

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6

дисциплина: Архитектура компьютера

Студент: Мартынова Милана Александровна

Группа: НКАбд-04-25

МОСКВА

2025 г.

Содержание

1. Цель работы.....	4
2. Порядок выполнения лабораторной работы.....	4
2.1 Символьные и численные данные в NASM.....	4
2.2 Выполнение арифметических операций в NASM.....	5
2.3 Задание для самостоятельной работы.....	7
3. Выводы.....	7

Список иллюстраций

1. Создание нового каталога.....	4
2. Сохранение новой программы.....	4
3. Запуск изначальной программы.....	4
4. Измененная программа.....	4
5. Запуск измененной программы.....	5
6. Вторая программа.....	5
7. Вывод второй программы.....	5
8. Вывод измененной второй программы.....	5
9. Замена функции вывода во второй программе.....	5
10. Третья программа.....	6
11. Запуск третьей программы.....	6
12. Изменение третьей программы.....	6
13. Запуск измененной третьей программы.....	6
14. Программа для подсчета варианта.....	7
15. Запуск программы для подсчета варианта.....	7
16. Программа.....	7

1. Цель работы

Практическое освоение арифметических операций в NASM-ассемблере

2. Порядок выполнения лабораторной работы

2.1 Символьные и численные данные в NASM

Создаю каталог для программ лабораторной работы №6 и перехожу в него, создаю там файл (рис. 1).

```
mamartynova@VirtualBox:~$ mkdir ~/work/arch-pc/lab06
mamartynova@VirtualBox:~$ cd ~/work/arch-pc/lab06
mamartynova@VirtualBox:~/work/arch-pc/lab06$ touch lab6-1.asm
```

Рис. 1: Создание нового каталога

В созданном файле ввожу программу из листинга (рис. 2).

```
%include 'in_out.asm'

SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:

    mov eax, '6'
    mov ebx, '4'
    add eax,ebx
    mov [buf1],eax
    mov eax,buf1
    call sprintLF
    call quit
```

Рис. 2: Сохранение новой программы

Создаю исполняемый файл и запускаю его, вывод программы отличается от предполагаемого изначально, так как коды символов в сумме дают символ j по таблице ASCII. {#fig:003 width=70%}

```
mamartynova@VirtualBox:~/work/arch-pc/lab06$ gedit lab6-1.asm
mamartynova@VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
mamartynova@VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
mamartynova@VirtualBox:~/work/arch-pc/lab06$ ./lab6-1
j
```

Рис. 3: Запуск изначальной программы

Изменяю текст изначальной программы, убрав кавычки (рис. 4).

```
%include 'in_out.asm'

SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:

    mov eax, 6
    mov ebx, 4
    add eax,ebx
    mov [buf1],eax
    mov eax,buf1
    call sprintLF
    call quit
```

Рис. 4: Измененная программа

На этот раз программа выдала пустую строчку, это связано с тем, что символ 10 означает переход на новую строку (рис. 5).

```
mamartynova@VirtualBox:~/work/arch-pc/lab06$ gedit lab6-1.asm
mamartynova@VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
mamartynova@VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
mamartynova@VirtualBox:~/work/arch-pc/lab06$ ./lab6-1
```

```
mamartynova@VirtualBox:~/work/arch-pc/lab06$ █
```

Рис. 5: Запуск измененной программы

Создаю новый файл для будущей программы и записываю в нее код из листинга (рис. 6).

```
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
    mov eax, '6'
    mov ebx, '4'
    add eax, ebx
    call iprintLF
    call quit
```

Рис. 6: Вторая программа

Создаю исполняемый файл и запускаю его, теперь отображается результат 106, программа, как и в первый раз, сложила коды символов, но вывела само число, а не его символ, благодаря замене функции вывода на iprintLF (рис. 7).

```
mamartynova@VirtualBox:~/work/arch-pc/lab06$ touch lab6-2.asm
mamartynova@VirtualBox:~/work/arch-pc/lab06$ gedit lab6-2.asm
mamartynova@VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
mamartynova@VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
mamartynova@VirtualBox:~/work/arch-pc/lab06$ ./lab6-2
106
mamartynova@VirtualBox:~/work/arch-pc/lab06$ █
```

Рис. 7: Вывод второй программы

Убрав кавычки в программе, я снова ее запускаю и получаю предполагаемый изначально результат. (рис. 8).

```
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
    mov eax, 6
    mov ebx, 4
    add eax, ebx
    call iprintLF
    call quit
```

```
mamartynova@VirtualBox:~/work/arch-pc/lab06$ gedit lab6-2.asm
mamartynova@VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
mamartynova@VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
mamartynova@VirtualBox:~/work/arch-pc/lab06$ ./lab6-2
10
mamartynova@VirtualBox:~/work/arch-pc/lab06$ █
```

