«Национальный исследовательский университет

«Московский институт электронной техники»

Лабораторная работа №1 по дисциплине «Архитектура вычислительных систем»

Представление данных в ЭВМ

Выполнили студены группы МП-25

Калинкин Никита Анатольевич

Констандогло Александр Витальевич

Саядян Артём Грачикович

**Москва 2017**

Вариант №2

**Задание 2.** Изучите, как интерпретируется одна и та же область памя­ти, если она рассматривается как знаковое или беззнаковое число, а также как одно и то же число записывается в различных системах счисления.

Необходимо сравнить:

а) беззнаковую интерпретацию переменной в шестнадцатеричной форме;

б) беззнаковую интерпретацию в десятичной форме;

в) знаковую интерпретацию в десятичной форме.

Для этого определите и запишите в отчёт десятичное, двоичное (16  
бит) и шестнадцатеричное представления шестнадцатибитных чисел *x* и *y*, а также беззнаковую интерпретацию этого представления в десятичном виде.

*x* = −2, *y* = 214 + 1

**Результат:**

x

Hexadecimal form: fffe

Unsigned decimal: 65534

Signed decimal: -2

Binary form: 1111111111111110

y

Hexadecimal form: 4001

Unsigned decimal: 16385

Signed decimal: 16385

Binary form: 0100000000000001

**Задание 3.** Найдите и выпишите в отчёт минимальное и максимальное  
16-битное число со знаком и без знака в формах представления (а), (б), (в)  
и в двоичной форме (4 числа, каждое из которых представлено в 4 формах).

**Результат:**

Signed max

Hexadecimal form: 7fff

Unsigned decimal: 32767

Signed decimal: 32767

Binary form: 0111111111111111

Signed min

Hexadecimal form: 8000

Unsigned decimal: 32768

Signed decimal: -32768

Binary form: 1000000000000000

Unsigned max

Hexadecimal form: ffff

Unsigned decimal: 65535

Signed decimal: -1

Binary form: 1111111111111111

Unsigned min

Hexadecimal form: 0

Unsigned decimal: 0

Signed decimal: 0

Binary form: 0000000000000000

**Задание 4.** Разработайте программу на языке C++, выполняющую над  
беззнаковыми шестнадцатибитными целыми числами следующие поразрядные  
операции (результат должен печататься в десятичной и шестнадцатеричной  
формах):

* бинарные *x* ∧ *y* (конъюнкция), *x* ∨ *y* (дизъюнкция),

*x* ⊕ *y* (сложение по модулю два);

* унарные ¬ *x* (отрицание),

neg(*x*) (дополнение до двух, *x* + neg(*x*) = 2разрядность *x*);

* *x* ≪ *y* (логический сдвиг влево), *x* ≫ *y* (логический сдвиг вправо).

**Результат:**

*Platform: Linux Ubuntu 12.04 LTS Precise Pangolin*

*Compiler: GNU GCC*

*Visual C++ 32-bit, 64-bit (Microsoft Visual Studio 2015)*

32776 & 4369 = 0 0x8008 & 0x1111 = 0x0

32776 | 4369 = 37145 0x8008 | 0x1111 = 0x9119

32776 ^ 4369 = 37145 0x8008 ^ 0x1111 = 0x9119

~32776 = 32759 ~0x8008 = 0x7ff7

~4369 = 61166 ~0x1111 = 0xeeee

neg(32776) = 32760 neg(0x8008) = 0x7ff8

neg(4369) = 61167 neg(0x1111) = 0xeeef

32776 << 4369 = 0 0x8008 << 0x1111 = 0x0

32776 >> 4369 = 0 0x8008 >> 0x1111 = 0x0

5 & 2 = 0 0x5 & 0x2 = 0x0

5 | 2 = 7 0x5 | 0x2 = 0x7

5 ^ 2 = 7 0x5 ^ 0x2 = 0x7

~5 = 65530 ~0x5 = 0xfffa

~2 = 65533 ~0x2 = 0xfffd

neg(5) = 65531 neg(0x5) = 0xfffb

neg(2) = 65534 neg(0x2) = 0xfffe

5 << 2 = 20 0x5 << 0x2 = 0x14

5 >> 2 = 1 0x5 >> 0x2 = 0x1

**Задание 5.** Измените в программе из задания 4 тип переменных на знаковый. Объясните изменение (или неизменность) результата.

