|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **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 (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

**ОТЧЕТ**

|  |  |
| --- | --- |
| **По лабораторной работе №2** |  |

**Название :** Программирование целочисленных вычислений

**Дисциплина:** Машинно-зависимые языки и основы компиляции

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ-42б |  |  | С.В. Астахов |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  |  |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

1 вариант

Москва, 2021

# Задание

f = a\*b – b3/(k2+2)

1. Разработать программу, вычисляющую заданное выражение. Просмотреть в отладчике и зафиксировать в отчете ход выполнения вычислений (покомандно). Убедиться в правильности программы.

2. Посмотреть в отладчике форматы 3-4 команд mov и расшифровать двоичные коды этих команд, используя материалы теоретической части.

Часть 1

Код программы:

; Template for console application

.586

.MODEL flat, stdcall

OPTION CASEMAP:NONE

Include kernel32.inc

Include masm32.inc

IncludeLib kernel32.lib

IncludeLib masm32.lib

.CONST

MsgExit DB "Press Enter to Exit",0AH,0DH,0

MsgInp DB "Enter number",0AH,0DH,0

MsgLn DB 0AH,0DH,0

reqA DB 'Input A: ',13,10,0 ; çàïðîñ

reqB DB 'Input B: ',13,10,0

reqK DB 'Input K: ',13,10,0

MsgResult DB 'Result of a\*b - b^3/(k^2+2): ',13,10,0

.DATA

buffer DB 10 dup ('0') ; áóôåð ââîäà

.DATA?

inbuf DB 100 DUP (?)

outstr DB 10 DUP (?)

A SWORD ?

B SWORD ?

K SWORD ?

dbgOut SWORD ?

ab SWORD ?

K2add2 SWORD ?

B3 SWORD ?

fraction SWORD ?

result SWORD ?

.CODE

Start:

