



«Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ _____
КАФЕДРА _____ КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ _____

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

О Т Ч Е Т

по лабораторной работе № 2

Дисциплина: Машинно-зависимые языки и основы компиляции

Название лабораторной работы: Программирование целочисленных вычислений

Вариант: 17

Студент гр. ИУ6-43Б

04.03.2022

(Подпись, дата)

М.А. Мяделец

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

М.В. Широкова

(И.О. Фамилия)

Москва, 2022

Программирование целочисленных вычислений

Цель работы: изучение форматов машинных команд, команд целочисленной арифметики ассемблера и программирование целочисленных вычислений.

Задание 1.

Рассмотрим задание для лабораторной работы 2 (смотри рисунок 1).

Лабораторная работ №2. Программирование целочисленных вычислений.

Вычислить целочисленное выражение:

$$c = \frac{a}{3} - k + (d + 2) * 5$$

Рисунок 1 - Условие задания.

Поместим в оперативную память двухбайтовые значения:

- 1) $a = 22$
- 2) $k = 5$
- 3) $d = -6$

Рассмотрим некоторые операции и их результаты (смотри рисунок 2).

$$\begin{aligned} c &= \frac{a}{3} - k + (d + 2) \cdot 5 \\ c &= \frac{22}{3} - 5 + (-6 + 2) \cdot 5 = -18 = ff\bar{f}.ee \\ \textcircled{1} \quad \frac{22}{3} &= 7 \text{ (остаток: 1)} \Rightarrow \begin{matrix} AX = 7 \\ DX = 1 \end{matrix} \\ \textcircled{2} \quad -6 + 2 &= -4_{10} = \underbrace{111\dots 1}_8 \cdot \underbrace{11111100}_8 = ff\bar{f}.c \\ \textcircled{3} \quad -4 \cdot 5 &= -20 \Rightarrow \begin{matrix} AX = \underbrace{111\dots 1}_8 \underbrace{11101100}_8 = ff.ec \\ DX = \underbrace{111\dots 1}_8 \underbrace{111\dots 1}_8 = ff.ff \end{matrix} \end{aligned}$$

Рисунок 2 - Описание работы операций деления и умножения.

Реализуем данную задачу на языке ассемблера (смотри рисунки 3, 4).

```

1      section .data    ; сегмент инициализированных переменных
2
3      a      dw  22
4      k      dw  5
5 ▼    d      dw  -6
6
7      section .bss     ; сегмент неинициализированных переменных
8
9 ▼    c      resw 1
10
11     section .text    ; сегмент кода
12     global _start
13 ▼   _start:
14     mov AX, [a]
15
16
17     mov CX, 3
18     cwd
19     div CX           ; a / 3
20
21     sub AX, [k]      ; (a / 3) - k
22     mov BX, AX
23
24     add word[d], 2   ; d + 2
25     mov AX, [d]
26
27     mov CX, 5
28     imul CX          ; (d + 2) * 5
29
30     add AX, BX       ; (a / 3) - k + (d + 2) * 5
31
32     mov [c], AX
33

```

Рисунок 3 - Программа по вычислению выражения.

0804:8080 Lab2_1!_start	66 a1 b8 90 04 08	mov ax, [0x80490b8]
0804:8086	66 b9 03 00	mov cx, 3
0804:808a	66 99	cwd
0804:808c	66 f7 f1	div cx
0804:808f	66 2b 05 ba 90 04 08	sub ax, [0x80490ba]
0804:8096	66 89 c3	mov bx, ax
0804:8099	66 83 05 bc 90 04 08...	add word [0x80490bc], 2
0804:80a1	66 a1 bc 90 04 08	mov ax, [0x80490bc]
0804:80a7	66 b9 05 00	mov cx, 5
0804:80ab	66 f7 e9	imul cx
0804:80ae	66 01 d8	add ax, bx
0804:80b1	66 a3 c0 90 04 08	mov [0x80490c0], ax

Рисунок 4 - Машинный и дисассемблированный коды программы.