**Результат:**

*Platform: Linux Ubuntu 12.04 LTS Precise Pangolin*

*Compiler: GNU GCC*

*Visual C++ 32-bit, 64-bit (Microsoft Visual Studio 2015)*

-32760 & 4369 = 0 0x8008 & 0x1111 = 0x0

-32760 | 4369 = -28391 0x8008 | 0x1111 = 0x9119

-32760 ^ 4369 = -28391 0x8008 ^ 0x1111 = 0x9119

~-32760 = 32759 ~0x8008 = 0x7ff7

~4369 = -4370 ~0x1111 = 0xeeee

neg(-32760) = 32760 neg(0x8008) = 0x7ff8

neg(4369) = -4369 neg(0x1111) = 0xeeef

-32760 << 4369 = 0 0x8008 << 0x1111 = 0x0

**-32760 >> 4369 = -1 0x8008 >> 0x1111 = 0xffff**

5 & 2 = 0 0x5 & 0x2 = 0x0

5 | 2 = 7 0x5 | 0x2 = 0x7

5 ^ 2 = 7 0x5 ^ 0x2 = 0x7

~5 = -6 ~0x5 = 0xfffa

~2 = -3 ~0x2 = 0xfffd

neg(5) = -5 neg(0x5) = 0xfffb

neg(2) = -2 neg(0x2) = 0xfffe

5 << 2 = 20 0x5 << 0x2 = 0x14

5 >> 2 = 1 0x5 >> 0x2 = 0x1

**Задание 6.** Разработайте программу на языке C++ (или дополните программу из задания 4), которая расширяет шест­надцатибитное представление числа *x* до 32-битового, рассматривая числа как

* знаковые (signed);
* беззнаковые (unsigned).

