Министерство образования Республики Беларусь

Учреждение образования

“Брестский государственный технический университет”

**Лабораторная работа №1**

**По дисциплине КСиС за 4 семестр**  
**Тема: «Знакомство с Ассемблером»**

**Выполнил:**

Студент группы ПО-6(1)  
 2-го курса

Мартынович Даниил

**Проверил:**

Бойко Д. О

Брест 2021

Задание:

1. Изучить теоретический материал по архитектуре и программной модели процессора x86, основам организации программ на языке Ассемблер (файл modelproc.pdf, branchmass.pdf, symproc.pdf, другие источники).

2. Написать ассемблерную вставку, реализующую обработку строки

согласно варианту. Оформить ее в виде отдельной функции.

3. Реализовать данную обработку строки также в виде функции на С++.

4. Сравнить быстродействие обоих вариантов. В отчете отразить выводы.

ПРИМЕЧАНИЕ: для разработки рекомендуется использовать MS Visual Studio.

Вариант: 8

Сместить все символы на один вперед циклично.

#include <iostream>

#include <windows.h>

#include <string.h>

#include <tchar.h>

#include <ctime>

#include <string>

#define \_CRT\_SECURE\_NO\_WARNINGS

void MoveASM(char\* str, size\_t cnt)

{

size\_t len = strnlen\_s(str, 32) - 1;

\_\_asm

{

push eax // уменьшает значение регистра стека eax

push ecx // уменьшает значение регистра стека ecx

push ebx // уменьшает значение регистра стека ebx

mov ecx, cnt //cnt помещаем в ecx

lbl1 :

push ecx //уменьшаем значение ecx

xor eax, eax // Логическое исключающее

mov ebx, str //строку в регистр ebx

mov al, byte ptr[ebx]//размер строки в al

push eax //уменьшение значения регистра eax

mov ecx, len //перемещение в ecx переменную len

mov esi, str //новую строку перемещаем в esi

inc esi //переходим на следующий символ

mov edi, str//символ в edi

rep movsb//повторить ->

pop eax //извлекаем значение eax

mov ebx, str //символ в ebx

add ebx, len //складываем ebx с len и храним в ebx

mov byte ptr[ebx], al //перемещаем al на кол-во байтов

pop ecx//извлекаем значение ecx

dec ecx//вычитаем 1 из ecx

jecxz ext //цикл

jmp lbl1//переходим к lbl1

ext :

pop ebx

pop ecx

pop eax

}

}

void MOVEonC(char\* str)

{

int count = strlen(str)-1;

for(int i = 0; i < count; i++)

{

std::swap(str[count], str[i]);

}

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

setlocale(LC\_ALL, "ru");

char\* str = new char[32];

char\* str1 = new char[32];

std::cout << "Input strok:\n";

std::cin >> str;

strcpy(str1, str);

srand(time(0));

std::cout << "\nAssembler code\n";

MoveASM(str, 5);

std::cout << "\nResult:\n" << str;

std::cout << "\nTime ASM: " << clock() / 1000.0;

std::cout << "\n";

srand(time(0));

std::cout << "\nC++ code\n";

MOVEonC(str1);

std::cout << "\nResult:\n"<<str1;

std::cout << "\nTime C++: " << clock() / 1000.0;

std::cout << "\n";

std::cin.get();

system("pause");

return 0;

}

Вывод: на моем ПК время выполнения программы на C++ одинаково с программой на Assembler.