XOR EAX,EAX

Invoke StdOut,ADDR reqA

Invoke StdIn,ADDR buffer,LengthOf buffer

Invoke StripLF,ADDR buffer

Invoke atol,ADDR buffer ;result in EAX

mov dword ptr A, EAX

Invoke StdOut,ADDR reqB

Invoke StdIn,ADDR buffer,LengthOf buffer

Invoke StripLF,ADDR buffer

Invoke atol,ADDR buffer ;result in EAX

mov dword ptr B, EAX

Invoke StdOut,ADDR reqK

Invoke StdIn,ADDR buffer,LengthOf buffer

Invoke StripLF,ADDR buffer

Invoke atol,ADDR buffer ;result in EAX

mov dword ptr K, EAX

mov AX, A

imul B ;DX:AX:=AX\*B=A\*B

mov ab, AX

mov AX, K

imul K ;DX:AX:=AX\*K=K\*K

add AX, 2

mov K2add2, AX

mov AX, B

imul B ;DX:AX:=AX\*B=B

imul B ;DX:AX:=AX\*B=B\*B

mov B3, AX

mov AX, B3

cwd ;DX:AX = AX

idiv K2add2 ;AX:=(DX:AX):K2add2

mov fraction, AX

mov AX, ab

sub AX, fraction

mov result, AX

Invoke StdOut,ADDR MsgResult

Invoke dwtoa,result,ADDR outstr

Invoke StdOut,ADDR outstr

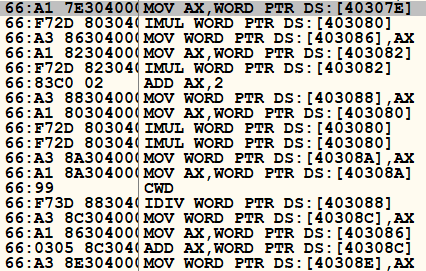
Invoke StdOut,ADDR MsgLn

Invoke StdOut,ADDR MsgExit

Invoke StdIn,ADDR inbuf,LengthOf inbuf

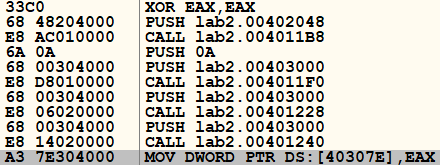
Invoke ExitProcess,0

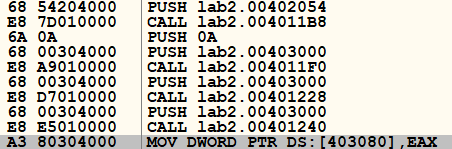
End Start

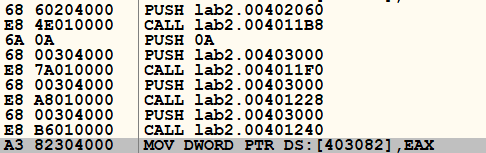
Рисунок 1 — код основной части программы в окне отладчика(операции ввода-вывода опущены)

Ход выполнения программы

Ниже приведены фрагменты кода отвечающие за ввод-вывод и преобразование значений через процедуры StdOut, StdIn, atol

Рисунок 2 — ввод и преобразование к числу значения переменной А

Рисунок 3 — ввод и преобразование к числу значения переменной B

Рисунок 4 — ввод и преобразование к числу значения переменной K

Далее, для удобства пояснения, рассмотрим ход выполнения основной части программы в виде таблицы.

Таблица 1 — ход выполнения основной программы.

|  |  |  |
| --- | --- | --- |
| Команда в исходном коде | Команда в отладчике | Пояснения |
| mov AX, A | MOV AX,WORD PTR DS:[40307E] | Запись значения переменной А в регистр АХ |
| imul B | IMUL WORD PTR DS:[403080] | Умножение значения в регистре АХ на В  DX:AX = AX \* B = A \* B |
| mov ab, AX | MOV WORD PTR DS:[403086],AX | Сохранение значения в регистре AX в переменную ab  ab = AX = A \* B (если A \* B умещается в разрядную сетку AX) |
| mov AX, K | MOV AX,WORD PTR DS:[403082] | Запись значения переменной K в регистр АХ |
| imul K | IMUL WORD PTR DS:[403082] | Умножение значения в регистре AX на K  DX:AX = AX \* K = K \* K = K2 |
| add AX, 2 | ADD AX,2 | Сложение значения в регистре AX с 2  AX = AX + 2 = K2 + 2(если K2 умещается в разрядную сетку AX) |
| mov K2add2, AX | MOV WORD PTR DS:[403088],AX | Запись значения регистра AX в переменную K2add2  K2add2 = AX = K2 + 2 |
| mov AX, B | MOV AX,WORD PTR DS:[403080] | Запись значения переменной B в регистр АХ |
| imul B | IMUL WORD PTR DS:[403080] | Умножение значения в регистре AX на B  DX:AX = AX \* B = B \* B = B2 |
| imul B | IMUL WORD PTR DS:[403080] | Умножение значения в регистре AX на B  DX:AX = AX \* B = B2 \* B = B3 (если B2 умещается в разрядную сетку AX) |
| mov B3, AX | MOV WORD PTR DS:[40308A],AX | Запись значения регистра AX в переменную B3  B3 = AX = B3 |
| mov AX, B3 | MOV AX,WORD PTR DS:[40308A] | Избыточная запись значения B3 в регистр АХ(сделана для читаемости кода) |
| cwd | CWD | Расширения двубайтного числа в регистре АХ до четырехбайтного в DX:AX |
| idiv K2add2 | IDIV WORD PTR DS:[403088] | Целочисленное деление значения в DX:AX на значение переменной K2add2 c записью в АХ  AX = (DX:AX) / K2add2 = B3 / (K2 + 2) |
| mov fraction, AX | MOV WORD PTR DS:[40308C],AX | Запись результата деления в переменную fraction  fraction = AX = (DX:AX) / K2add2 = B3 / (K2 + 2) |
| mov AX, ab | MOV AX,WORD PTR DS:[403086] | Запись значения переменной ab в регистр АХ  AX = ab = A \* B |
| sub AX, fraction | ADD AX,WORD PTR DS:[40308C] | Вычетание значения регстра АХ и результата деления  AX = AX - fraction = A \* B - B3 / (K2 + 2) |
| mov result, AX | MOV WORD PTR DS:[40308E],AX | Запись значения регистра АХ в переменную result |

