Министерство образования Республики Беларусь

УО «Брестский государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №1**

По дисциплине: “Компьютерные системы и сети”

Тема: “Знакомство с ассемблером”

**Вариант №9**

**Выполнил**: студент 2 курса группы ПО-7 Крупенков Михаил Дмитриевич

**Проверил:** Савицкий Ю.В.

Брест 2022

**Цель работы**

1. Изучить теоретический материал по архитектуре и программной модели процессора x86, основам организации программ на языке Ассемблер.

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.

**Постановка задачи**

9. Перевернуть две половины строки

**Ход работы**

**Main.cpp**

#include <iostream>

#include <ctime>

char text[] = "HelloWorld";

unsigned int half = strlen(text) / 2;

void coup\_halfs\_asm() {

\_\_asm {

mov ecx, half

mov edi, half

mov esi, 0

forbegin:

mov al, byte ptr[text + esi]

mov dl, byte ptr[text + edi]

mov byte ptr[text + edi], al

mov byte ptr[text + esi], dl

inc esi

inc edi

loop forbegin

}

}

void coup\_halfs\_cxx() {

char buffer;

int j = half;

for (int i = 0; i < half; i++, j++) {

buffer = text[i];

text[i] = text[j];

text[j] = buffer;

}

}

int main() {

std::cout << "Input pure: " << text << "\n";

coup\_halfs\_asm();

std::cout << "Output asm: " << text << "\n";

coup\_halfs\_cxx();

std::cout << "Output c++: " << text << "\n";

std::cout << "\nTime testing\n";

unsigned int start\_time = clock();

for (int i = 0; i < 1000000; i++) {

coup\_halfs\_asm();

}

std::cout << "asm 1kk loops: " << clock() - start\_time << " ms\n";

start\_time = clock();

for (int i = 0; i < 1000000; i++) {

coup\_halfs\_cxx();

}

std::cout << "c++ 1kk loops: " << clock() - start\_time << " ms\n";

return 0;

}

**Результаты программы**

Input pure: HelloWorld

Output asm: WorldHello

Output c++: HelloWorld

Time testing

asm 1kk loops: 57 ms

c++ 1kk loops: 47 ms

**Подведение итогов:** на небольшой программе быстрее работает С++

**Вывод:** был получен опыт в работе с ассемблерными вставками.