

Лабораторная работа 7

Арифметические операции в NASM

Козлов Всеволод Павлович

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	24
	Список литературы	25

Список иллюстраций

2.1	Создание файла lab7-1.asm	6
2.2	Редактирование файла lab7-1.asm	7
2.3	Запуск исполняемого файла	7
2.4	Исправление текста программы	8
2.5	Запуск исполняемого файла	9
2.6	Создал файл lab7-2.asm	9
2.7	Редактирование файла lab7-2.asm	10
2.8	Запуск исполняемого файла	11
2.9	Редактирование текста программы	12
2.10	Запуск исполняемого файла	13
2.11	Редактирование текста программы	13
2.12	Запуск исполняемого файла	14
2.13	Создание файла lab7-3.asm	14
2.14	Ввод текста программы в lab7-3.asm	15
2.15	Запуск исполняемого файла	16
2.16	Редактирование текста программы	17
2.17	Запуск исполняемого файла	18
2.18	Ввод текста программы в lab7-3.asm	19
2.19	Запуск исполняемого файла	20
2.20	Создание файла	21
2.21	Ввод текста программы	22
2.22	Создание исполняемого файла	23
2.23	Проверка работоспособности программы	23
2.24	Проверка работоспособности программы	23

Список таблиц

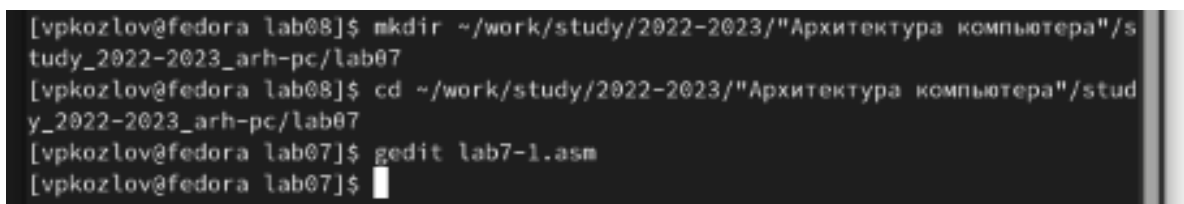
1 Цель работы

Цель лабораторной работы: освоение арифметических инструкций языка ассемблера NASM.

2 Выполнение лабораторной работы

Ход выполнения лабораторной работы:

- 1) Создал каталог для программ лабораторной работы № 7, перешел в него и создал файл lab7-1.asm (рис. 2.1)



```
[vpkozlov@fedora lab08]$ mkdir ~/work/study/2022-2023/"Архитектура компьютера"/study_2022-2023_arh-pc/lab07
[vpkozlov@fedora lab08]$ cd ~/work/study/2022-2023/"Архитектура компьютера"/study_2022-2023_arh-pc/lab07
[vpkozlov@fedora lab07]$ gedit lab7-1.asm
[vpkozlov@fedora lab07]$
```

Рис. 2.1: Создание файла lab7-1.asm

- 2) Ввел в файл lab7-1.asm текст программы из листинга (рис. 2.2)

```

1 %include 'in_out.asm'
2 SECTION .bss
3 buf1: RESB 80
4 SECTION .text
5 GLOBAL _start
6 _start:
7 mov eax, '6'
8 mov ebx, '4'
9 add eax, ebx
10 mov [buf1], eax
11 mov eax, buf1
12 call sprintLF
13 call quit

```

Рис. 2.2: Редактирование файла lab7-1.asm

Создал исполняемый файл и запустил его (рис. 2.3)

```

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

```

Рис. 2.3: Запуск исполняемого файла

3) Изменил некоторые строки программы (рис. 2.4)

```
1 %include 'in_out.asm'
2 SECTION .bss
3 buf1: RESB 80
4 SECTION .text
5 GLOBAL _start
6 _start:
7 mov eax,6
8 mov ebx,4
9 add eax,ebx
10 mov [buf1],eax
11 mov eax,buf1
12 call sprintfLF
13 call quit
```

Рис. 2.4: Исправление текста программы

Создал исполняемый файл для отредактированной программы и запустил его (рис. 2.5)

