**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Кубанский Государственный Университет**

Кафедра информационных технологий

ОТЧЕТ

о выполнении лабораторной работы №1

по дисциплине «Системы реального времени»

Выполнил:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ст. гр. 44 Ерюшев И.Д.\_\_\_\_

Проверил:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

доц. каф. ИТ Полетайкин А.Н.

Краснодар

2024

**ЛАБОРАТОРНАЯ РАБОТА 1**

**ПРЕДСТАВЛЕНИЕ ЦЕЛЫХ ЧИСЕЛ В ПАМЯТИ ЭВМ**

Цель: изучение принципов представления числовой информации в памяти ЭВМ; приобретение практических навыков представления целых чисел со знаком в машинном формате.

Задание:  
 1. Представить целые числа 151 и -7814 в формате DB, DW, DD.

2. Составить и откомпилировать программу, определив число 151 в форматах DB, DW, DD, а число -7814 в форматах DW, DD.

3. Задать такие операции пересылки данных:

– загрузить регистр BX числом -7814 из сегмента данных;

– с использованием заданного варианта косвенной адресации([BP][DI]) записать содержимое BX в сегмент данных со смещением на I1 байт относительно метки I2 (число 151 предварительно загрузить в соответствующий базовый или индексный регистр, при наличии других операндов в формуле адресации определить их произвольно).

4. Проверить результаты расчетов и пересылок в дампе памяти.

5. Сделать расчет времени выполнения программы (методика расчета представлена в прил. А).

**Задание 1.**

Для представления целых чисел 151 и -7814 в формате DB, DW, DD нужно выполнить алгоритм, состоящий из 4 для положительных и 5 операций для отрицательных.

|  |  |  |
| --- | --- | --- |
| Операции | Формат DB | Формат DW |
| 1) модуль в двоичной форме | 1001 0111 | 1001 0111 |
| 2) дописываем 0 | 1001 0111 | 0000 0000 1001 0111 |
| 3) в 16-ричном виде | 69 | FF 69 |
| 4) в дампе памяти | 69 | 69 FF |
| Формат DD | | |
| 1) модуль в двоичной форме | 1000110 | |
| 2) дописываем 0 | 0000 0000 0000 0000 0000 0000 1001 0111 | |
| 3) в 16-ричном виде | FF FF FF 69 | |
| 4) в дампе памяти | 69 FF FF FF | |

Рисунок 1 – перевод числа 151 в 16-ый код.

|  |  |
| --- | --- |
| Операции | Формат DW |
| 1) модуль в двоичной форме | 1111010000110 |
| 2) дописываем 0 | 0001 1110 1000 0110 |
| 3) инвертируем и прибавляем 1 | 1110 0001 0111 1010 |
| 3) в 16-ричном виде | E1 7A |
| 4) в дампе памяти | 7A E1 |
| Формат DD | |
| 1) модуль в двоичной форме | 1111010000110 |
| 2) дописываем 0 | 0000 0000 0000 0000 0001 1110 1000 0110 |
| 3) инвертируем и прибавляем 1 | 1111 1111 1111 1111 1110 0001 0111 1010 |
| 3) в 16-ричном виде | FF FF E1 7A |
| 4) в дампе памяти | 7A E1 FF FF |

Рисунок 2 – од числа -7814 в 16-й код.

**Задание 2.**

Используя полученные знания из курса по параллельному и низкоуровневому программированию, составим и откомпилируем программу с архитектурой ARM NASM x64.

Для компиляции программы был использован терминал iTerm.

nasm -g -felf64 asm.s

После этой команды будет создан файл asm.o.

Код программы:

section .data

i1db db 151

i1dw dw 151

i1dd dd 151

i2dw dw -7814

i2dd dd -7814

section .text

global \_start

\_start:

debug

int 0x80

**Задание 3.**

Для загрузки в базовый регистр BX, который используется для указания базового (начального) адреса объекта данных в памяти, была использована инструкция MOV, которая копирует данные.

Код программы:

section .data

i1db db 151

i1dw dw 151

i1dd dd 151

i2dw dw -7814

i2dd dd -7814

section .text

global \_start

\_start:

mov rbp, rsp ;debug

mov ecx, [i2dd]

mov ebx, i2dd

mov edi, [i1dd]

mov [ebp+edi], ecx

mov eax, 1

int 0x80

**Задание 4.**

Чтобы проверить результаты расчетов и пересылок в дампе памяти воспользуемся инструментами отладки OllyDbg. Запустив программу в отладчике, находим адрес и значение переменных в памяти в различных форматах:

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 1 – дамп памяти значений переменных

Также находим результат пересылок, пролистав вниз. Получим, что смещение числа I2 произошло ровно на I1=151 байт:

Изображение выглядит как текст, Шрифт, снимок экрана, число

Автоматически созданное описание

Рисунок 2 – дамп памяти пересылок

**Задание 5.**

Посчитаем количество тактов процессора на выполнение каждой команды:

Start:

mov EAX, [I2\_DD] ; 12 + 6

mov ESI, I1\_DD ; 12

mov EBX, offset I2\_DD ; 12 + 6

mov [EBX + ESI], EAX ; 13 + 7

end Start

Выполнение программы требует 68 тактов процессора. При тактовой частоте процессора 2.1 ГГц, без учёта прочих факторов, время работы программы должно составлять 32 наносекунды.

**РЕЗУЛЬТАТЫ**

Для задания 1 было сделано 2 рисунка с переводом целых положительных и отрицательных чисел в формат DB, DW, DD. Для этого использовался алгоритм, предложенный в учебном пособии.

В задании 2 были изучены типы данных и базовая структура программы для компиляции.

В задании 3 был написан код внутри функции main для работы с сегментами данных и косвенной речью.

Для задания 4 использовался RadAsm dump до и после компиляции.

В задании 5 нужно было сделать расчет времени выполнения программы, для этого использовалось учебное пособие, а именно приложение А и его таблицы.

**ВЫВОД**

В итоге, была написана программа для выполнения задания из лабораторной работы №1. Для написания программы были изучены РОН, флаги, принцип работы памяти и способ представления информации в ней.