|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования  Российской Федерации | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования | | |
| «Новосибирский государственный технический университет» | | |
|  | | |
| Теоретической и прикладной математики | | |
|  | | |
| Лабораторная работа № 3 | | |
| по дисциплине «Низкоуровневое программирование» | | |
|  | | |
| **ИНТЕРФЕЙС С ЯЗЫКОМ С++** | | |
|  | | |
|  | 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 использования процедур на языке Ассемблера и их интерфейса с языком С++.

1. **Содержание работы**

* Изучить способы вызова процедуры на языке Ассемблера, возврата из процедуры.
* Изучить условия взаимодействия функции на языке С++ с процедурой на языке Ассемблера. Изучить правила передачи управления в процедуру и обратно. Изучить способы обмена данными между вызывающей функцией на языке С++ и процедурой на языке Ассемблера.
* Написать процедуру на языке Ассемблера, реализующую функцию, которая по исходной строке формирует инвертированную строку и проверяет, совпадают ли обе строки заданного варианта. Написать вызывающую функцию на языке С++, осуществляющую ввод исходных данных и вывод результатов.
* Отладить программу, убедиться в правильности ее работы на тестовых примерах.

1. **Описание использованных функций**

* lstrlenA –функция для получения длины строки;
* InvString - функция инвертирования строки;
* Comparison – функция сравнения исходной и инвертированной строки

1. **Текст програ****ммы**

Файл StringСomparison.asm:

.386

.model flat

extern \_lstrlenA@4: proc ; функция определения длины строки

.data ; сегмент данных

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

strn1 db 10, "The strings match!",13,10,0 ; первая выводимая строка,

strn2 db 10, "The strings don't match!",13,10,0 ; вторая выводимая строка,

strn3 db 10, "Incorrect data entered!",13,10,0 ; третья выводимая строка,

lens dd ? ; количество выведенных и введенных символов

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

.code ; сегмент кода

; начало процедуры инвертирования строки

@InvString@4 proc

mov esi, ecx ; помещаем регистр esi на начало переданной строки

; получаем длину строки и записываем в lens

push ecx

CALL \_lstrlenA@4

mov lens, eax

; проверяем на ввод пустой строки или слишком длинной

.if lens == 0 || lens > 400

jmp error

.endif

; инвертирование строки

std

mov ecx, lens

lea edi, invbuf

add esi, lens

dec esi

convert:

movsb

add edi,2

loop convert

cld

; запись инвертированной строки в регистр eax для возврата из процедуры

lea edi, invbuf

mov eax, edi

jmp exit

; в случае ошибочного ввода помещаем в регистр eax строку с сообщением об ошибке

error:

lea edi, strn3

mov eax, edi

exit:

ret

; Конец процедуры инвертирования строки

@InvString@4 endp

; начало процедуры сравнения строк

@Comparison@8 proc

mov edi, edx

mov esi, ecx

; получаем длину строки и записываем в lens

push ecx

CALL \_lstrlenA@4

mov lens, eax

; проверяем на ввод пустой строки или слишком длинной

.if lens == 0 || lens > 400

jmp error

.endif

mov ecx, lens

repe cmpsb

; если строки совпадат, переходим на флаг equal

je equal

; если строки не совпадают, переходим на флаг notEdual

jne notEqual

; если строки равны, помещаем в регистр eax "The strings match"

equal:

lea edi,strn1

mov eax,edi

jmp exit

; если строки не равны, помещаем в регистр eax "The strings don't match!"

notEqual:

lea edi,strn2

mov eax,edi

jmp exit

; в случае ошибочного ввода помещаем в регистр eax строку с сообщением об ошибке

error:

lea edi,strn3

mov eax,edi

; выход из программы

exit:

ret;

; начало процедуры сравнения строк

@Comparison@8 endp

end

Файл Main.cpp:

#include <iostream>

#include <string>

// Функция инвертирования строки

extern "C" char\* \_\_fastcall InvString(char\* string);

// Функция сравнения исходной и инвертированной строки

extern "C" char\* \_\_fastcall Comparison(char\* string, char\* invString);

int main()

{

std::string originalString;

std::cout << "Enter the string: ";

getline(std::cin, originalString);

char\* cstr = new char[originalString.length() + 1];

strcpy(cstr, originalString.c\_str());

char\* invStr = InvString(cstr);

char\* result = Comparison(cstr, invStr);

if(originalString != "" && originalString.length() <= 400)

std::cout << "Inverted string: " << invStr;

std::cout << result;

return 0;

}

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) или (строка длинной более 400 символов) | Будет выведено сообщение о некорректных введенных данных и программа завершится | | Инвертирование и сравнение пустых строк, а также ввод строки, длина которой превышает допустимое значение |
| 5 | Топот | топоТ | Не одинаковы | Проверка симметричного слова, но с различными регистрами на концах |

1. **Вывод**

Контрольные вопросы проработаны.