

1 Форматы данных и арифметические операции

1.1 Цель работы

- Изучение форматов представления данных в ЭВМ.
- Изучение арифметических команд МП с архитектурой i386.

1.2 Порядок выполнения работы

- Изучение методических указаний к работе.
- Выполнение общих заданий.
- Защита работы с выполнением контрольных заданий преподавателя.

1.3 Теоретические сведения

1.3.1 Основные понятия и определения

В ЭВМ различают два основных типа численных данных:

- целые двоичные числа - числа с фиксированной точкой (ФТ);
- вещественные двоичные числа - числа с плавающей точкой (ПТ).

При работе с числами на ЭВМ необходимо различать:

- значение числа;
- внутреннее представление его (внутри ЭВМ числа представляются в двоичной, 16-ричной, либо двоично-десятичной форме);
- отображение внутреннего представления в памяти ЭВМ.

Целое число X представленное в форме с ФТ (точка фиксируется после младшего разряда), например $X=1001112$, может иметь различную интерпретацию.

Целое без знака (все шесть двоичных разрядов числа являются значащими, т.е. имеющими соответствующий вес)

$$\begin{array}{ccccccccc}
 \text{СТ} & \sqrt{} & & & \sqrt{} & \text{мл} & & & \\
 5 & 4 & 3 & 2 & 1 & 0 & \text{(i-й разряд)} & & \\
 X = & 1 & 0 & 0 & 1 & 1 & 1_2 & & \\
 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 & (2^i - \text{вес } i\text{-го двоичного разряда}) &
 \end{array}$$

и тогда десятичный эквивалент его (X_{10}) можно определить по формуле (1)

$$X_{10} = \pm \sum_{i=0}^{n-1} a_i q^i \quad (1).$$

Т.к. $q=2$, а коэффициенты двоичного ряда $a_i \in \{0,1\}$, то

$$\begin{array}{ccccccc}
 X_{10} = & 1 \cdot 2^5 & + & 1 \cdot 2^2 & + & 1 \cdot 2^1 & + & 1 \cdot 2^0 = 32 + 4 + 2 + 1 = 39. & (2) \\
 & a_5 & & a_2 & & a_1 & & a_0
 \end{array}$$

Целое со знаком (старший бит не имеет веса и отображает знак). Единица в знаковом разряде – признак отрицательного числа (3) и, следовательно, приведенная последовательность – машинный код числа X (как правило, дополнительный).

1.3.2 Представление чисел в памяти

Для представления целых чисел в памяти ПК используют три машинных формата: byte, word, long (Рисунок 1). Указанные машинные форматы на языке C++ задаются соответствующими ключевыми словами (char, short, int). Форматом числа называют представление его в конкретной разрядной сетке ЭВМ, под которой понимают набор двоичных разрядов для представления машинного слова в конкретной ЭВМ.

