Федеральное государственное бюджетное образовательное учреждение высшего образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича»

Кафедра «Программная инженерия и вычислительная техника»

«Машино-зависимые языки программирования»

Отчет

по лабораторной работе №2

«Организация условных переходов»

|  |  |
| --- | --- |
|  | Выполнила |
|  | студентка группы ИКПИ-25 |
|  | А. Ю. Яскунова |
|  |  |
|  | Проверил |
|  | Ст. преподаватель |
|  | И.Л. Савельев |
|  |  |

2023 г.

**1. Задание**

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.

**Вариант №26**

b\*a-20, если a < b

20, если a = b

9\*a/b, если a > b

**2. Текст программы**

**2.1 Модуль integer.c**

#include <stdio.h>

#include <stdint.h>

#include <inttypes.h>

int a, b, result\_asm\_denom;

long result, result\_asm\_num;

extern void calculate\_expression\_asm();

long calculate\_expression(int a, int b) {

if (a < b) {

return b \* a - 20;

} else if (a == b) {

return 20;

} else {

return 9 \* a / b;

}

}

int check\_int(int num) {

if (num >= -32768 && num<=32767){

printf("\nValue %d is correct", num);

return 1;

}

else{

printf("\nValue %d has error!\n", num);

return 0;

}

}

int main() {

printf("b\*a-20, если a<b\n20, если a=b\n9\*a/b, если a>b\n");

printf("Enter values for a and b from -32768 to 32767: ");

scanf("%d %d", &a, &b);

if (check\_int(a) == 1 && check\_int(b) == 1){

result = calculate\_expression(a, b);

printf("\n\nC Result: X=%ld\n", result);

calculate\_expression\_asm();

printf("\n\nAssembly Result: X=%ld\n", result\_asm\_num);

}

printf("\n");

return 0;

}

**2.2 Модуль asm\_integer.asm**

section .data

extern a

extern b

extern result\_asm\_num

; b\*a-20, если a<b 20, если a=b 9\*a/b, если a>b

section .text

global calculate\_expression\_asm

calculate\_expression\_asm:

mov eax, [a]

mov ebx, [b]

cmp eax, ebx ; Сравнение a и b

jl less\_than ; Если a < b, переход к метке less\_than

je equal ; Если a = b, переход к метке equal

jg greater\_than ; Если a > b, переход к метке greater\_than

less\_than:

mov eax, dword [a] ; Загрузка значения a в EAX

imul eax, dword [b] ; Умножение EAX на b

sub eax, 20 ; Вычитание 20 из EAX

mov [result\_asm\_num], eax ; Сохранение результата в result\_asm\_num

ret

equal:

mov eax, 20 ; Загрузка значения 20 в EAX

mov [result\_asm\_num], eax ; Сохранение результата в result\_asm\_num

ret

greater\_than:

cmp ebx, 0

je error

mov eax, 9 ; Загрузка значения 9 в EAX

imul eax, dword [a] ; Умножение EAX на a

cdq ; Расширение знака из EAX в EDX:EAX

idiv dword [b] ; Деление EDX:EAX на b

mov [result\_asm\_num], eax ; Сохранение результата в result\_asm\_num

ret

error:

ret

**2.3 Модуль word.c**

#include <stdio.h>

#include <stdint.h>

unsigned int a, b;

int result, result\_asm\_num, result\_asm\_denom;

extern void calculate\_expression\_asm();

int calculate\_expression(unsigned int a, unsigned int b) {

if (a < b) {

return b \* a - 20;

} else if (a == b) {

return 20;

} else {

if (b == 0){

printf("Error! b cannot be 0");

return 0;

}

else

return 9 \* a / b;

}

}

int check\_int(int num) {

if (num >= -32768 && num<=32767){

printf("\nValue %d is correct", num);

return 1;

}

else{

printf("\nValue %d has error!\n", num);

return 0;

}

}

unsigned int check\_uint(unsigned int num) {

if (num >= 0 && num<=65535){

printf("\nValue %u is correct", num);

return 1;

}

else{

printf("\nValue %u has error!\n", num);

return 0;

}

}

int main() {

printf("b\*a-20, если a<b\n20, если a=b\n9\*a/b, если a>b\n");

printf("Enter values for a and b from 0 to 65535: ");

scanf("%u %u", &a, &b);

if (check\_uint(a) == 1 && check\_uint(b) == 1){

result = calculate\_expression(a, b);

printf("\n\nC Result: X=%d\n", result);

check\_int(result);

calculate\_expression\_asm();

printf("\n\nAssembly Result: X=%d\n", result\_asm\_num);

check\_int(result\_asm\_num);

}

printf("\n");

return 0;

