

## Лабораторная работа №5

### Программирование целочисленных вычислений

Цель работы:

- знакомство с работой центрального процессора и памяти;
- изучение арифметических команд Ассемблера;
- получение навыков работы с отладчиком.

Немного теории...

#### Краткое описание арифметических команд

ADD	Сложение
ADC	Сложение с переносом
SUB	Вычитание
SBB	Вычитание с заемом
IMUL	Знаковое умножение
MUL	Беззнаковое умножение
IDIV	Знаковое деление
DIV	Беззнаковое деление
INC	Инкремент
DEC	Декремент

Команда ADD выполняет арифметическое сложение двух операндов *операнд\_1* и *операнд\_2*. Операнды должны иметь одинаковый размер. Результат записывается на место *операнд\_1*.

$$\text{ADD } \text{операнд\_1}, \text{операнд\_2} \\ \text{операнд\_1} = \text{операнд\_1} + \text{операнд\_2}$$

Команда ADC (Add with Carry) производит целочисленное сложение двух знаковых или беззнаковых операндов и флага переноса. Первый операнд может быть переменной в регистре или в памяти (*r8*, *r16*, *r32*, *r/m8*, *r/m16*, *r/m32*). Второй операнд – непосредственным значением (*imm8*, *imm16*, *imm32*), переменной в регистре или в памяти. При этом оба операнда одновременно не могут быть переменными в памяти.

$$\text{ADC } \text{операнд\_1}, \text{операнд\_2} \\ \text{операнд\_1} = \text{операнд\_1} + \text{операнд\_2} + \text{флаг CF}$$

Команда ADC обычно используется в многобайтных или много-словных операциях сложения:

```

xor edx, edx           ; EDX = 0
mov  eax, 0FFFF FFFFh ; EAX = 4 294 967 29510
add  eax, 0FFFF FFFFh ; EAX = EAX + 4 294 967 29510
adc  edx, 0             ; EDX = EDX + CF (учитываем перенос)
                        ; EDX:EAX = 0000 0001h:FFFF FFEh

```

Команда SUB (SUBtract) выполняет арифметическое вычитание двух операндов *операнд\_1* и *операнд\_2*. Операнды должны иметь одинаковый размер. Результат записывается на место *операнд\_1*.

SUB *операнд\_1*, *операнд\_2*  
*операнд\_1* = *операнд\_1* - *операнд\_2*

В процессорах x86 для умножения чисел без знака предназначена команда MUL. У этой команды только один операнд — второй множитель, который должен находиться в регистре или в памяти. Местоположение первого множителя и результата задаётся неявно и зависит от размера операнда:

Размер операнда	1-й множитель	Результат
1 байт	AL	AX
2 байта	AX	DX:AX
4 байта	EAX	EDX:EAX

Запись «DX:AX» означает, что старшее слово результата будет находиться в DX, а младшее — в AX.

Пример:

```

mul bl                 ; AX = AL · BL
mul ax                 ; DX:AX = AX · AX

```

Если старшая часть результата равна нулю, то флаги CF и OF будут иметь нулевое значение. В этом случае старшую часть результата можно отбросить. Это свойство можно использовать в программе, если результат должен быть такого же размера, как множители.

Команда IMUL умножает целые числа со знаком и может использовать один, два или три операнда:

- один операнд, аналогично MUL;

- два операнда

*IMUL операнд\_1, операнд\_2*  
*операнд\_1 = операнд\_1 \* операнд\_2*

варианты: *IMUL reg, reg*  
*IMUL reg, mem*  
*IMUL reg, imm*

Пример:

```
imul edx, ecx           ; EDX = EDX * ECX  
imul ebx, A            ; EBX = EBX * A  
imul ecx, 6            ; ECX = ECX * 6
```

- три операнда

*IMUL операнд\_1, операнд\_2, операнд\_3*  
*операнд\_1 = операнд\_2 \* операнд\_3*

варианты: *IMUL reg, reg, imm*  
*IMUL reg, mem, imm*  
*IMUL reg, imm, imm.*

Пример:

```
imul ebx, A, 9          ; EBX = A * 9  
imul ecx, edx, 11       ; ECX = EDX * 11
```

В процессорах x86 для деления чисел без знака предназначена команда DIV. У этой команды только один операнд — делитель, который должен находиться в регистре или в памяти. Местоположение делимого и результата задаётся неявно и зависит от размера операнда:

Размер операнда	Делимое	Частное	Остаток
1 байт	AX	AL	AH
2 байта	DX:AX	AX	DX
4 байта	EDX:EAX	EAX	EDX

Запись «DX:AX» означает, что старшее слово результата будет находиться в DX, а младшее — в AX.

