**“ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА”**

**(СПбГУТ)**

**Факультет инфокоммуникационных Сетей и систем (иксс)**

**кафедра программной инженерии и вычислительной техники (пи и вт)**

Дисциплина: «Машинно-зависимые языки программирования»

Лабораторная работа №2.

**Тема: «Организация условных переходов»**

**вариант №1**

Выполнил: Студент группы ИКПИ-95

Алюшин В.В.

Подпись\_\_\_\_\_\_\_\_\_\_\_\_

Принял: Коробов С.А.

Подпись\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_\_»\_\_\_\_\_\_\_\_ 2020

2020г

**ЦЕЛЬ РАБОТЫ:**

Вычислить заданное условное целочисленное выражение для данных в форматах INTEGER (int), WORD (unsigned int), используя команды сравнения, условного и безусловного переходов. Результат **X** тоже целочисленный и его диапазон (формат) зависит от специфики решаемого условного выражения. Исходные данные должны вводиться корректно(с проверкой на область допустимых значений). Результат также должен быть проверен на область допустимых значений. Обмен данными между **Си** и **ASM** - модулем должен осуществляться через глобальные переменные, определенные в модуле **Си**.

**(a+2)/b , если a > b**

**21 , a = b**

**(a\*b)/4 , если a < b**

**Таблица Результатов signed**

|  |  |  |
| --- | --- | --- |
| **a** | **b** | **Результат** |
| **1** | **2** | **0** |
| **10** | **-1** | **-12** |
| **12** | **13** | **39** |

**Таблица Результатов unsigned**

|  |  |  |
| --- | --- | --- |
| **a** | **b** | **Результат** |
| **34** | **45** | **382** |
| **6** | **8** | **12** |
| **11** | **13** | **35** |

**Код:**

**Файл main.c:**

#include <stdio.h>

#include <io.h>

#include <fcntl.h>

#include <conio.h>

extern "C"

{

signed \_\_int16 a\_sig;

signed \_\_int16 b\_sig;

unsigned \_\_int16 a\_unsig;

unsigned \_\_int16 b\_unsig;

signed \_\_int16 x\_return;

void signedF();

void unsignedF();

}

void signedFT()

{

int a = a\_sig;

int b = b\_sig;

if (a < b)

{

wprintf(L"Тест x: %i\n", (a \* b) / 4);

}

else if (a == b)

{

wprintf(L"Тест x: %i\n", 21);

}

else if (a > b)

{

wprintf(L"Тест x: %i\n", (a + 2) / b);

}

}

void unsignedFT()

{

int a = a\_unsig;

int b = b\_unsig;

if (a < b)

{

wprintf(L"Тест x: %i\n", (a \* b) / 4);

}

else if (a == b)

{

wprintf(L"Тест x: %i\n", 21);

}

else if (a > b)

{

wprintf(L"Тест x: %i\n", (a + 2) / b);

}

}

int sig()

{

int data;

wprintf(L"a: ");

wscanf(L"%i", &data);

a\_sig = data;

wprintf(L"b: ");

wscanf(L"%i", &data);

b\_sig = data;

signedFT();

signedF();

return x\_return;

}

int unsig()

{

int data;

wprintf(L"a: ");

wscanf(L"%i", &data);

a\_unsig = data;

wprintf(L"b: ");

wscanf(L"%i", &data);

b\_unsig = data;

unsignedFT();

unsignedF();

return x\_return;

}

int main()

{

//рус. яз.

\_setmode(\_fileno(stdout), \_O\_U16TEXT);

\_setmode(\_fileno(stdin), \_O\_U16TEXT);

\_setmode(\_fileno(stderr), \_O\_U16TEXT);

while (1)

{

int mode;

wprintf(L"1 Знаковый\n");

wprintf(L"2 Беззнаковый\n");

wprintf(L"Enter mode: ");

wscanf(L"%i", &mode);

switch (mode)

{

case(1):

{

wprintf(L"x: %i\n", sig());

break;

}

case(2):

{

wprintf(L"x: %i\n", unsig());

break;

}

default:

break;

}

}

\_getch();

return 0;

}

**Файл ASM (ASM):**

;.686

;.model flat, c

.data

extern a\_sig: word

extern b\_sig: word

extern a\_unsig: word

extern b\_unsig: word

extern x\_return: word

.code

; signed

;;;;;;;;;;;;;;;;;;;;;;;;

;a == b

signed\_s1 PROC

mov ax, 21d

mov x\_return, ax

ret

signed\_s1 ENDP

;a > b

signed\_s2 PROC

mov ax, a\_sig

add ax, 2

cwd

idiv b\_sig

mov x\_return, ax

ret

signed\_s2 ENDP

;a < b

signed\_s3 PROC

mov ax, a\_sig

cwd

imul b\_sig

mov bx, 4d

idiv bx

mov x\_return, ax

ret

signed\_s3 ENDP

signedF PROC

mov ax, a\_sig

mov bx, b\_sig

cmp ax, bx

je signed\_s1 ;ax==bx

jg signed\_s2 ;ax>bx

jl signed\_s3 ;ax<bx

signedF ENDP

; unsigned

;;;;;;;;;;;;;;;;;;;;;;;;

;a == b

unsigned\_s1 PROC

mov ax, 21d

mov x\_return, ax

ret

unsigned\_s1 ENDP

;a > b

unsigned\_s2 PROC

mov ax, a\_unsig

add ax, 2

xor dx, dx

idiv b\_unsig

mov x\_return, ax

ret

unsigned\_s2 ENDP

;a < b

unsigned\_s3 PROC

mov ax, a\_unsig

xor dx, dx

imul b\_unsig

mov bx, 4d

idiv bx

mov x\_return, ax

ret

unsigned\_s3 ENDP

unsignedF PROC

mov ax, a\_unsig

mov bx, b\_unsig

cmp ax, bx

je unsigned\_s1 ;ax==bx

jg unsigned\_s2 ;ax>bx

jl unsigned\_s3 ;ax<bx

unsignedF ENDP

END