

Отчёт по лабораторной работе №6

Арифметические операции в NASM

Скобеева Алиса Алексеевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Символьные и численные данные в NASM	6
2.2	Выполнение арифметических операций в NASM	9
2.3	Ответы на вопросы по программе	12
2.4	Задание для самостоятельной работы	13
3	Выводы	15

Список иллюстраций

2.1	Используем команды mkdir и touch	6
2.2	Вводим текст	6
2.3	Запускаем файл и смотрим на его работу	7
2.4	Редактируем файл	7
2.5	Запускаем файл и смотрим на его работу	7
2.6	Создаем файл	7
2.7	Заполняем файл	8
2.8	Смотрим на работу программы	8
2.9	Редактируем файл	8
2.10	Смотрим, как сработала программа	9
2.11	Редактируем файл	9
2.12	Смотрим, как сработала программа	9
2.13	Создание файла	9
2.14	Редактируем файл	10
2.15	Смотрим на результат работы программы	10
2.16	Редактируем файл	11
2.17	Смотрим на результат работы программы	11
2.18	Редактируем файл	12
2.19	Проверяем результат работы программы	12
2.20	Редактируем файл	14
2.21	Проверяем результат работы программы	14
2.22	Проверяем результат работы программы	14

Список таблиц

1 Цель работы

Освоить арифметические инструкции языка ассемблера NASM и написать программы для вычисления арифметических выражений с неизвестной.

2 Выполнение лабораторной работы

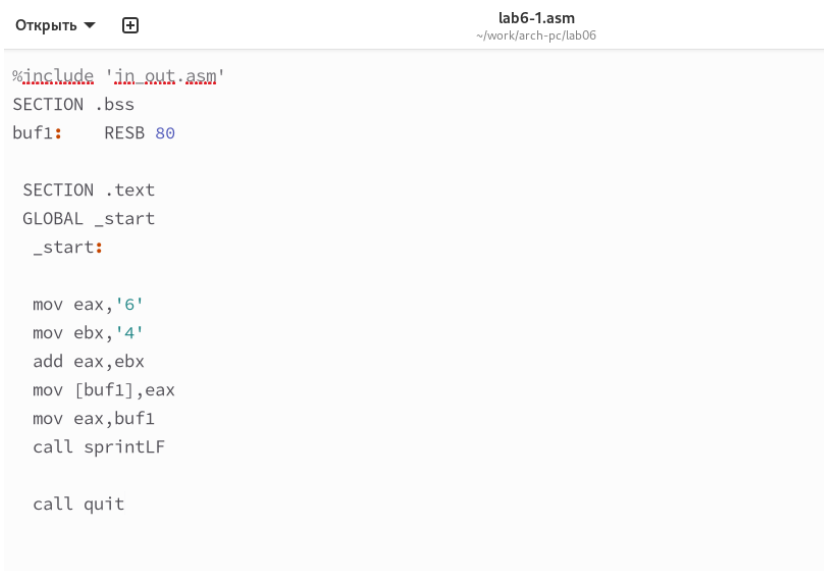
2.1 Символьные и численные данные в NASM

Создаем каталог для программ лабораторной работы №6, переходим в него и создаем файл lab6-1.asm:

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

Рис. 2.1: Используем команды mkdir и touch

Вводим в файл текст программы из листинга 6.1:



```
lab6-1.asm
~/work/arch-pc/lab06

%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: Вводим текст

Создаем исполняемый файл и запускаем его:

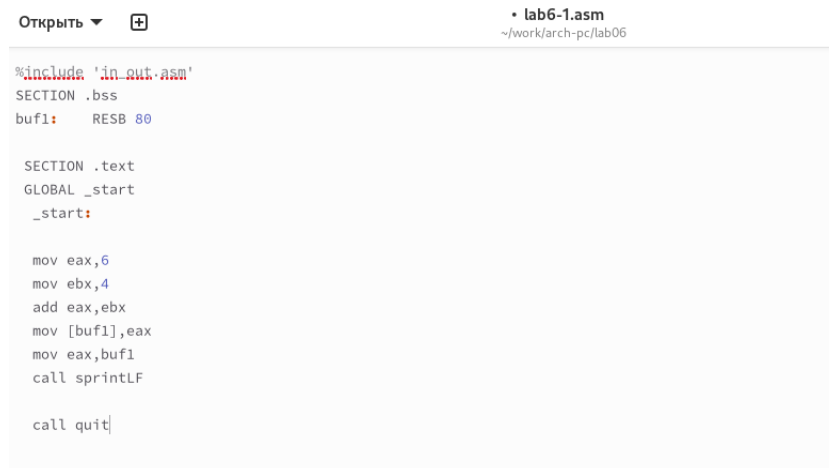
```

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

```

Рис. 2.3: Запускаем файл и смотрим на его работу

Открываем файл для редактирования и убираем кавычки с числовых значений:



```

Открыть ▾ + lab6-1.asm
~/work/arch-pc/lab06

%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: Редактируем файл

Создаем исполняемый файл и запускаем его:

```

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

```

Рис. 2.5: Запускаем файл и смотрим на его работу

Создаем новый файл в каталоге:

```

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