Рассмотрим подробнее некоторые операции в отладчике (смотри нижеприведенное объяснение и рисунки 5-8).

1) Операция деления (19 строчка на рисунке 4)

Registers	
EAX	00000007
ECX	00000003
EDX	00000001

Рисунок 5 - Значения регистров после операции div.

Перед операцией необходимо расширить регистр AX до DX:AX командой CWD, чтобы деление выполнилось корректно.

$$\frac{DX:AX}{AX},$$

причем после выполнения операции в регистр DX помещается остаток.

2) Операция сложения (24 строчка на рисунке 3)

$d + 2 = -6 + 2 = -4 = 1111\ 1111\ 1111\ 1100$ (в дополнительном коде) = **ff fc** (в 16-ричной).

0804:90b0	d8	66	a3	c0	90	04	08	00	16	00	05	00	fc	ff
-----------	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Рисунок 6 - Значения по адресу d.

3) Операция умножения (28 строчка на рисунке 3)

Registers	
EAX	0000ffec
ECX	00000005
EDX	0000ffff

Рисунок 7 - Значения регистров после операции imul.

В регистр EDX помещается старшая часть результата после умножения. В нашем случае $-4 * 5 = -20$, соответственно регистр DX будет содержать знаковый разряд.

4) Результат (32 строчка на рисунке 3)

Значение в памяти, которую символизирует имя C (итоговое значение совпало с ответом на рисунке 2)



Рисунок 8 - Значения по адресу C.

Задание 2. Расшифровка команд *mov*

Перевод строчки 1 (смотри рисунок 4).

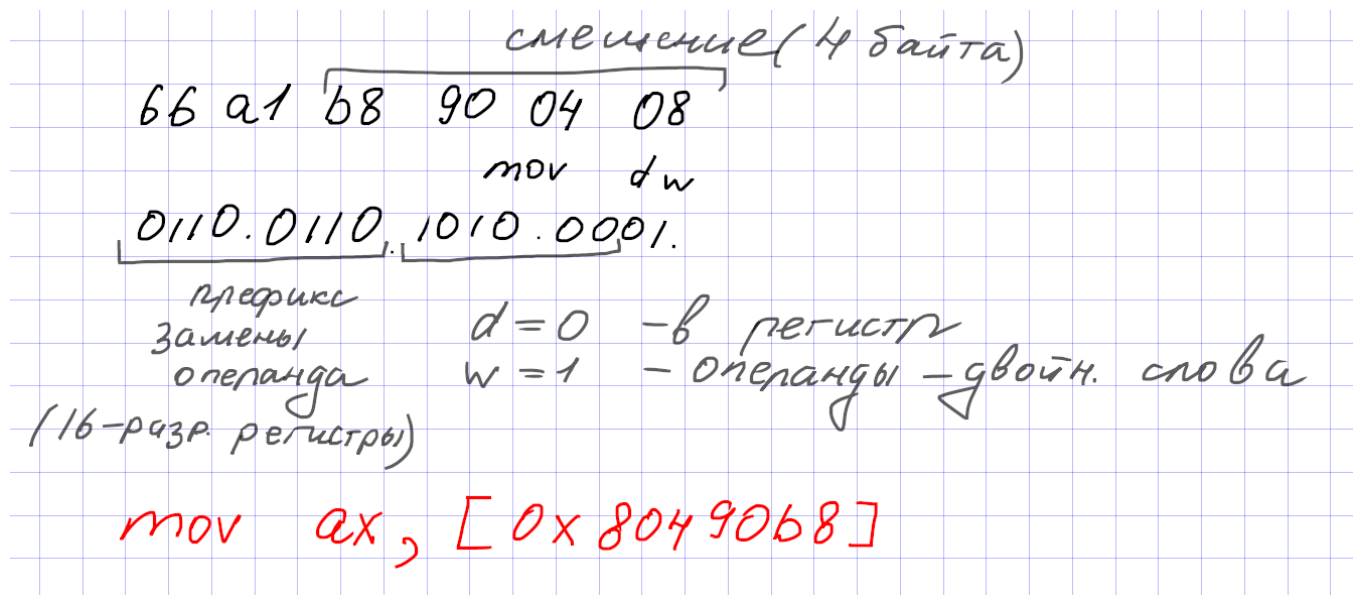


Рисунок 9 - Перевод машинного кода 1

Перевод строчки 2 (смотри рисунок 4).

