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»

Выполнил

студент группы ИКПИ-33

А.Р.Коломиец

Проверил

Ст. преподаватель

И.Л. Савельев

2024 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 №7

(a+12)/5 , если a < b
a/b-21 , если a > b
210 , a = b

2. Текст программы

2.1. Модуль main.c

```
// main.c
#include <stdio.h>
#include <stdint.h> // Для использования int64_t и uint64_t
#include <limits.h>

// Глобальные переменные для обмена данными между Си и ASM
// Signed int
int64_t a_signed, b_signed; // Исходные данные
int64_t x_signed;           // Результат

// Unsigned int
uint64_t a_unsigned, b_unsigned; // Исходные данные
uint64_t x_unsigned;             // Результат

// Прототипы функций ASM
void signed_int();     // Функция для signed int
void unsigned_int();   // Функция для unsigned int

int main() {
    int choice;
    printf("Выберите тип данных:\n");
```

```
        printf("1. Signed int\n");
        printf("2. Unsigned int\n");
        printf("Ваш выбор: ");
        if (scanf("%d", &choice) != 1) {
            printf("Неверный выбор.\n");
            return 1;
        }

        if (choice == 1) {
            // Ввод значений для signed int
            printf("Введите целое число a (signed int): ");
            while (scanf("%lld", &a_signed) != 1) {
                printf("Неверный ввод. Пожалуйста, введите целое
число a: ");
                while (getchar() != '\n');
            }

            printf("Введите целое число b (signed int): ");
            while (scanf("%lld", &b_signed) != 1) {
                printf("Неверный ввод. Пожалуйста, введите целое
число b: ");
                while (getchar() != '\n');
            }

            // Проверка на деление на ноль при a > b
            if ((a_signed > b_signed) && (b_signed == 0)) {
                printf("Ошибка: Деление на ноль при a > b.\n");
                return 1;
            }

            // Вызов ASM-функции для signed int
            signed_int();

            // Вывод результата
            printf("Результат X (signed int): %lld\n", x_signed);
        } else if (choice == 2) {
            // Ввод значений для unsigned int
            printf("Введите целое число a (unsigned int): ");
            while (scanf("%llu", &a_unsigned) != 1) {
                printf("Неверный ввод. Пожалуйста, введите целое
число a: ");
                while (getchar() != '\n');
            }

            printf("Введите целое число b (unsigned int): ");
            while (scanf("%llu", &b_unsigned) != 1) {
```

```
                printf("Неверный ввод. Пожалуйста, введите целое
число b: ");
                while (getchar() != '\n');
            }

            // Проверка на деление на ноль при a > b
            if ((a_unsigned > b_unsigned) && (b_unsigned == 0)) {
                printf("Ошибка: Деление на ноль при a > b.\n");
                return 1;
            }

            // Вызов ASM-функции для unsigned int
            unsigned_int();

            // Вывод результата
            printf("Результат X (unsigned int): %llu\n", x_unsigned);
        } else {
            printf("Неверный выбор.\n");
            return 1;
        }

        return 0;
    }
```

2.2. Модуль signed_int.asm

```
    ; signed_int.asm
    global signed_int
    extern a_signed
    extern b_signed
    extern x_signed

    section .text

    signed_int:
        ; Пролог функции
        push rbp
        mov rbp, rsp

        ; Загружаем значения 'a' и 'b' из глобальных переменных
        mov rax, [rel a_signed]
        mov rbx, [rel b_signed]

        ; Сравниваем 'a' и 'b' (signed comparison)
        cmp rax, rbx
        jl less_than    ; a < b
        jg greater_than ; a > b
```

```
        ; Если ни одно из вышеуказанных условий не выполнено, то a ==
b
        jmp equal

    less_than:
        ; X = (a + 12) / 5
        add rax, 12
        cqo             ; Расширение знака для деления (64-битный
эквивалент cdq)
        mov rcx, 5
        idiv rcx        ; Signed division
        mov [rel x_signed], rax    ; Сохраняем результат в 'x_signed'
        jmp end_function

    greater_than:
        ; Проверка на деление на ноль
        cmp rbx, 0
        je division_by_zero
        ; X = a / b - 21
        mov rax, [rel a_signed]
        cqo             ; Расширение знака для деления
        idiv rbx        ; Signed division
        sub rax, 21
        mov [rel x_signed], rax    ; Сохраняем результат в 'x_signed'
        jmp end_function

    equal:
        ; X = 210
        mov rax, 210
        mov [rel x_signed], rax    ; Сохраняем результат в 'x_signed'
        jmp end_function

    division_by_zero:
        ; Установка кода ошибки в 'x_signed'
        mov rax, 0x7FFFFFFFFFFFFFFF  ; INT64_MAX
        mov [rel x_signed], rax

    end_function:
        ; Эпилог функции
        mov rsp, rbp
        pop rbp
        ret
```