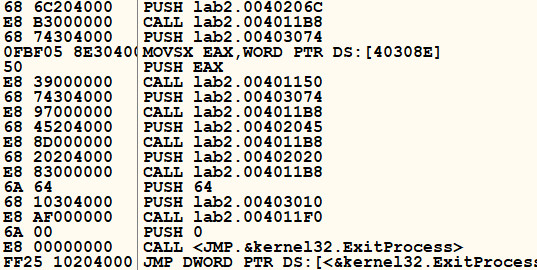
Рисунок 5 — вывод результата и завершение работы программы

Таблица 2 — тесты

|  |  |  |
| --- | --- | --- |
| Входные данные | Ожидаемый результат | Результат |
| 0 0 0 | 0 | 0 |
| 2 6 2 | -24 | -24 |
| 2 6 10 | 10 | 10 |
| 3 2 10 | 6 | 6 |
| 5 -10 11 | -42 | -42 |

Часть 2

Команда: MOV AX,BX

Код: 66:8B C3

Двоичный код: 01100110 10001011 11000011

Команда: MOV AX,5

Код: 66:B8 0500

Двоичный код: 01100110 10111000 00000101

Команда: MOV DWORD PTR DS:[40307E],EAX

Код: A3 7E 30 40 00

Двоичный код: 10100011 01111110 00110000 01000000 00000000

Контрольные вопросы

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

Машинная команда представляет собой код, определяющий элементарную операцию в ЭВМ и ее необходимые исходные данные. Формат машинной команды в IA-32 предусматривает наличие

* однобайтового префикса (повторения, размера адреса, размера операнда, замены сегмента, блокировки шины)
* кода операции
* байта режима (mod)
* байта sib (scale, index, base)
* байтов смещения в команде
* непосредственного операнда

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

Add, adc(добавляет к результату значение CF), sub, sbb(вычетает значение CF), inc, dec, mul, imul, div, idiv.

Для команд mul, imul, div, idiv операнд не может являться непосредственным значением.

Команды «развертывания» чисел — cbw(byte → word, AL → AX), cwd(word → double, AX → DX:AX), cdq(double → quadro, EAX → EDX:EAX), cwde(word → double, AX → EAX).

3. Сформулируйте основные правила построения линейной программы вычисления заданного выражения.

Все команды выполняются строго последовательно, нет передачи управления или параллельных потоков.

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

Ввод-вывод на языке ассемблера не программируют с использованием соответствующих машинных команд, так как эти операции слишком сложны для низкоуровневой реализации.

Для организации ввода-вывода используется библиотека MASM32.lib

5. Расскажите, какие процедуры использую для организации ввода вывода. Какие операции выполняет каждая процедура?

* StdIn PROC lpszBuffer:DWORD, bLen:DWORD — стандартный ввод, аргументы — адрес буфера и длина буфера
* StripLF PROC lpszBuffer:DWORD — замена символов конца строки нулем
* atol proc lpszBuffer:DWORD — преобразует строку в число и записывает в регистр EAX
* StdOut PROC lpszBuffer:DWORD – вывод строки в консоль (строка должна завершаться нулем)
* dwtoa PROC public dwValue:DWORD, lpBuffer:PTR BYTE — преобразование числа в строку

Вывод: в ходе данной работы были изучены основы и специфика целочисленной арифметики в языке ассемблера, также были изучены процедуры ввода-вывода и преобразования строк и чисел.