**Результат:**

*Platform: Linux Ubuntu 12.04 LTS Precise Pangolin*

*Compiler: GNU GCC*

*Visual C++ 32-bit, 64-bit (Microsoft Visual Studio 2015)*

Signed int

0xffff8008 & 0x1111 = 0x0

0xffff8008 | 0x1111 = 0xffff9119

0xffff8008 ^ 0x1111 = 0xffff9119

~0xffff8008 = 0x7ff7

~0x1111 = 0xffffeeee

neg(0xffff8008) = 0x17ff8

neg(0x1111) = 0xeeef

0xffff8008 << 0x1111 = 0x100000

0xffff8008 >> 0x1111 = 0xffffffff

0x5 & 0x2 = 0x0

0x5 | 0x2 = 0x7

0x5 ^ 0x2 = 0x7

~0x5 = 0xfffffffa

~0x2 = 0xfffffffd

neg(0x5) = 0xfffb

neg(0x2) = 0xfffe

0x5 << 0x2 = 0x14

0x5 >> 0x2 = 0x1

Unsigned int

0x8008 & 0x1111 = 0x0

0x8008 | 0x1111 = 0x9119

0x8008 ^ 0x1111 = 0x9119

~0x8008 = 0xffff7ff7

~0x1111 = 0xffffeeee

neg(0x8008) = 0x7ff8

neg(0x1111) = 0xeeef

0x8008 << 0x1111 = 0x100000

0x8008 >> 0x1111 = 0x0

0x5 & 0x2 = 0x0

0x5 | 0x2 = 0x7

0x5 ^ 0x2 = 0x7

~0x5 = 0xfffffffa

~0x2 = 0xfffffffd

neg(0x5) = 0xfffb

neg(0x2) = 0xfffe

0x5 << 0x2 = 0x14

0x5 >> 0x2 = 0x1

**Задание 7.** Определите и выпишите в отчёт, как хранятся в памяти компьютера:

* целое число 0x12345678; по результату исследования определите порядок следования байтов в словах для вашего процессора:
* Little-Endian (от младшего к старшему, порядок Intel);
* Big-Endian (от старшего к младшему, порядок Motorola);
* строки "abcd" и "абвг" (массив из char);
* «широкие» строки L"abcd" и L"абвг" (массив из wchar\_t).

**Результат:**

*Platform: Linux Ubuntu 12.04 LTS Precise Pangolin*

*Compiler: GNU GCC*

*Прямой порядок (Little-Endian)*

int x = 0x12345678;

0xffff80c: 78 56 34 12

char\* str1 = "abcd";

0x8048f60: 61 62 63 64 00 d0 b0 d0|b1 d0 b2 d0 b3 00 00 00

char\* str2 = "абвг";

0x8048f65: d0 b0 d0 b1 d0 b2 d0 b3|00 00 00 61 00 00 00 62

wchar\_t\* wstr1 = L"abcd";

0x8048f70: 61 00 00 00 62 00 00 00|63 00 00 00 64 00 00 00 a...b...c...d...

0x8048f80: 00 00 00 00 30 04 00 00|31 04 00 00 32 04 00 00

wchar\_t\* wstr2 = L"абвг";

0x8048f84: 30 04 00 00 31 04 00 00|32 04 00 00 33 04 00 00 0...1...2...3...

0x8048f94: 00 00 00 00 21 00 77 00|2f 68 6f 6d 65 2f 75 73

*Visual C++ 32-bit (Microsoft Visual Studio 2015)*

*Обратный порядок (Big-Endian)*

int x = 0x12345678;

0x004FF664 12345678 cccccccc xV4.ММММ

char\* str1 = "abcd";

0x001A6358 64636261 00000000 abcd....

0x001A6360 00000000 00000000 ........

char\* str2 = "абвг";

0x001A6368 e3e2e1e0 00000000 абвг....

0x001A6370 00000000 00000000 ........

wchar\_t\* wstr1 = L"abcd";

0x001A6378 00620061 00640063 a.b.c.d.

0x001A6380 00000000 00000000 ........

wchar\_t\* wstr2 = L"абвг";

0x001A6388 04310430 04330432 0.1.2.3.

*Visual C++ 64-bit (Microsoft Visual Studio 2015)*

*Обратный порядок (Big-Endian)*

int x = 0x12345678;

0x0000009426DBFA14 12345678 cccccccc xV4.ММММММММ

char\* str1 = "abcd";

0x00007FF676ED8218 64636261 00000000 abcd....

char\* str2 = "абвг";

0x00007FF676ED8220 e3e2e1e0 00000000 абвг....

wchar\_t\* wstr1 = L"abcd";

0x00007FF676ED8228 00620061 00640063 a.b.c.d.

0x00007FF676ED8230 00000000 00000000 ........

wchar\_t\* wstr2 = L"абвг";

0x00007FF676ED8238 04310430 04330432 0.1.2.3.

0x00007FF676ED8240 00000000 00000000 ........

*Platform: Linux Ubuntu 16.04.3 LTS x86\_64*

*Compiler: GNU GCC 5.3.1*

*IDE: Qt Creator 4.3.1*

*Прямой порядок (Little-Endian)*

int x = 0x12345678;

0x40156e <+0x0017> c7 45 d4 78 56 34 12

char\* str1 = "abcd";

0x40158b <+0x0034> 48 8d 05 44 30 00 00

0x401592 <+0x003b> 48 89 45 d8

char\* str2 = "Ð°Ð±Ð²Ð³";