```

Рис. 2.6: Создаем файл

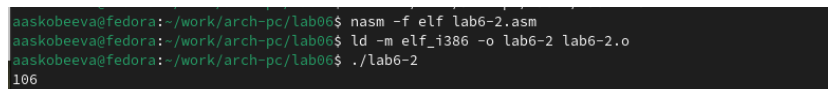
Заполняем файл в соответствии с листингом 6.2:



```
Открыть ▾  • lab6-2.asm  
~/work/arch-pc/lab06  
%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: Заполняем файл

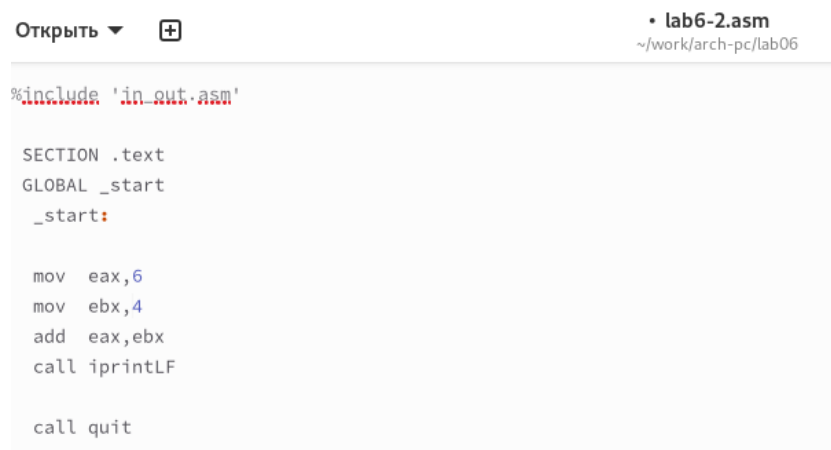
Создаем исполняемый файл и запускаем его:



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

Рис. 2.8: Смотрим на работу программы

Снова открываем файл для редактирования и убираем кавычки с числовых значений:




```
Открыть ▾  • lab6-2.asm  
~/work/arch-pc/lab06  
%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: Редактируем файл

Создаем исполняемый файл и запускаем его:


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

Рис. 2.10: Смотрим, как сработала программа

Снова открываем файл для редактирования и меняем `iprintLF` на `iprint`:



```
Открыть ▾ + lab6-2.asm
~/work/arch-pc/lab06

%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: Редактируем файл

Создаем исполняемый файл и запускаем его:

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

Рис. 2.12: Смотрим, как сработала программа

Таким образом, можем сделать вывод, что вывод функций `iprintLF` и `iprint` отличаются только тем, что `LF` переносит на новую строку.

2.2 Выполнение арифметических операций в NASM

Создаем новый файл в каталоге:

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

Рис. 2.13: Создание файла

Открываем файл и редактируем его в соответствии с листингом 6.3:

```
%include 'in_out.asm'
SECTION .data

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: Редактируем файл

Создаем исполняемый файл и запускаем его

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

Рис. 2.15: Смотрим на результат работы программы

Открываем файл и редактируем его для вычисления выражения $f(x) = (4*6^2)/5$:

```

%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

call quit

```

Рис. 2.16: Редактируем файл

Компилируем файл и запускаем программу:

```

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

```

Рис. 2.17: Смотрим на результат работы программы

Создаем новый файл в каталоге, открываем его и редактируем его в соответ-

ствии с листингом 6.4:

```
%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. 2.18: Редактируем файл

Компилируем файл и запускаем его:

```
aaskobeeva@fedora:~/work/arch-pc/lab06$ nasm -f elf variant.asm
aaskobeeva@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
aaskobeeva@fedora:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132246836
Ваш вариант: 17
```

Рис. 2.19: Проверяем результат работы программы

2.3 Ответы на вопросы по программе

1. Строка “mov eax,rem” и строка “call sprintf” отвечают за вывод на экран сообщения ‘Ваш вариант:’.
2. Данные инструкции используются для чтения строки с вводом данных от пользователя. Начальный адрес строки сохраняется в регистре ecx, а количество символов в строке (макс. кол-во 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 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

Создаем новый файл в каталоге, открываем его и заполняем, чтобы решалось выражение $f(x) = x^3/3 + 21$:

```

%include 'in_out.asm'
SECTION .data
msg: DB 'Введите x: ',0
div: DB 'Результат: ',0
SECTION .bss
rez: RESB 80
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

mov ebx,eax
mul eax
mul ebx
xor ebx,ebx
mov ebx,3
div ebx
xor ebx,ebx
add eax,21
mov [rez],eax

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

```

Рис. 2.20: Редактируем файл

Компилируем программу и проверяем для x=1:

```

aaskobeeva@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-4.asm
aaskobeeva@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-4 lab6-4.o
aaskobeeva@fedora:~/work/arch-pc/lab06$ ./lab6-4
Введите x:
1
Результат: 21

```

Рис. 2.21: Проверяем результат работы программы

Компилируем программу и проверяем для x=3:

```

aaskobeeva@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-4.asm
aaskobeeva@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-4 lab6-4.o
aaskobeeva@fedora:~/work/arch-pc/lab06$ ./lab6-4
Введите x:
3
Результат: 30

```

Рис. 2.22: Проверяем результат работы программы

3 Выводы

Я приобрела навыки создания исполняемых файлов для решения выражений и освоила арифметические инструкции в NASM.