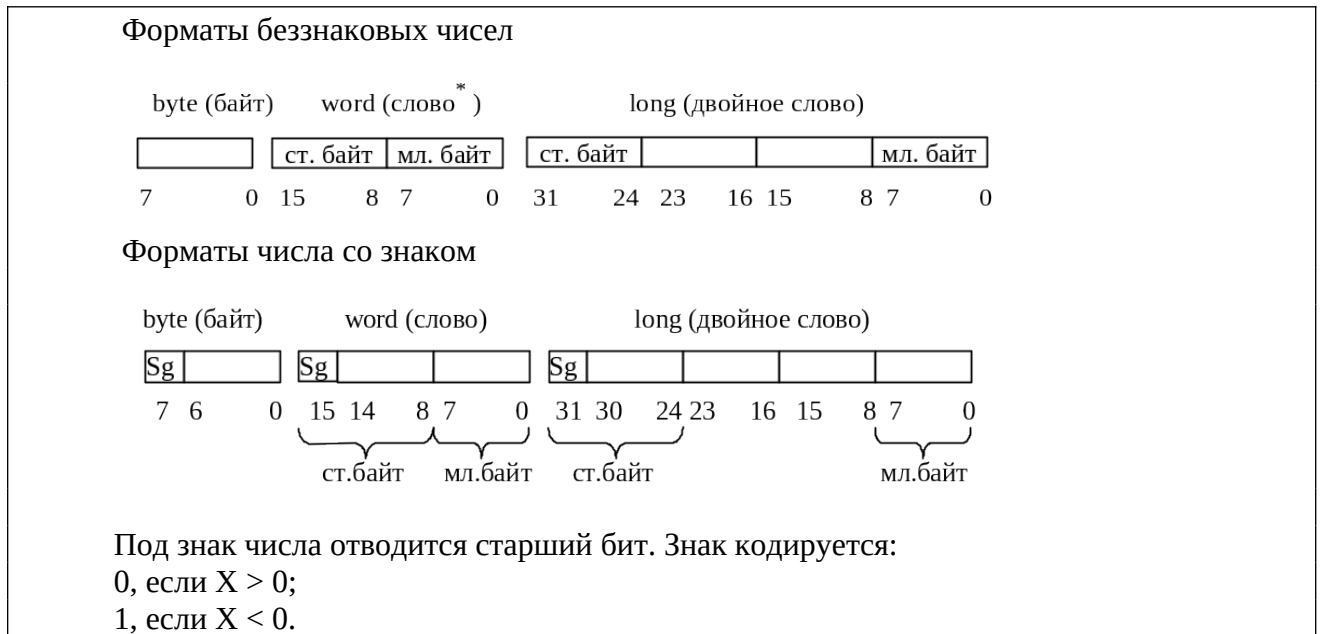


Рисунок 1 – Форматы представления целых чисел

1.3.3 Машинные коды

ЭВМ работает не с числами, а с их кодами, т.е. с машинными кодами (МК). Для хранения отрицательных чисел и выполнения арифметических операций широко используется дополнительный код (ДК). Правило образования ДК для двоичных чисел имеет вид:

$$[X]_{\text{ок}} = \begin{cases} X, & \text{если } X \geq 0 \\ \overline{X} + 1, & \text{если } X < 0 \end{cases}$$

Если X положительное число со знаком ($X = \pm 01011_2$), то ДК положительного и отрицательного X соответственно равны:

$[+X]_{\text{ДК}} = 01011_2$	$[-X]_{\text{ДК}} = \overline{X} + 1 = 10101_2$
------------------------------	---

Длина числа в ДК может быть увеличена до любого количества разрядов путем копирования (тиражирования) его знакового разряда слева. Так, рассмотренное число X в формате байта (8 бит) и слова (16 бит) имеет соответственно вид:

$$[-X]_{\text{ДК}} = 1111\ 0101_2 = \text{F5h (8 бит)} = 1111\ 1111\ 1111\ 0101_2 = \text{FFF5h (16 бит)},$$

где h – идентификатор 16-ричной системы счисления.

1.3.4 Диапазон представимых чисел

Диапазон целых чисел определяется неравенством

$$- X_{\min} \leq X \leq X_{\max}$$

и является одним из важнейших понятий. Ниже приведены диапазоны чисел для принятых форматов: байт и слово.

Байт

а) без знака: $X=00h \div FFh=0 \div 255$;

б) со знаком: $X=80h \div Fh = -128 \div +127$.

Слово

а) без знака: $X=0000h \div FFFFh=0 \div 65535$;

б) со знаком: $X=8000h \div FFFFh = -32768 \div +32767$.

Следовательно, представление беззнакового числа $X > 65535$, либо знакового $X > +32767$ приведет к переполнению 16-разрядной сетки.

1.3.5 Переход от кода к числу, т.е. $[X]_{\text{дк}} \rightarrow X$

Переход от ДК к числу выполняется по тому же правилу, что и от числа к коду:

$$[X] = \begin{cases} X_{\text{дк}}, & \text{если } X \geq 0 \\ \overline{X_{\text{дк}}} + 1, & \text{если } X < 0 \end{cases}$$

Пример. Найти десятичный эквивалент числа X , представленного в ДК, в формате байта (а) и слова (б).

а) $[X]_{\text{ДК}} = A3h = 1.0100011_2$;

$X = -01011101_2 = -5Dh = -93$.

б) $[X]_{\text{ДК}} = FFA3h = 1.1111111 \ 10100011_2 =$

$= -0000 \ 0000 \ 01011101_2 = -005Dh = -93$.

1.3.6 Арифметические команды МП

Для выполнения арифметических операций в МП x86 используется специальный набор команд, часть из которых представлены в таблице (Таблица 1).

Таблица 1 – Мнемоники арифметических и логических команд МП

Мнемоника команды	Описание
ADD D1, D2	Сложение: $D1 + D2 \rightarrow D1$ Пример: _asm ADD AX, BX
SUB D1, D2	Вычитание: $D1 - D2 \rightarrow D1$ Пример: _asm SUB AX, BX
NOT D1	Инверсия $D1 \rightarrow \overline{D1}$ Пример: _asm NOT EAX // Инверсия регистра EAX
AND D1, D2	Логическое «И» $D1 \wedge D2 \rightarrow D1$ Пример: _asm AND EAX, EBX

