

Лабораторная работа №7

Арифметические операции в NASM.

Лушин Артем Андреевич

Содержание

1	Цель работы	3
2	Выполнение лабораторной работы	4
3	Вопросы:	14
4	Самостоятельная работа	15
5	Вывод	18

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Выполнение лабораторной работы

- 1) Я создал каталог lab07 и внутри создал файл lab7-1.asm.

```
[aalushin@fedora ~]$ mkdir work/study/2022-2023/arch-pc/lab07
[aalushin@fedora ~]$ cd work/study/2022-2023/arch-pc/lab07
[aalushin@fedora lab07]$ pwd
/home/aalushin/work/study/2022-2023/arch-pc/lab07
[aalushin@fedora lab07]$ ls
[aalushin@fedora lab07]$ touch lab7-1.asm
[aalushin@fedora lab07]$ ls
lab7-1.asm
[aalushin@fedora lab07]$
```

Рис. 2.1: Создание файла lab7-1.asm

- 2) В файл lab7-1.asm я ввел нужный текст, создал файл и проверил его.
Программа вывела j.

```

%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 sprintfLF

call quit

```

Рис. 2.2: Текст в файле lab7-1.asm

```

[aalushin@fedora lab07]$ nasm -f elf lab7-1.asm
[aalushin@fedora lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[aalushin@fedora lab07]$ ./lab7-1
j
[aalushin@fedora lab07]$

```

Рис. 2.3: Создание файла и проверка работы

- 3) Я изменил текст программы. Когда я сделал программу и запустил ее, она вывела невидимый символ, как и должна была.

```

#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 sprintf

call quit

```

Рис. 2.4: Изменения программы lab7-1.asm

```

[aalushin@fedora lab07]$ nasm -f elf lab7-1.asm
[aalushin@fedora lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[aalushin@fedora lab07]$ ./lab7-1

[aalushin@fedora lab07]$ ./lab7-1

[aalushin@fedora lab07]$

```

Рис. 2.5: Проверка программы

4) Я создал файл lab7-2.asm и ввел в него текст программы.

```

lab7-2.asm [-----] 0 L
%include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

mov eax,'6'
mov ebx,'4'

add eax,ebx
call iprintLF
call quit

```

Рис. 2.6: Создание файла lab7-2.asm

- 5) Создал исполняемый файл и запустил его. Как и предполагалось, он вывел число 106.

```

[aalushin@fedora lab07]$ nasm -f elf lab7-2.asm
[aalushin@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[aalushin@fedora lab07]$ ./lab7-2
106
[aalushin@fedora lab07]$

```

Рис. 2.7: Запуск программы lab7-2.asm

- 6) Я изменил текст программы lab7-2 и запустил ее. Она вывела число 10, как и должна была.

```
lab7-2.asm [-----]
#include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

mov eax,6
mov ebx,4

add eax,ebx
call iprintLF
call quit
```

Рис. 2.8: Изменения в программе lab7-2.asm

```
[aalushin@fedora lab07]$ nasm -f elf lab7-2.asm
[aalushin@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[aalushin@fedora lab07]$ ./lab7-2
10
[aalushin@fedora lab07]$
```

Рис. 2.9: Проверка программы

7) Я заменил функцию `iprintLF` на `iprint`. Проверил файл и результат отличался в выводе данных. Когда используется команда `iprintLF` мы начинаем вводить команду на следующей строке, а при использовании команды `iprint` мы вводим следующие данные на той же строке.


```

lab7-2.asm [-----]
#include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

mov  eax,6
mov  ebx,4

add  eax,ebx
call iprint
call quit

```

Рис. 2.10: Замена iprintLF на iprint

```

[aalushin@fedora lab07]$ nasm -f elf lab7-2.asm
[aalushin@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[aalushin@fedora lab07]$ ./lab7-2
10[aalushin@fedora lab07]$ ./lab7-2
10[aalushin@fedora lab07]$

```

Рис. 2.11: Запуск программы с изменениями

- 8) Я создал файл lab7-3.asm и ввел текст программы. Запустил и проверил ее работу.

```

%include 'in_out.asm' ; подключение внешнего файла

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. 2.12: Текст в программе lab7-3.asm