2.3. Модуль unsigned_int.asm

```
; unsigned_int.asm
global unsigned_int
```

```
    extern a_unsigned
    extern b_unsigned
    extern x_unsigned

    section .text

    unsigned_int:
        ; Пролог функции
        push rbp
        mov rbp, rsp

        ; Загружаем значения 'a' и 'b' из глобальных переменных
        mov rax, [rel a_unsigned]
        mov rbx, [rel b_unsigned]

        ; Сравниваем 'a' и 'b' (unsigned comparison)
        cmp rax, rbx
        jb less_than    ; a < b
        ja greater_than ; a > b
        ; Если ни одно из вышеуказанных условий не выполнено, то a ==
b
        jmp equal

    less_than:
        ; X = (a + 12) / 5
        add rax, 12
        xor rdx, rdx    ; Обнуляем rdx для unsigned деления
        mov rcx, 5
        div rcx         ; Unsigned division
        mov [rel x_unsigned], rax    ; Сохраняем результат в
'x_unsigned'
        jmp end_function

    greater_than:
        ; Проверка на деление на ноль
        cmp rbx, 0
        je division_by_zero
        ; X = a / b - 21
        mov rax, [rel a_unsigned]
        xor rdx, rdx    ; Обнуляем rdx для unsigned деления
        div rbx         ; Unsigned division
        sub rax, 21
        mov [rel x_unsigned], rax    ; Сохраняем результат в
'x_unsigned'
        jmp end_function
```

```
    equal:
        ; X = 210
        mov rax, 210
        mov [rel x_unsigned], rax    ; Сохраняем результат в
'x_unsigned'
        jmp end_function

    division_by_zero:
        ; Установка кода ошибки в 'x_unsigned'
        mov rax, 0xFFFFFFFFFFFFFFFF  ; UINT64_MAX
        mov [rel x_unsigned], rax

    end_function:
        ; Эпилог функции
        mov rsp, rbp
        pop rbp
        ret
```

2.4 Модуль Makefile

```
    all:
         gcc -m64 -c -g -o main.o main.c
         nasm -f elf64 signed_int.asm -o signed_int.o
         nasm -f elf64 unsigned_int.asm -o unsigned_int.o
         gcc -m64 -no-pie -o program main.o signed_int.o
unsigned_int.o

    clean:
         rm -f *.o program
```

3. Сборка проекта

```
make
```

4. Выполнение программы

4.1. Запуск программы

```
./program
```

4.2. Входные данные

Выберите тип переменных:

1 - signed char

2 - unsigned int

Ваш выбор: 2

Введите a: 12

Введите b: 23

4.3. Ожидаемый результат выполнения

Корректно подсчитанный результат выполнения уравнения на языках программирования C и ASM. Для данного набора переменных - Результат: 4

4.4. Результат выполнения

```
ikpi33n40@termserver2:~/gitea/2курс_мзяп/2lab$ ./program
Выберите тип данных:
1. Signed int
2. Unsigned int
Ваш выбор: 2
Введите целое число a (unsigned int): 12
Введите целое число b (unsigned int): 23
Результат X (unsigned int): 4
```

5. Вывод

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