параметр

push offset lens ; 4-й параметр

push 400 ; 3-й параметр

push offset buf ; 2-й параметр

push din ; 1-й параметр

call ReadConsoleA@20

; проверяем, чтобы не вводилась пустая строка

.if lens == 2

jmp try

.endif

; инвертирование строки

std

sub lens, 2

mov ecx, lens

lea edi, invbuf

lea esi, buf

add esi, lens

dec esi

convert:

movsb

add edi,2

loop convert

cld

; вывод строки "Инвертированная строка: "

push offset strn2 ; в стек помещаем указатель на строку

call lstrlenA@4 ; длина в eax

push 0 ; в стек помещаем 5-й параметр

push offset lens ; 4-й параметр

push eax ; 3-й параметр

push offset strn2 ; 2-й параметр

push dout ; 1-й параметр

call WriteConsoleA@20

; вывод инвертированной строки

push offset invbuf ; в стек помещаем указатель на строку

call lstrlenA@4

push 0 ; в стек помещаем 5-й параметр

push offset lens ; 4-й параметр

push eax ; 3-й параметр

push offset invbuf ; 2-й параметр

push dout ; 1-й параметр

call WriteConsoleA@20

; сравнение строк

lea edi, invbuf

lea esi, buf

mov ecx, lens

repe cmpsb

; если строки совпадат, переходим на флаг equal

je equal

; если строки не совпадают, переходим на флаг notEdual

jne notEqual

; вывод строки "Одинаковы"

equal:

push offset strn4 ; в стек помещаем указатель на строку

call lstrlenA@4

push 0 ; в стек помещаем 5-й параметр

push offset lens ; 4-й параметр

push eax ; 3-й параметр

push offset strn4 ; 2-й параметр

push dout ; 1-й параметр

call WriteConsoleA@20

jmp exit

; вывод строки "Не одинаковы"

notEqual:

push offset strn3 ; в стек помещаем указатель на строку

call lstrlenA@4

push 0 ; в стек помещаем 5-й параметр

push offset lens ; 4-й параметр

push eax ; 3-й параметр

push offset strn3 ; 2-й параметр

push dout ; 1-й параметр

call WriteConsoleA@20

; выход из программы

exit:

push 0 ; параметр: код выхода

call ExitProcess@4

end start

1. **Тесты**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Введенная**  **строка** | **Результат** | | **Комментарий** |
| **Инвертированная**  **строка** | **Ответ** |
| 1 | топот | топот | Одинаковы | Проверка симметричной строки |
| 2 | а б в г д | д г в б а | Не одинаковы | Проверка не симметричной строки |
| 3 | (пробел или несколько пробелов) | (пробел или несколько пробелов) | Одинаковы | Строка пробелов совпадает со строкой пробелов |
| 4 | (пустая строка) | Ввод данных будет продолжаться пока не будут введены корректные данные | | При пустой строке просим повторный ввод, потому что инвертирование и сравнение пустых строк некорректно |
| 5 | Топот | топоТ | Не одинаковы | Проверка симметричного слова, но с различными регистрами на концах |

1. **Вывод**

Контрольные вопросы проработаны.