

Лабораторная работа №6. Арифметические операции в NASM.

Аджигалиева Амина Руслановна

Содержание

1	Цель работы	4
2	Порядок выполнения лабораторной работы	5
2.1	Символьные и численные данные в NASM	5
2.2	Выполнение арифметических операций в NASM	10
2.3	Ответы на вопросы:	16
2.4	Задание для самостоятельной работы	16
3	Выводы	19

Список иллюстраций

2.1	Создание lab6-1.asm	5
2.2	Заполняем файл	6
2.3	Запуск программы	6
2.4	Замена текста	7
2.5	Запуск программы	7
2.6	Файл lab6-2.asm	8
2.7	Ввод текста	8
2.8	Запуск программы	8
2.9	Замена текста	9
2.10	Запуск программы	9
2.11	Функция iprint	10
2.12	Запуск программы	10
2.13	Создание файла	10
2.14	Ввод программы	11
2.15	Запуск программы	12
2.16	Замена текста	13
2.17	Запуск программы	14
2.18	Файл variant.asm	14
2.19	Программа со студ.билетом	15
2.20	Запуск программы	15
2.21	Решение уравнения на ассамблере	17
2.22	Запуск программы	18
2.23	Запуск программы	18

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Порядок выполнения лабораторной работы

2.1 Символьные и численные данные в NASM

Создадим каталог для программ лабораторной работы № 6, перейдите в него и создайте файл lab6-1.asm: (рис. 2.1).

```
aminaadzhigalieva@fedora:~$ mkdir ~/work/arch-pc/lab06
aminaadzhigalieva@fedora:~$ cd ~/work/arch-pc/lab06
aminaadzhigalieva@fedora:~/work/arch-pc/lab06$ touch lab6-1.asm
aminaadzhigalieva@fedora:~/work/arch-pc/lab06$ mc
```

Рис. 2.1: Создание lab6-1.asm

Открываем файл в Midnight Commander и заполняем его в соответствии с листингом 6.1 (рис. 2.2).

```
%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. 2.2: Заполняем файл

Создаем исполняемый файл и запускаем его (рис. 2.3).

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

Рис. 2.3: Запуск программы

Изменим текст программы и вместо символов, запишем в регистры числа. Исправим текст программы (Листинг 6.1): заменим строки (рис. 2.4).

```
%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. 2.4: Замена текста

Создаем исполняемый файл и запускаем его (рис. 2.5).

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

Рис. 2.5: Запуск программы

Создаем файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 (рис. 2.6).

```
aminaadzhigalieva@fedora:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-2.asm
aminaadzhigalieva@fedora:~/work/arch-pc/lab06$ mc
```

Рис. 2.6: Файл lab6-2.asm

Введем в него текст программы из листинга 6.2 (рис. 2.7).

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

Рис. 2.7: Ввод текста

Создаем исполняемый файл и запускаем его (рис. 2.8).

```
aminaadzhigalieva@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
aminaadzhigalieva@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
aminaadzhigalieva@fedora:~/work/arch-pc/lab06$ ./lab6-2
106
aminaadzhigalieva@fedora:~/work/arch-pc/lab06$
```

Рис. 2.8: Запуск программы

Снова открываем файл для редактирования и убираем кавычки с числовых значений (рис. 2.9).


```

#include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

mov eax,6
mov ebx,4
add eax,ebx
call iprintLF

call quit

```

Рис. 2.9: Замена текста

Создаем исполняемый файл и запускаем его (рис. 2.10).

```

aminaadzhigalieva@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
aminaadzhigalieva@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
aminaadzhigalieva@fedora:~/work/arch-pc/lab06$ ./lab6-2
10

```

Рис. 2.10: Запуск программы

Заменим функцию iprintLF на iprint (рис. 2.11).

```
%include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

mov eax,6
mov ebx,4
add eax,ebx
call iprint

call quit
```

Рис. 2.11: Функция iprint

Создаем исполняемый файл и запускаем его (рис. 2.12).

```
aminaadzhigalieva@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
aminaadzhigalieva@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
aminaadzhigalieva@fedora:~/work/arch-pc/lab06$ ./lab6-2
10aminaadzhigalieva@fedora:~/work/arch-pc/lab06$
```

Рис. 2.12: Запуск программы

Функции iprintLF и iprint отличаются только тем, что LF переносит на новую строку.

2.2 Выполнение арифметических операций в NASM

Создаем файл lab6-3.asm в каталоге ~/work/arch-pc/lab06: (рис. 2.13).

```
aminaadzhigalieva@fedora:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-3.asm
aminaadzhigalieva@fedora:~/work/arch-pc/lab06$
```

Рис. 2.13: Создание файла

Изучим текст программы из листинга 6.3 и введем в lab6-3.asm (рис. 2.14).

/home/aminaadzhigalieva/work/a

```
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0

SECTION .text
GLOBAL _start
_start:

mov eax,5
mov ebx,2
mul ebx
add eax,3
xor edx,edx
mov ebx,3
div ebx

mov edi,eax

mov eax,div
call sprint
mov eax,edi
call iprintLF

mov eax,rem
call sprint
mov eax,edx
call iprintLF

call quit
```

Рис. 2.14: Ввод программы

Создаем исполняемый файл и запускаем его (рис. 2.15).