OR D1, D2	Логическое «ИЛИ» $D1 \vee D2 \rightarrow D1$ Пример: _asm OR EAX, EBX
DEC D1	Декремент $D1 - 1 \rightarrow D1$ Пример: _asm DEC AX
INC D1	Инкремент $D1 + 1 \rightarrow D1$ Пример: _asm INC AX
MUL D1	Умножение $D1 * AX(EAX) \rightarrow D3$ Пример: _asm MUL BX // $AX * BX \rightarrow DX:AX$ _asm MUL ECX // $EAX * ECX \rightarrow EDX:EAX$ <i>Примечание: при умножении одним из операндов всегда является аккумулятор</i>
IMUL D1	Умножение операндов со знаком <i>Примечание: при умножении одним из операндов всегда является аккумулятор</i>
DIV D1	Беззнаковое деление. Операнд – делитель. Если размер операнда - байт, то делимое находится в ax, частное возвращается в al, остаток в ah. Если слово - делимое dx:ax, частное ax, остаток dx Если двойное слово - делимое edx:eax, частное eax, остаток edx Пример: _asm DIV EBX // Делимое должно находиться в регистровой паре edx:eax, частное в EAX.
IDIV D1	Деление со знаком, команда аналогична div. Остаток всегда имеет знак делимого. <i>Примечание: для команд деления старшую часть делимого для упрощения отладки программы можно задавать равной 0.</i>

Арифметические команды кроме результата выполнения могут генерировать сигналы исключительных ситуаций (сигналы прерывания при возникновении ошибок, таких как переполнение разрядной сетки, деление на 0 и т.п.). Исключительные ситуации вызывают автоматическое прерывание процесса выполнения программы и обрабатываются операционной системой.

1.4 Практическая часть

Практическая работа состоит из двух частей:

- создание консольного приложения на языке C++ в среде Microsoft Visual Studio;
- выполнение индивидуального задания и ответы на вопросы преподавателя.

1.4.1 Создание проекта консольного приложения

Необходимо произвести следующие действия:

- Запустить среду Microsoft Visual Studio.
- Выбрать пункты меню File-New-Project.
- В появившемся диалоговом окне выбрать тип проекта «Project types – Visual C++ - Win32» (Рисунок 2).

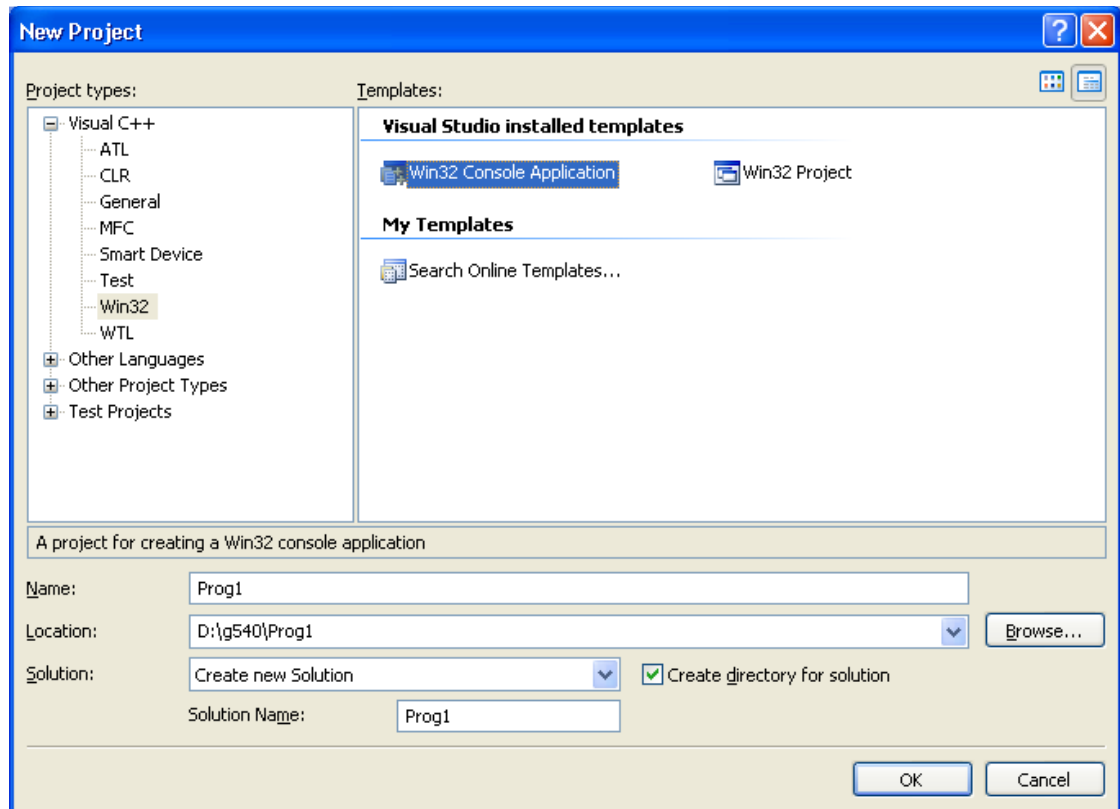


Рисунок 2 – Создание проекта консольного приложения

- Выбрать тип приложения «Win32 Console Application». Указать имя проекта (поле Name), каталог для создания нового проекта (название каталога уточните у преподавателя) и нажать кнопку Ок.