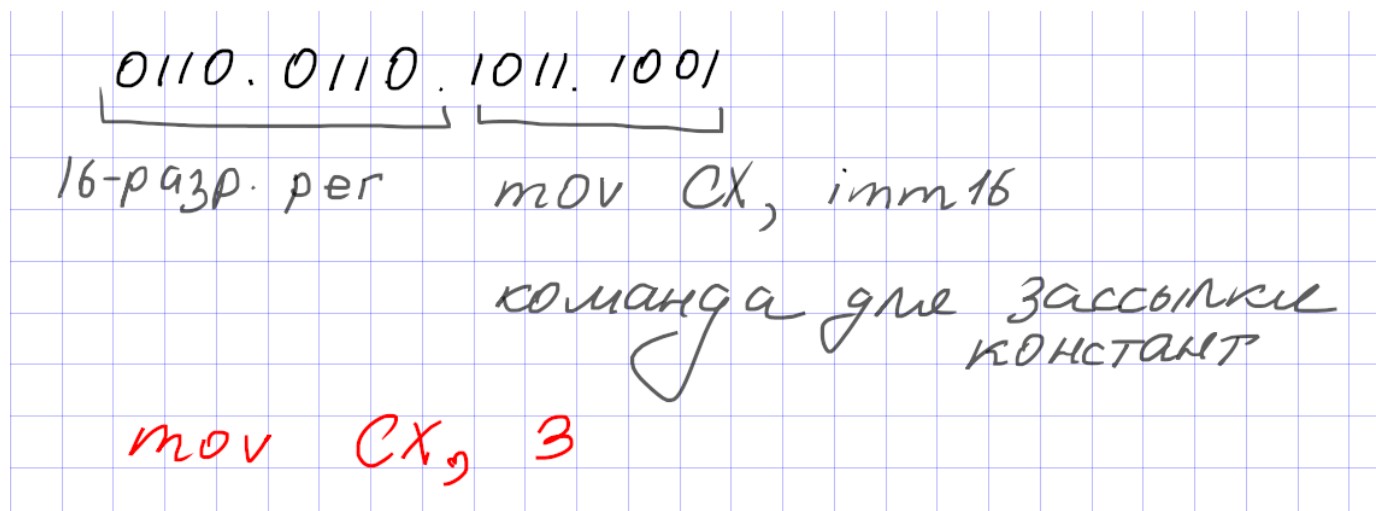


Рисунок 10 - Перевод машинного кода 2

Перевод строчки 3 (смотри рисунок 10).

66 89 C3 *mov dw mod reg1 reg2*
 0110. 0110. 1000. 1001. 1100. 0011
 └──────────┘ └──────────┘ └──────────┘
 предфикс разм d=0 - из регистра
 операнда w=1 - операнды - дв. слова
 (16-разр рег) mod=11 - операнды - регистры
 reg1 = AX , reg2 = BX

mov BX, AX

Рисунок 11 - Перевод машинного кода 3

Контрольные вопросы

- 1) Что такое машинная команда? Какие форматы имеют машинные команды процессора IA32? Чем различаются эти форматы?

Машинная команда – команда процессоре (смотри рисунок 12).

