Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №2

за 4 семестр

По дисциплине: «КСиС»

Выполнил:

Студент 2 курса

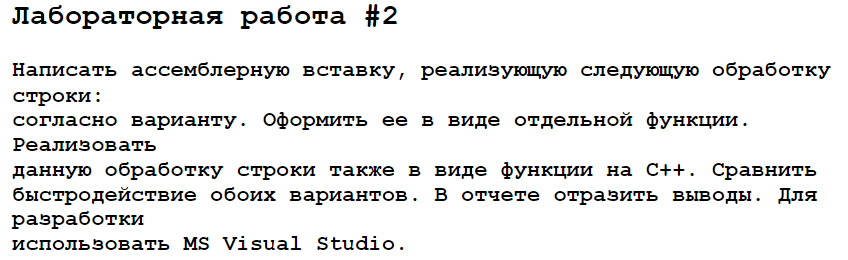
Группы ПО-4(2)

Кречко К.А.

Проверил:

Савицкий Ю. В.

Брест 2021



**Вариант – 8**

****

**Ассемблерная вставка:**

#include <iostream>

#include <ctime> //для проверки быстродействия

#include <string.h>

using namespace std;

int MoveASM(char\* str){

\_asm {

push str //запись в стек строки

call strlen //вызов функции для вычисления длины строки

add sp, 4 //sp-указатель на вершину стека, add-"сложить"

mov ecx, eax //перенос eax(равен кол-ву символов в строке) в ecx(счётчик)

dec ecx //декремент регистра ecx

mov esi, str //присваивание регистру esi строки str(напр.17823912)

mov edi, esi //присваивание регистру edi значения esi

lodsb //копир.1 байта из памяти по адресу DS:Si в регистр AL (esi увеличится на 1 байт)

push eax //запись в стек eax

rep movsb //записать в ячейку по адресу ES:(E)DI байт из ячейки с адресом DS:(E)SI (обнуление ecx, к edi и esi +4 байта)

pop eax // pop - чтение из стека

stosb //сохранение AL по адресу ES:(E)DI

}

return 0;

};

int main()

{

system("color f0");

//создание исходной строки

char str[128];

cout << "Input your string: ";

gets\_s(str);

clock\_t start = clock(); //начало замера быстродействия

MoveASM(str); //вызов функции

clock\_t finish = clock(); //конец замера быстродействия

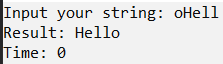
cout << "Result: " << str << endl; //вывод строки

cout << "Time: " << finish - start << endl; //вывод быстродействия

return 0;

};

**Результат:**

****

Т.к. функция выполняется слишком быстро, мы вставим её в цикл и многократно прогоним. Результат поделим на число иттераций.

Прогнав функцию 2’000’000 раз получили:



**Чистый C++:**

#include <iostream>

#include <ctime> //для проверки быстродействия

#include <string>

using namespace std;

int shift(char\* str) { //функция смещения

int i = 1;

char n = str[0]; //присваиваем n значение первого элемента

while (str[i] != '\0') { //пока строка не пуста

str[i - 1] = str[i]; //присваиваем предыдущему элементу значение следующего (сдвигаем влево)

i++;

}

str[i-1] = n; //т.к. смещение циклично последнему элементу присваиваем начальное значение

return \*str;

}

int main()

{

system("color f0");

//создание исходной строки

char str[128];

cout << "Input your string: ";

gets\_s(str);

clock\_t start = clock(); //начало замера быстродействия

shift(str); //вызов функции

clock\_t finish = clock(); //конец замера быстродействия

cout << "Result: " << str << endl; //вывод строки

cout << "Time: " << finish - start << endl; //вывод быстродействия

return 0;

};

**Результат:**

****

Прогнав функцию 2’000’000 раз получили:



**Вывод:** В данной лабораторной работе я ознакомился с аппаратно-программной архитектуройпроцессоров семейства Intel и программированием на языке Ассемблера.