Если все сделано правильно, то на экране отобразится окно, содержащее шаблон текста программы на C++ (Рисунок 3).

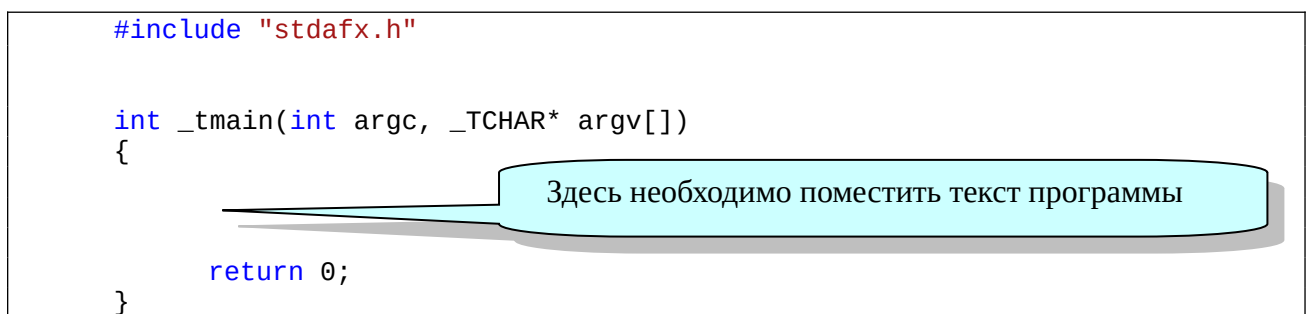


Рисунок 3 – Текст шаблона программы на C++

Текст ассемблерной программы необходимо поместить внутри функции «_tmain», которая ограничена фигурными скобками.

1.4.2 Выполнение индивидуального задания

В таблице () приведены варианты индивидуальных заданий по вариантам. Все задания должны выполняться с помощью ассемблерных вставок. Исходные данные можно размещать в переменных C++, например:

<code>int A = 0x10;</code>	//Присвоение 32-разрядной переменной значения 10h (16-ричная система)
<code>short B = 0xFF;</code>	//Присвоение 16-разрядной переменной значения FFh (16-ричная система)
<code>char C = 20;</code>	//Присвоение 8-разрядной переменной значения 20 (10-ричная система)

Таблица 2 – Варианты заданий

Вариант №	Расшифровка задания
1	Сложить 2 положительных числа. Сложить положительное и отрицательное число. Результаты записать в отдельные переменные.
2	Умножить 2 числа со знаком. Результат поместить в отдельную переменную.
3	Присвоить двум переменным значения с отрицательным знаком в дополнительном коде. Перевести числа из дополнительного кода в прямой. Сложить 2 отрицательных числа в прямом коде. Объяснить результат.
4	Вычислить разность 2х положительных чисел. Вычислить разность положительного и отрицательного числа. Результаты записать в отдельные переменные.
5	Выполнить операцию деления чисел со знаком. Результат поместить в отдельную переменную. Отдельно выполнить операцию деления на 0. Объяснить результат.
6	Выполнить операции представленные формулами: $a = \overline{x \vee y}, b = \overline{x} \wedge \overline{y}$ Сравнить значения переменных a и b.
7	Выполнить операции представленные формулами: $a = x \vee y \wedge z, b = (x \vee y) \wedge (x \vee z)$ Сравнить значения переменных a и b.
8	Выполнить операцию деления чисел без знака. Значение операнда делимого расположить в регистровой паре EDX:EAX (должно быть задано 64-битное число). Отдельно выполнить деление данного числа на 1. Объяснить результат.

Общее задание:

- представить отрицательное число в прямом коде, число должно соответствовать номеру бригады * 10;
- перевести данное число в дополнительный код;
- убедиться в правильности полученного кода отрицательного числа в режиме отладчика.

1.5 Контрольные вопросы

- Какие вы знаете арифметические операции МП?
- Какие вы знаете логические операции МП?
- Что такое дополнительный код?
- Почему в операции умножения указывается только один операнд?
- Каким образом размещается значение делимого в МП?