}

**2.3. Модуль makefile**

all:

gcc -c -g -ggdb -o integer.o integer.c

gcc -c -g -ggdb -o word.o word.c

nasm -f elf64 -F dwarf -g asm\_integer.asm

gcc -m64 -o integer integer.o asm\_integer.o -g -ggdb -fno-pie -no-pie

gcc -m64 -o word word.o asm\_integer.o -g -ggdb -fno-pie -no-pie

**3. Сборка проекта**

make

**4. Выполнение программы**

**4.1. Запуск программы**

./integer

./word

**4.2. Входные данные**

**Для integer:**

a = -32000

b= -30000

**Для word:**

a = 65000

b= 32384

**4.3. Ожидаемый результат выполнения**

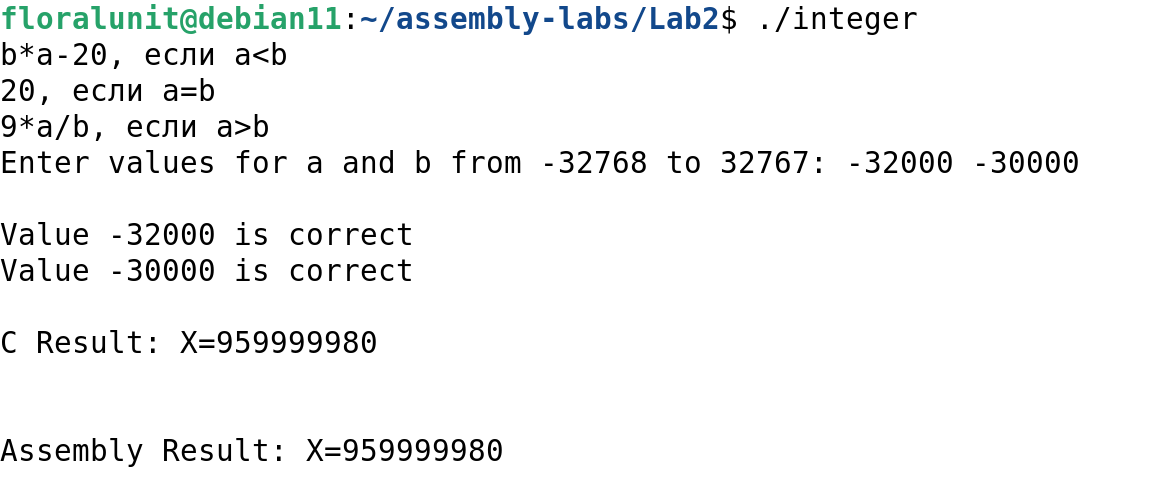
**Для integer:**

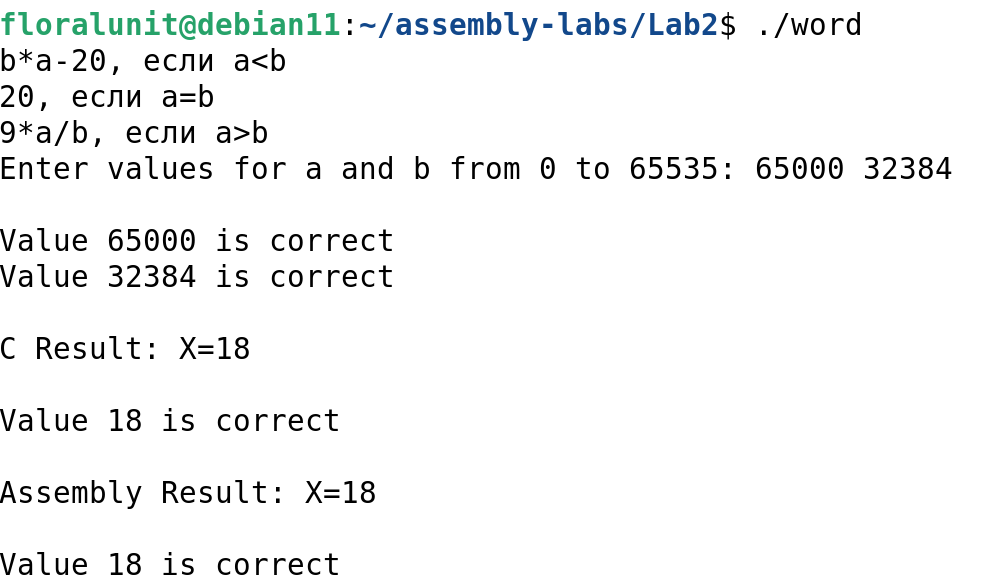
959999980

**Для word:**

18

**4.4. Результат выполнения**





**5. Вывод**

Результат выполнения программы соответствует ожидаемому результату. Работа выполнена в полном объеме.