###### 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 x86/x86-64»

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) 3-4

[ЗАКЛЮЧЕНИЕ 7](#_Toc18443924)

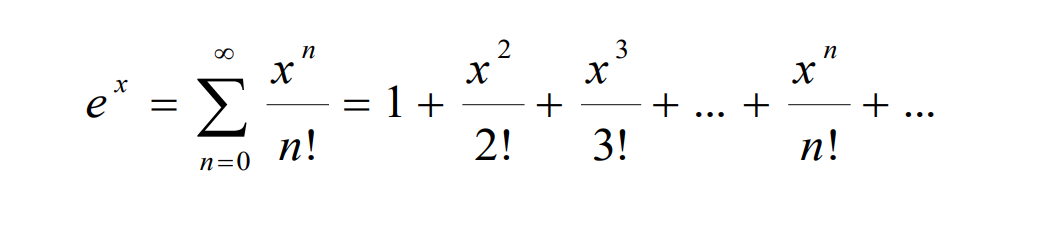
# ЦЕЛИ

1. Знакомство с программной архитектурой x86/x86-64.

2. Анализ ассемблерного листинга программы для архитектуры x86-64.

# 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 используются команды test, jne, xor; не используются команды leave, push

При оптимизации -O2 отличаются некоторые используемые команды.

**Ассемблерный листинг для уровня оптимизации O0 (x86-64 gcc 13.2)**

exponent:

        push    rbp

        mov     rbp, rsp

        movsd   QWORD PTR [rbp-40], xmm0

        mov     DWORD PTR [rbp-44], edi

        movsd   xmm0, QWORD PTR .LC0[rip]

        movsd   QWORD PTR [rbp-8], xmm0

        movsd   xmm0, QWORD PTR .LC0[rip]

        movsd   QWORD PTR [rbp-16], xmm0

        movsd   xmm0, QWORD PTR .LC0[rip]

        movsd   QWORD PTR [rbp-24], xmm0

        mov     DWORD PTR [rbp-28], 1

        jmp     .L2

.L3:

        movsd   xmm0, QWORD PTR [rbp-16]

        mulsd   xmm0, QWORD PTR [rbp-40]

        movsd   QWORD PTR [rbp-16], xmm0

        pxor    xmm0, xmm0

        cvtsi2sd        xmm0, DWORD PTR [rbp-28]

        movsd   xmm1, QWORD PTR [rbp-24]

        mulsd   xmm0, xmm1

        movsd   QWORD PTR [rbp-24], xmm0

        movsd   xmm0, QWORD PTR [rbp-16]

        divsd   xmm0, QWORD PTR [rbp-24]

        movsd   xmm1, QWORD PTR [rbp-8]

        addsd   xmm0, xmm1

        movsd   QWORD PTR [rbp-8], xmm0

        add     DWORD PTR [rbp-28], 1

.L2:

        mov     eax, DWORD PTR [rbp-28]

        cmp     eax, DWORD PTR [rbp-44]

        jle     .L3

        movsd   xmm0, QWORD PTR [rbp-8]

        movq    rax, xmm0

        movq    xmm0, rax

        pop     rbp

        ret

.LC1:

        .string "%f"

main:

        push    rbp

        mov     rbp, rsp

        sub     rsp, 16

        movsd   xmm0, QWORD PTR .LC0[rip]

        movsd   QWORD PTR [rbp-8], xmm0

        mov     DWORD PTR [rbp-12], 1000000

        mov     edx, DWORD PTR [rbp-12]

        mov     rax, QWORD PTR [rbp-8]

        mov     edi, edx

        movq    xmm0, rax

        call    exponent

        movq    rax, xmm0

        movq    xmm0, rax

        mov     edi, OFFSET FLAT:.LC1

        mov     eax, 1

        call    printf

        mov     eax, 0

        leave

        ret

.LC0:

        .long   0

        .long   1072693248

**Ассемблерный листинг для уровня оптимизации O2 (x86-64 gcc 13.2)**

exponent:

        movapd  xmm4, xmm0

        test    edi, edi

        jle     .L4

        movsd   xmm1, QWORD PTR .LC0[rip]

        add     edi, 1

        mov     eax, 1

        movapd  xmm2, xmm1

        movapd  xmm0, xmm1

.L3:

        mulsd   xmm2, xmm4

        pxor    xmm3, xmm3

        cvtsi2sd        xmm3, eax

        add     eax, 1

        mulsd   xmm1, xmm3

        movapd  xmm3, xmm2

        divsd   xmm3, xmm1

        addsd   xmm0, xmm3

        cmp     edi, eax

        jne     .L3

        ret

.L4:

        movsd   xmm0, QWORD PTR .LC0[rip]

        ret

.LC1:

        .string "%f"

main:

        sub     rsp, 8

        movsd   xmm3, QWORD PTR .LC0[rip]

        mov     eax, 1

        movapd  xmm1, xmm3

        movapd  xmm0, xmm3

.L8:

        pxor    xmm2, xmm2

        cvtsi2sd        xmm2, eax

        add     eax, 1

        mulsd   xmm0, xmm2

        movapd  xmm2, xmm3

        divsd   xmm2, xmm0

        addsd   xmm1, xmm2

        cmp     eax, 1000001

        jne     .L8

        movapd  xmm0, xmm1

        mov     edi, OFFSET FLAT:.LC1

        mov     eax, 1

        call    printf

        xor     eax, eax

        add     rsp, 8

        ret

.LC0:

        .long   0

        .long   1072693248

**Ассемблерный листинг для уровня оптимизации O0 (x86 gcc 13.2)**

exponent:

        push    ebp

        mov     ebp, esp

        sub     esp, 40

        mov     eax, DWORD PTR [ebp+8]

        mov     edx, DWORD PTR [ebp+12]

        mov     DWORD PTR [ebp-40], eax

        mov     DWORD PTR [ebp-36], edx

        fld1

        fstp    QWORD PTR [ebp-8]

        fld1

        fstp    QWORD PTR [ebp-16]

        fld1

        fstp    QWORD PTR [ebp-24]

        mov     DWORD PTR [ebp-28], 1

        jmp     .L2

.L3:

        fld     QWORD PTR [ebp-16]

        fmul    QWORD PTR [ebp-40]

        fstp    QWORD PTR [ebp-16]

        fild    DWORD PTR [ebp-28]

        fld     QWORD PTR [ebp-24]

        fmulp   st(1), st

        fstp    QWORD PTR [ebp-24]

        fld     QWORD PTR [ebp-16]

        fdiv    QWORD PTR [ebp-24]

        fld     QWORD PTR [ebp-8]

        faddp   st(1), st

        fstp    QWORD PTR [ebp-8]

        add     DWORD PTR [ebp-28], 1

.L2:

        mov     eax, DWORD PTR [ebp-28]

        cmp     eax, DWORD PTR [ebp+16]

        jle     .L3

        fld     QWORD PTR [ebp-8]

        leave

        ret

main:

        lea     ecx, [esp+4]

        and     esp, -8

        push    DWORD PTR [ecx-4]

        push    ebp

        mov     ebp, esp

        push    ecx

        sub     esp, 20

        fld1

        fstp    QWORD PTR [ebp-16]

        mov     DWORD PTR [ebp-20], 1000000

        sub     esp, 4

        push    DWORD PTR [ebp-20]

        push    DWORD PTR [ebp-12]

        push    DWORD PTR [ebp-16]

        call    exponent

        fstp    st(0)

        add     esp, 16

        mov     eax, 0

        mov     ecx, DWORD PTR [ebp-4]

        leave

        lea     esp, [ecx-4]

        ret

**Ассемблерный листинг для уровня оптимизации O2 (x86 gcc 13.2)**

exponent:

        sub     esp, 12

        mov     edx, DWORD PTR [esp+24]

        fld     QWORD PTR [esp+16]

        test    edx, edx

        jle     .L4

        fld1

        add     edx, 1

        mov     eax, 1

        fld     st(0)

        fld     st(1)

.L3:

        fmul    st, st(3)

        mov     DWORD PTR [esp+4], eax

        fild    DWORD PTR [esp+4]

        add     eax, 1

        fmulp   st(2), st

        fld     st(0)

        fdiv    st, st(2)

        faddp   st(3), st

        cmp     edx, eax

        jne     .L3

        fstp    st(0)

        fstp    st(0)

        fstp    st(1)

        add     esp, 12

        ret

.L4:

        fstp    st(0)

        fld1

        add     esp, 12

        ret

.LC2:

        .string "%f"

main:

        lea     ecx, [esp+4]

        fld1

        and     esp, -16

        mov     eax, 1

        push    DWORD PTR [ecx-4]

        fld     st(0)

        push    ebp

        mov     ebp, esp

        push    ecx

        sub     esp, 20

.L9:

        mov     DWORD PTR [ebp-12], eax

        fild    DWORD PTR [ebp-12]

        add     eax, 1

        fmulp   st(2), st

        fld     st(1)

        fdivr   DWORD PTR .LC0

        faddp   st(1), st

        cmp     eax, 1000001

        jne     .L9

        fstp    st(1)

        sub     esp, 12

        fstp    QWORD PTR [esp]

        push    OFFSET FLAT:.LC2

        call    printf

        mov     ecx, DWORD PTR [ebp-4]

        add     esp, 16

        xor     eax, eax

        leave

        lea     esp, [ecx-4]

        ret

.LC0:

        .long   1065353216

# 

# ЗАКЛЮЧЕНИЕ

Ознакомились с программной архитектурой x86/x86-64.

Были проанализированы ассемблерные листинги программы для архитектуры

x86-64 и их различия при использивании уровней оптимизации -O0 и -O2.