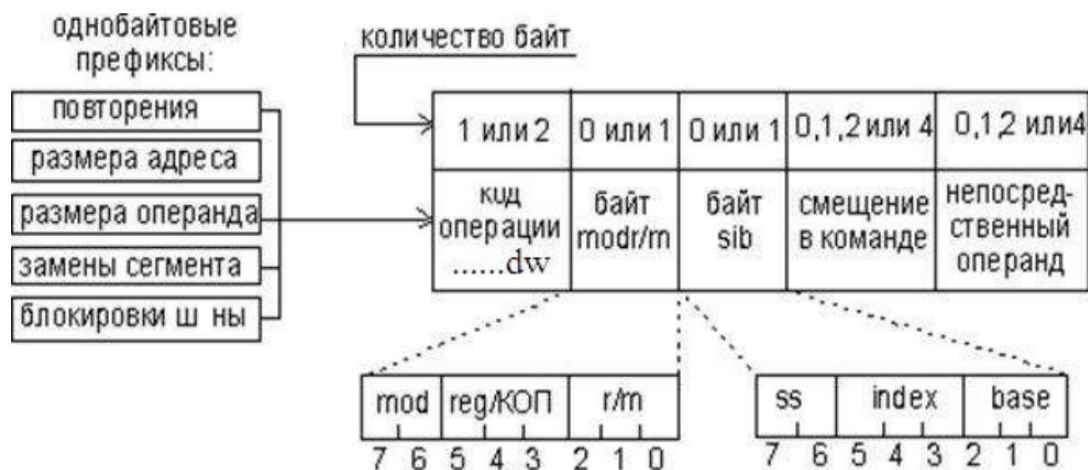


Рисунок 12 - Общий формат команды процессора старших моделей

- 2) Назовите мнемоники основных команд целочисленной арифметики. Какие форматы для них можно использовать?

Add, sub, mul / imul, div / idiv, dec, inc

Форматы различаются по типам операндов: регистры, значения в оперативной памяти или литералы, а также по количеству операндов. Например, команда add принимает 2 операнда, mul и inc – один.

- 3) *Сформулируйте основные правила построения линейной программы вычисления заданного выражения.*

Определяем инициализированные и неинициализированные данные в оперативной памяти, далее расставляем порядок действий в выражении, в соответствие с которым последовательно выполняем команды целочисленной арифметики для получения ответа.

- 4) *Почему ввод-вывод на языке ассемблера не программируют с использованием соответствующих машинных команд? Какая библиотека используется для организации ввода вывода в данной лабораторной?*

Системные функции read и write осуществляют ввод и вывод данных в виде символьных строк. Для перевода числа в строку и обратно в библиотеке для выполнения лабораторных работ *lib.asm* предусмотрены подпрограммы *IntToStr* и *StrToInt* соответственно.

5. *Расскажите, какие операции используют при организации ввода-вывода.*

Для операции ввода-вывода используются следующие операции, которые подробно описаны в комментариях (смотри рисунок 13).

```
; write
mov     eax, 4           ; системная функция 4 (write)
mov     ebx, 1           ; дескриптор файла stdout=1
mov     ecx, ExitMsg     ; адрес выводимой строки
mov     edx, lenExit     ; длина выводимой строки
int     80h              ; вызов системной функции
; read
mov     eax, 3           ; системная функция 3 (read)
mov     ebx, 0           ; дескриптор файла stdin=0
mov     ecx, InBuf       ; адрес буфера ввода
mov     edx, lenIn       ; размер буфера
int     80h              ; вызов системной функции
; exit
mov     eax, 1           ; системная функция 1 (exit)
xor     ebx, ebx         ; код возврата 0
int     80h              ; вызов системной функции
```

Рисунок 13 – Шаблон для 32-разрядной системы

Вывод: была сделана лабораторная работа 2, которая была связана с программированием целочисленных вычислений. Было посчитано выражение, результат которого был помещен в оперативную память. В процессе выполнения задания были использованы команды целочисленной арифметики, применены знаковые расширения, проанализированы различия знаковых и беззнаковых команд `div / idiv`, `mul / imul`, а также произведен перевод машинных кодов на язык ассемблера. Результаты тестирования корректны, работы выполнена успешно.