

Новосибирский Государственный Университет

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ

Курс “ЭВМ и периферийные устройства”

Лабораторная работа №3

«ВВЕДЕНИЕ В АРХИТЕКТУРУ x86/x86-64»

Выполнил:

Пятаев Егор, гр. 15206

Преподаватель:

Городничев Максим Александрович

Новосибирск 2016

Цели работы

1. Знакомство с программной архитектурой x86/x86-64.
2. Анализ ассемблерного листинга программы для архитектуры x86/x86-64.

Вариант задания

Алгоритм вычисления функции e^x с помощью разложения в ряд Маклорена по первым N членам этого ряда.

Листинг реализованной программы

header.h:

```
#ifndef H_1
#define H_1
#define BAD_ARGS 1
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

double calc_ex(double, double);

#endif
```

calc_ex.c:

```
#include "header.h"
double calc_ex(double n, double x) {

    double ex = 1;
    double i;
    double j = 1;

    /*Calculate ex*/
    if (x != 0) {
        for (i = 1; i < n; i++){
            j*=(x/i);
            ex+=j;
        }
    } else return ex;

    return ex;
}
```

main.c:

```
#include "header.h"
```

```
int main(int argc, char *argv[]) {  
    double ex;  
    double n;  
    double x;
```

```
if (argc == 3 ) {
```

```
    n = atof(argv[1]);  
    x = atof(argv[2]);
```

```
    if(n <= 0){  
        printf("Enter N > 0");  
        return BAD_ARGS;  
    }
```

```
    ex = calc_ex(n, x);
```

```
    printf("e^x = %.10f\n", ex);
```

```
} else printf("Bad arguments");
```

```
    return 0;
```

```
}
```

Команда компиляции: gcc [-O*] main.c calc.c -o calc [-lrt]

Команда запуска: ./calc [значение N] [значение X]

Ассемблерный листинг

Выводы

При оптимизации -Os в x86-64 уменьшен размер листингов, используются векторные расширения, некоторые встроенные функции преобразованы (printf в printf_chk), уменьшено количество обращений на регистры, добавлены метки типа .text.unlikely для предполагаемо редко вызываемых функций.

При оптимизации -OS в x86 также уменьшен размер листингов, уменьшено количество обращений на регистры, некоторые обращение к регистрам общего назначения заменено на обращение к стеку регистра сопроцессора,

x86 и x86-64 используют различные регистры, x86-64 использует векторные расширения, а x86 команды сопроцессора для проведения вычислений.