Пример:

*div bl*  
*div bx*

*; AL = AX / BL*  
*; AX = DX:AX / BX*

Команда IDIV используется для деления чисел со знаком, синтаксис ее такой же, как у команды DIV.

### Пример линейной программы

Написать программу, вычисляющую значение выражения

$$x = (a + b) \cdot (b - 1) / (d + 8).$$

*.data*

*a sword 25*

*b sword -6*

*d sword 11*

*.data?*

*x sword ?*

*.code*

*start:*

*mov cx, d*

*; CX = d*

*add cx, 8*

*; CX = d + 8*

*mov bx, b*

*; BX = b*

*dec bx*

*; BX = b - 1*

*mov ax, a*

*; AX = a*

*add ax, d*

*; AX = a + d*

*imul bx*

*; DX:AX = (a + d) · (b - 1)*

*idiv cx*

*; AX = (a + d) · (b - 1) / cx*

*mov x, ax*

*; x = AX*

*invoke ExitProcess, NULL*

*end start*

### Задание.

1. Написать программу, вычисляющую заданное в соответствии с номером студента в журнале выражение:

1.  $a = (b^2 - (c + 1) \cdot d) / b;$

2.  $c = a / c - k + (d + 1) \cdot 5;$

3.  $b = a \cdot j - j^2 / (k + 2);$

4.  $a = a \cdot (a + b / 4) / (k - 1);$

5.  $d = 3 \cdot a \cdot x / [5 \cdot (b - 5)];$

6.  $a = a \cdot x - 3 \cdot (b + 3 / k);$

7.  $a = a^3 / 3 - c \cdot (x + 3);$

8.  $d = (k - 5)^2 / 4 + 2 \cdot k;$

9.  $d = a \cdot x / 2 - (a + b) / 2;$

10.  $a = (b^2 - 2 \cdot b) / (3 \cdot a + b);$
11.  $b = (a^2 - b^2) / 2 + a \cdot (k + 1);$
12.  $e = (a - c)^2 + 2 \cdot a \cdot c / k;$
13.  $p = (t^3 - 1) / (j - 4) - 5;$
14.  $a = b^2 \cdot (y + d) + (d - 1) / c;$
15.  $s = q^3 - 2 \cdot a \cdot q + a^2 / q;$
16.  $n = q^2 / 3 - a \cdot d + 5;$
17.  $m = a \cdot c^2 - b \cdot a / c + a / b;$
18.  $x = a \cdot y \cdot (b - a) / 4 + a^2 - 2;$
19.  $n = a \cdot x^2 - b \cdot y / a + x / (y + a);$
20.  $k = (1 - a)^2 / c + k - 1 + c / 2;$
21.  $s = (a - b^2) / (y - a) + a^2 - c;$
22.  $b = (m - 5) \cdot (m + 2) + m + a / 2;$
23.  $c = (a + b) / d - d^2 \cdot a - b;$
24.  $a = b \cdot (c - d) - c / (d - 1);$
25.  $q = a^2 / 2 - b^3 / (4 - a) + b;$
26.  $s = a \cdot b / 2 - k + a / 2 - b;$

2. Открыть программу в отладчике. Указать адреса расположения переменных в памяти, заполнив таблицу:

Имя переменной	Адрес	Порядок байт в памяти

3. Выполнить программу в пошаговом режиме. После выполнения каждого шага заносить данные в таблицу:

Регистры				Флаги				прочее...
EAX	EBX	...	...	FC	FZ	...	...	

\* в таблицу должны быть включены те элементы (регистры, флаги и пр.), которые имеют место в конкретном варианте программы

4. По результатам п.3 и п.4 сделайте выводы.

### Содержание отчёта.

1. Название работы.
2. Цель работы.
3. Постановка задачи.
4. Текст программ с комментариями.
5. Результаты выполнения программы в пошаговом режиме в отладчике.
6. Выводы.