```

[aalushin@fedora lab07]$ nasm -f elf lab7-3.asm
[aalushin@fedora lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
ld: невозможно найти lab7-3.0: Нет такого файла или каталога
[aalushin@fedora lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[aalushin@fedora lab07]$ ./lab7-3
Результат: 4
Остаток от деления: 1
[aalushin@fedora lab07]$

```

Рис. 2.13: Запуск программы lab7-3

9) Я изменил программу для примера $F(x) = (4*6+2)/5$. Запустил ее и проверил

```
%include 'in_out.asm' ; подключение внешнего файла

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. 2.14: Изменения в программе lab7-3

```
[aalushin@fedora lab07]$ nasm -f elf lab7-3.asm
[aalushin@fedora lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[aalushin@fedora lab07]$ ./lab7-3
Результат: 5
Остаток от деления: 1
[aalushin@fedora lab07]$
```

Рис. 2.15: Запуск программы

10) Я создал файл variant.asm и ввел туда нужный текст.

```

variant.asm      [-----]  9 L:[  1+33  34/ 34
%include 'in_out.asm'

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

SECTION .bss
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

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

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

call quit

```

Рис. 2.16: Текст программы в файле variant.asm

- 11) Я ввел в эту программу свой студенческий билет “1132226520” Он вывел ответ 1, я проверил это аналитически, ответ совпал.

```
[aalushin@fedora lab07]$ nasm -f elf variant.asm
[aalushin@fedora lab07]$ ld -m elf_i386 -o variant variant.o
[aalushin@fedora lab07]$ ./variant
Введите № студака:
1136520
Ваш вариант: 1
[aalushin@fedora lab07]$ ./variant
Введите № студака:
1132226520
Ваш вариант: 1
```

Рис. 2.17: Запуск программы variant

3 Вопросы:

1) Какие строки листинга 7.4 отвечают за вывод на экран сообщения 'Ваш вариант:'?

```
mov eax,msg call sprintLF
```

2) Для чего используются следующие инструкции? `asm mov ecx, x mov edx, 80 call sread`

Эти инструкции используются для ввода переменной X с клавиатуры и сохранения введенных данных.

3) Для чего используется инструкция "call atoi"?

Эта инструкция используется для преобразования Кода переменной ASCII в число.

4) Какие строки листинга 7.4 отвечают за вычисления варианта?

```
mov ebx,20 div ebx inc edx
```

5) В какой регистр записывается остаток от деления при выполнении инструкции "div ebx"?

В регистре ebx.

6) Для чего используется инструкция "inc edx"?

Для увеличения значения edx на 1.

7) Какие строки листинга 7.4 отвечают за вывод на экран результата вычислений?

```
mov eax,edx call iprintLF
```

4 Самостоятельная работа

Я написал программу которая будет решать выражение и выводить ответ при введенных x . Так как в 11 пункте мне выдало 1 вариант, то программу я писал для выражения $F(x) = (10+2x)/3$. Я вводил значения переменных 1 и 10 и ответы выдавало правильные.

```

/home/aalushin/work/study/2022-2023/arch-pc/lab07/1.asm
#include 'in_out.asm'

SECTION .data
prim: DB '(10+2x)/3',0
x1: DB 'Введите значение x: ',0
otv1: DB 'Ответ при x= ',0

SECTION .bss
p: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax,prim
call sprintLF

mov eax,x1
call sprint

mov ecx,p
mov edx,80
call sread

mov eax,p
call atoi

xor edx,edx

mov ebx,2
mul ebx
add eax,10
xor ebx,ebx
mov ebx,3
div ebx

mov edi,eax

mov eax,otv1
call sprint
mov eax,p
call sprint
mov eax,edi

```

Рис. 4.1: Программа для вычисления записанная в файле 1.asm


```
[aalushin@fedora lab07]$ nasm -f elf 1.asm
[aalushin@fedora lab07]$ ld -m elf_i386 -o 1 1.o
[aalushin@fedora lab07]$ ./1
(10+2x)/3
Введите значение x: 1
Ответ при x= 1
4
[aalushin@fedora lab07]$ ./1
(10+2x)/3
Введите значение x: 10
Ответ при x= 10
10
[aalushin@fedora lab07]$ █
```

Рис. 4.2: Результаты работы этой программы

5 Вывод

Я освоил арифметические инструкции языка ассемблер NASM.