Рис. 8: Вывод измененной второй программы

Заменив функцию вывода на iprint, я получаю тот же результат, но без переноса строки (рис. 9).

```
mamartynova@VirtualBox:~/work/arch-pc/lab06$ gedit lab6-2.asm
mamartynova@VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
mamartynova@VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
mamartynova@VirtualBox:~/work/arch-pc/lab06$ ./lab6-2
10
mamartynova@VirtualBox:~/work/arch-pc/lab06$ █
```

Рис. 9: Замена функции вывода во второй программе

2.2 Выполнение арифметических операций в NASM

Создаю новый файл и копирую в него содержимое листинга (рис. 10).

```

1 %include 'in_out.asm'
2 SECTION .data
3 div: DB 'Результат: ', 0
4 rem: DB 'Остаток от деления: ', 0
5 SECTION .text
6 GLOBAL _start
7 _start:
8 mov eax, 5
9 mov ebx, 2
10 mul ebx
11 add eax, 3
12 xor edx, edx
13 mov ebx, 3
14 div ebx
15 mov edi, eax
16 mov eax, div
17 call sprint
18 mov eax, edi
19 call iprintLF
20 mov eax, rem
21 call sprint
22 mov eax, edx
23 call iprintLF
24 call quit

```

Рис. 10: Третья программа

Программа выполняет арифметические вычисления, на вывод идет результирующее выражения и его остаток от деления (рис. 11).

```

mamartynova@VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
mamartynova@VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
mamartynova@VirtualBox:~/work/arch-pc/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1

```

Рис. 11: Запуск третьей программы

Заменив переменные в программе для выражения $f(x) = (4*6+2)/5$ (рис. 12).

```

1 %include 'in_out.asm'
2 SECTION .data
3 div: DB 'Результат: ', 0
4 rem: DB 'Остаток от деления: ', 0
5 SECTION .text
6 GLOBAL _start
7 _start:
8 mov eax, 4
9 mov ebx, 6
10 mul ebx
11 add eax, 2
12 xor edx, edx
13 mov ebx, 5
14 div ebx
15 mov edi, eax
16 mov eax, div
17 call sprint
18 mov eax, edi
19 call iprintLF
20 mov eax, rem
21 call sprint
22 mov eax, edx
23 call iprintLF
24 call quit

```

Рис. 12: Изменение третьей программы

Запуск программы дает корректный результат (рис. 13).

```

mamartynova@VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
mamartynova@VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
mamartynova@VirtualBox:~/work/arch-pc/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1

```

Рис. 13: Запуск измененной третьей программы

Создаю новый файл и помещаю текст из листинга (рис. 14).

```

1 %include 'in_out.asm'
2 SECTION .data
3 msg: DB 'Введите № студенческого билета: ', 0
4 rem: DB 'Ваш вариант: ', 0
5 SECTION .bss
6 x: RESB 80
7 SECTION .text
8 GLOBAL _start
9 _start:
10 mov eax, msg
11 call sprintLF
12 mov ecx, x
13 mov edx, 80
14 call sread
15 mov eax, x
16 call atoi
17 xor edx, edx
18 mov ebx, 20
19 div ebx
20 inc edx
21 mov eax, rem
22 call sprint
23 mov eax, edx
24 call iprintLF
25 call quit

```

Рис. 14: Программа для подсчета варианта

Запустив программу и указав свой номер студенческого билета, я получил свой вариант для дальнейшей работы. (рис. 15).

```

namartynova@VirtualBox:~/work/arch-pc/lab06$ gedit variant.asm
namartynova@VirtualBox:~/work/arch-pc/lab06$ nasm -f elf variant.asm
namartynova@VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
namartynova@VirtualBox:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1032253522
Ваш вариант: 3

```

Рис. 15: Запуск программы для подсчета варианта

2.3 Задание для самостоятельной работы

В соответствии с выбранным вариантом, я реализую программу для подсчета функции $f(x) = (2 + x)^2$, проверка на нескольких переменных показывает корректное выполнение программы (рис. 16).

```

1 %include 'in_out.asm'
2 SECTION .data
3 msg: DB 'Введите значение переменной x: ', 0
4 rem: DB 'Результат: ', 0
5 SECTION .bss
6 x: RESB 80
7 SECTION .text
8 GLOBAL _start
9 _start:
10 mov eax, msg
11 call sprint
12 mov ecx, x
13 mov edx, 80
14 call sread
15 mov eax, x
16 call atoi
17 add eax, 2
18 mul eax
19 mov edi, eax
20 mov eax, rem
21 call sprint
22 mov eax, edi
23 call iprint
24 call quit

```

Рис. 16: Программа

3. Выводы

В ходе лабораторной работы были освоены арифметические инструкции языка ассемблер NASM. Практически исследованы различия между символьными и численными данными. Освоены подпрограммы преобразования данных: `iprint`, `iprintLF` для вывода чисел и `atoi` для преобразования введённых символов.

Разработаны программы для вычисления арифметических выражений. Изучены особенности работы с операцией деления и расположение результатов в регистрах. Приобретены навыки отладки ассемблерных программ.