0x401596 <+0x003f> 48 8d 05 3e 30 00 00

0x40159d <+0x0046> 48 89 45 e0

wchar\_t\* wstr1 = L"abcd";

0x4015a1 <+0x004a> 48 8d 05 3c 30 00 00

0x4015a8 <+0x0051> 48 89 45 e8

wchar\_t\* wstr2 = L"Ð°Ð±Ð²Ð³";

0x4015ac <+0x0055> 48 8d 05 45 30 00 00

0x4015b3 <+0x005c> 48 89 45 f0

**Задание 8.** Напишите программу, демонстрирующую переполнение целых чисел со знаком и без знака. Для этого опишите целое число со знаком и инициализи­руйте его максимально возможным для данного типа значением, результат распечатайте в шестнадцатеричном виде. Затем прибавьте к этому числу единицу, результат распечатайте также в шестнадцатеричном виде. Выполните аналогичные действия для беззнаковой переменной. Выпишите в отчёт распечатанные значения переменной, поясните полученные результаты.

**Результат:**

*Platform: Linux Ubuntu 12.04 LTS Precise Pangolin*

*Compiler: GNU GCC*

*Visual C++ 32-bit, 64-bit (Microsoft Visual Studio 2015)*

Signed short

SHRT\_MAX = 0x7fff SHRT\_MAX + 1 = 0x8000

Unsigned short

USHRT\_MAX = 0xffff USHRT\_MAX + 1 = 0x0

**Задание 9.** При помощи оператора sizeof выясните, сколько байтов занимают переменные следующих типов: char, bool, wchar\_t, short, int, long, long long, float, double, long double, size\_t, ptrdiff\_t, void\*. Результаты оформите в отчёте в виде таблицы, указывая для каждого типа его назначение.

Для выполнения единообразных действий над переменными различных типов используются макросы препроцессора C или шаблоны C++.

Проверьте, соответствуют ли размеры типов современному стандарту C++.

**Результат:**

*Platform: Linux Ubuntu 12.04 LTS Precise Pangolin*

*Compiler: GNU GCC*

1 char

1 bool

4 wchar\_t

2 short

4 int

4 long

8 long long

4 float

8 double

12 long double

4 size\_t

4 ptrdiff\_t

4 void\*

*Visual C++ 32-bit (Microsoft Visual Studio 2008/2010/2015)*

1 char

1 bool

2 wchar\_t

2 short

4 int

4 long

8 long long

4 float

8 double

8 long double

4 size\_t

4 ptrdiff\_t

4 void\*

*Visual C++ 64-bit (Microsoft Visual Studio 2008/2010/2015)*

1 char

1 bool

2 wchar\_t

2 short

4 int

4 long

8 long long

4 float

8 double

8 long double

8 size\_t

8 ptrdiff\_t

8 void\*

*Platform: Linux Ubuntu 16.04.3 LTS x86\_64*

*Compiler: GNU GCC 5.3.1*

*IDE: Qt Creator 4.3.1*

1 char

1 bool

4 wchar\_t

2 short

4 int

8 long

8 long long

4 float

8 double

10 long double

8 size\_t

8 ptrdiff\_t

8 void\*

**Задание 10.** Запустите программу (программы) из заданий 4-9 на двух других платформах, доступных на ВЦ — 32- и 64-разрядной версиях Microsoft Windows и повторите измерения.

Для каждого из заданий дополните таблицу результатами новых измере­ний. Платформы, для которых необходимо провести измерения:

* GNU/Linux Ubuntu, архитектура x86 (компилятор GCC, среда любая);
* 32-разрядная версия Microsoft Windows (ВЦ, среда Microsoft Visual  
  Studio);
* 64-разрядная версия Microsoft Windows (терминал Skylab, среда Microsoft Visual Studio, 64-битная конфигурация).

Результаты однотипных измерений, выполненных на различных платфор­мах, должны быть сгруппированы в таблицу или последовательно описаны в одном разделе.