###### МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

###### ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

###### НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

###### Факультет информационных технологий

**Кафедра параллельных вычислений**

ОТЧЕТ

О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ

«ВВЕДЕНИЕ В АРХИТЕКТУРУ ARM»

студента 2 курса, 22202 группы

**Бальчинова А.С.**

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:

В.А. Перепёлкин

Новосибирск 2023

**СОДЕРЖАНИЕ**

[ЦЕЛЬ 3](#_Toc18443921)

[ЗАДАНИЕ 3](#_Toc18443922)

[ОПИСАНИЕ РАБОТЫ](#_Toc18443923) 3-6

[ЗАКЛЮЧЕНИЕ 7](#_Toc18443924)

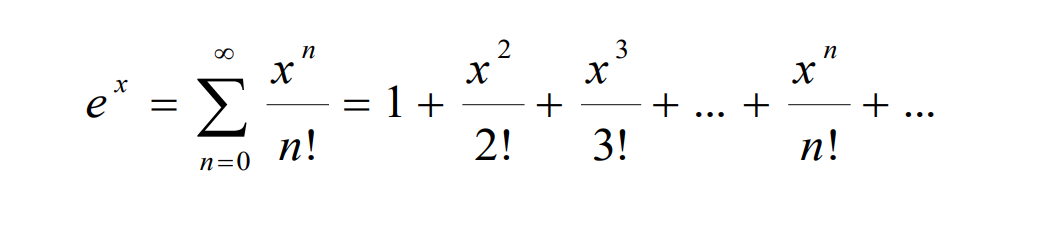
# ЦЕЛИ

1. Знакомство с программной архитектурой ARM.

2. Анализ ассемблерного листинга программы для архитектуры ARM.

# ЗАДАНИЕ

Алгоритм вычисления функции ex с помощью разложения в ряд Маклорена по первым N членам этого ряда:



Область сходимости ряда: − ∞ ≤ x ≤ ∞ .

# ОПИСАНИЕ РАБОТЫ

1. Реализация алгоритма из задания на языке программирования C.

#include <stdio.h>

#include <stdlib.h>

double exponent(double x, int N) {

    double exp = 1;

    double p = 1;

    double n\_fact = 1;

    for (int i = 1; i <= N; i++) {

        p \*= x;

        n\_fact \*= i;

        exp += (p / n\_fact);

    }

    return exp;

}

int main() {

    double x = 1;

    int N = 1000000;

    printf("%f", exponent(x, N));

    return 0;

}

Проводим анализ листингов: изучаем использованные команды и регистры, сопоставляем с исходным кодом.

Сравниваем листинги с разными уровнями оптимизаций, выявляем различия:

1. В -O2 происходит перемешивание команд, из-за чего их чтение становится труднее.
2. В –O2 обращения в память происходят намного реже, чем в –O2.
3. В –O2 не происходит вызов функции exponent, а все вычисления происходят в main и .L8.
4. В –O2 используется инструкция bne, не используется инструкция безусловного перехода b.

**Ассемблерный листинг для уровня оптимизации O0 (ARM64 GCC 13.2.0)**

|  |
| --- |
| 1 exponent:  2        sub     sp, sp, #48  3        str     d0, [sp, 8]  4        str     w0, [sp, 4]  5        fmov    d0, 1.0e+0  6        str     d0, [sp, 40]  7        fmov    d0, 1.0e+0  8        str     d0, [sp, 32]  9        fmov    d0, 1.0e+0  10        str     d0, [sp, 24]  11        mov     w0, 1  12        str     w0, [sp, 20]  13        b       .L2  14 .L3:  15        ldr     d1, [sp, 32]  16        ldr     d0, [sp, 8]  17        fmul    d0, d1, d0  18        str     d0, [sp, 32]  19        ldr     w0, [sp, 20]  20        scvtf   d0, w0  21        ldr     d1, [sp, 24]  22        fmul    d0, d1, d0  23        str     d0, [sp, 24]  24        ldr     d0, [sp, 24]  25        ldr     d1, [sp, 32]  26        fdiv    d0, d1, d0  27        ldr     d1, [sp, 40]  28        fadd    d0, d1, d0  29        str     d0, [sp, 40]  30        ldr     w0, [sp, 20]  31        add     w0, w0, 1  32        str     w0, [sp, 20]  33 .L2:  34       ldr     w1, [sp, 20]  35        ldr     w0, [sp, 4]  36        cmp     w1, w0  37        ble     .L3  38        ldr     d0, [sp, 40]  39        add     sp, sp, 48  40        ret  41 .LC0:  42        .string "%f"  43 main:  44        stp     x29, x30, [sp, -32]!  45        mov     x29, sp  46        fmov    d0, 1.0e+0  47        str     d0, [sp, 24]  48        mov     w0, 16960  49        movk    w0, 0xf, lsl 16  50        str     w0, [sp, 20]  51        ldr     w0, [sp, 20]  52        ldr     d0, [sp, 24]  53        bl      exponent  54        adrp    x0, .LC0  55        add     x0, x0, :lo12:.LC0  56        bl      printf  57        mov     w0, 0  58        ldp     x29, x30, [sp], 32  59        ret |

**Ассемблерный листинг для уровня оптимизации O2 (ARM64 GCC 13.2.0)**

|  |
| --- |
| 1 exponent:  2        fmov    d4, d0  3        cmp     w0, 0  4        ble     .L4  5        fmov    d1, 1.0e+0  6        add     w1, w0, 1  7        mov     w0, 1  8        fmov    d2, d1  9        fmov    d0, d1  10 .L3:  11        scvtf   d3, w0  12        fmul    d2, d2, d4  13        add     w0, w0, 1  14        fmul    d1, d1, d3  15        fdiv    d3, d2, d1  16        fadd    d0, d0, d3  17        cmp     w1, w0  18        bne     .L3  19        ret  20 .L4:  21        fmov    d0, 1.0e+0  22        ret  23 .LC0:  24        .string "%f"  25 main:  26        fmov    d0, 1.0e+0  27        mov     w1, 16961  28        stp     x29, x30, [sp, -16]!  29        mov     w0, 1  30        fmov    d1, d0  31        fmov    d3, d0  32        movk    w1, 0xf, lsl 16  33        mov     x29, sp  34 .L8:  35        scvtf   d2, w0  36        add     w0, w0, 1  37        fmul    d1, d1, d2  38        fdiv    d2, d3, d1  39        fadd    d0, d0, d2  40        cmp     w0, w1  41        bne     .L8  42        adrp    x0, .LC0  43        add     x0, x0, :lo12:.LC0  44        bl      printf  45        mov     w0, 0  46        ldp     x29, x30, [sp], 16  47        ret |

# ЗАКЛЮЧЕНИЕ

Ознакомились с программной архитектурой ARM64.

Были проанализированы ассемблерные листинги программы для архитектуры

ARM64 и их различия при использивании уровней оптимизации -O0 и -O2.