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

Рис. 2.15: Запуск программы

Изменим текст программы для вычисления выражения $y = (4 * 6 + 2)/5$ (рис. 2.16).

```
%include 'in_out.asm'

SECTION .data

div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0

SECTION .text
GLOBAL _start
_start:

mov eax,4
mov ebx,6
mul ebx
add eax,2
xor edx,edx
mov ebx,5
div ebx

mov edi,eax

mov eax,div
call sprint
mov eax,edi
call iprintLF

mov eax,rem
call sprint
mov eax,edx
call iprintLF
```

Рис. 2.16: Замена текста

Создаем исполняемый файл и запускаем его (рис. 2.17).

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

Рис. 2.17: Запуск программы

Создаем файл `variant.asm` в каталоге `~/work/arch-pc/lab06` (рис. 2.18).

```
aminaadzhigalieva@fedora:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/variant.asm
aminaadzhigalieva@fedora:~/work/arch-pc/lab06$
```

Рис. 2.18: Файл `variant.asm`

Изучим текст программы из листинга 6.4 и введем в файл `variant.asm` (рис. 2.19).

```

#include 'in_out.asm'

SECTION .data
msg: DB 'Введите № студенческого билета: ',0
rem: DB 'Ваш вариант: ',0

SECTION .bss
x: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax, msg
call sprintf

mov ecx, x
mov edx, 80
call sread

mov eax, x
call atoi

xor edx, edx
mov ebx, 20
div ebx
inc edx

mov eax, rem
call sprintf
mov eax, edx
call iprintLF

call quit

```

Рис. 2.19: Программа со студ.билетом

Создаем исполняемый файл и запускаем его (рис. 2.20).

```

aminaadzhigaliev@fedora:~/work/arch-pc/lab06$ nasm -f elf variant.asm
aminaadzhigaliev@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
aminaadzhigaliev@fedora:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132243023
Ваш вариант: 4
aminaadzhigaliev@fedora:~/work/arch-pc/lab06$

```

Рис. 2.20: Запуск программы

2.3 Ответы на вопросы:

1. Строка `"mov eax,mem"` и строка `"call sprint"` отвечают за вывод на экран сообщения 'Ваш вариант:'.
2. Эти инструкции используются для чтения строки с вводом данных от пользователя. Начальный адрес строки сохраняется в регистре `ecx`, а количество символов в строке (максимальное количество символов, которое может быть считано) сохраняется в регистре `edx`. Затем вызывается процедура `sread`, которая выполняет чтение строки.
3. Инструкция `"call atoi"` используется для преобразования строки в целое число. Она принимает адрес строки в регистре `eax` и возвращает полученное число в регистре `eax`.
4. Строка `"xor edx,edx"` обнуляет регистр `edx` перед выполнением деления. Строка `"mov ebx,20"` загружает значение 20 в регистр `ebx`. Строка `"div ebx"` выполняет деление регистра `eax` на значение регистра `ebx` с сохранением частного в регистре `eax` и остатка в регистре `edx`.
5. Остаток от деления записывается в регистр `edx`.
6. Инструкция `"inc edx"` используется для увеличения значения в регистре `edx` на 1. В данном случае, она увеличивает остаток от деления на 1.
7. Строка `"mov eax,edx"` передает значение остатка от деления в регистр `eax`. Строка `"call iprintLF"` вызывает процедуру `iprintLF` для вывода значения на экран вместе с переводом строки.

2.4 Задание для самостоятельной работы

Открываем файл и заполняем, чтобы решалось выражение $y = 4/3(x - 1) + 5$ (рис. 2.21).


```

%include 'in_out.asm'

SECTION .data
msg: DB 'Введите x: ', 0
div: DB 'Результат:', 0

SECTION .bss
rez: RESB 4
x: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax, msg
call sprintLF

mov ecx, x
mov edx, 80
call sread
mov eax, x
call atoi

sub eax, 1
imul eax, 4
cdq
mov ebx, 3
idiv ebx
add eax, 5
mov [rez],eax

mov eax, div
call sprint
mov eax, [rez]
call iprintLF
call quit

```

Рис. 2.21: Решение уравнения на ассамблере

Создаем исполняемый файл и запускаем его для $x = 4$ (рис. 2.22).

```
aminaadzhigalieva@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-4.asm
aminaadzhigalieva@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-4 lab6-4.o
aminaadzhigalieva@fedora:~/work/arch-pc/lab06$ ./lab6-4
Введите x:
4
Результат:9
aminaadzhigalieva@fedora:~/work/arch-pc/lab06$
```

Рис. 2.22: Запуск программы

Теперь для $x = 10$ (рис. 2.23).

```
aminaadzhigalieva@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-4.asm
aminaadzhigalieva@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-4 lab6-4.o
aminaadzhigalieva@fedora:~/work/arch-pc/lab06$ ./lab6-4
Введите x:
10
Результат:17
aminaadzhigalieva@fedora:~/work/arch-pc/lab06$
```

Рис. 2.23: Запуск программы

3 Выводы

Мы приобрели навыки создания исполнительных файлов для решения выражений и освоили арифметические инструкции в NASM.