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

[vpkozlov@fedora lab07]$
```

Рис. 2.5: Запуск исполняемого файла

(программа вывела символ “enter”, чьему коду соответствует число 10 в таблице ASCII)

4) Создал файл lab7-2.asm в каталоге ~/work/arch-pc/lab07 (рис. 2.6)

```
[vpkozlov@fedora lab07]$ touch lab7-2.asm
[vpkozlov@fedora lab07]$
```

Рис. 2.6: Создал файл lab7-2.asm

Ввел текст программы в lab7-2.asm (рис. 2.7)

```
1 %include 'in_out.asm'
2 SECTION .text
3 GLOBAL _start
4 _start:
5 mov eax, '6'
6 mov ebx, '4'
7 add eax, ebx
8 call iprintLF
9 call quit
10 |
```

Рис. 2.7: Редактирование файла lab7-2.asm

Создал исполняемый файл и запустил его (рис. 2.8)

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

Рис. 2.8: Запуск исполняемого файла

5) Изменил некоторые строки программы lab7-2.asm (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: Редактирование текста программы

Создал исполняемый файл и запустил его (в результате выполнения программы получил число 10) (рис. 2.10)

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

Рис. 2.10: Запуск исполняемого файла

Заменяю 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: Редактирование текста программы

Создал исполняемый файл и запустил его (в отличие от `iprintLF` функция `iprint` не переводит курсор на последующую строку) (рис. 2.12)

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

Рис. 2.12: Запуск исполняемого файла

6) Создал файл lab7-3.asm в каталоге ~/work/arch-рс/lab07 (рис. 2.13)

```
10[vpkozlov@fedora lab07]touch lab7-3.asm
[vpkozlov@fedora lab07]$
```

Рис. 2.13: Создание файла lab7-3.asm

Ввел текст программы из листинга в lab7-3.asm (рис. 2.14)

```

div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax,5 ; EAX=5
mov ebx,2 ; EBX=2
mul ebx ; EAX=EAX*EBX
add eax,3 ; EAX=EAX+3
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,3 ; EBX=3
div ebx ; EAX=EAX/3, EDX=остаток от деления
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax,div ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов
mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Остаток от деления: '
mov eax,edx ; вызов подпрограммы печати значения
call iprintLF ; из 'edx' (остаток) в виде символов
call quit ; вызов подпрограммы завершения

```

[vpkozlov@fedora lab07]\$

Рис. 2.14: Ввод текста программы в lab7-3.asm

Создал исполняемый файл и запустил его (рис. 2.15)

```
[vpkozlov@fedora lab07]$ nasm -f elf lab7-3.asm
[vpkozlov@fedora lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[vpkozlov@fedora lab07]$ ./lab7-3
Результат: 4
Остаток от деления: 1
[vpkozlov@fedora lab07]$
```

Рис. 2.15: Запуск исполняемого файла

Изменил текст программы файла lab7-3.asm (рис. 2.16)


```

#include 'in_out.asm' ; подключение внешнего файла
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax,4 ; EAX=4
mov ebx,6 ; EBX=6
mul ebx ; EAX=EAX*EBX
add eax,2 ; EAX=EAX+2
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,5 ; EBX=5
div ebx ; EAX=EAX/5, EDX=остаток от деления
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax,div ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов
mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Остаток от деления: '

```

[vorkozlov@fedora lab87]

Рис. 2.16: Редактирование текста программы

Создал исполняемый файл и запустил его (рис. 2.17)

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

Рис. 2.17: Запуск исполняемого файла

Ввел текст программы из листинга в lab7-3.asm (рис. 2.18)

```

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 ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, `eax=x`
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: Ввод текста программы в lab7-3.asm

Создал исполняемый файл и запустил его (рис. 2.19)

```
[vpkozlov@fedora lab07]$ nasm -f elf lab7-3.asm
[vpkozlov@fedora lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[vpkozlov@fedora lab07]$ ./lab7-3
Введите № студенческого билета:
1132226428
Ваш вариант: 9
[vpkozlov@fedora lab07]$
```

Рис. 2.19: Запуск исполняемого файла

Ответы на вопросы:

Ответ на вопрос 1: строки { rem: DB 'Ваш вариант:',0 ; mov eax,rem ; call sprint } отвечают за вывод на экран сообщения 'Ваш вариант:'.

Ответ на вопрос 2: nasm транслирует текст программы в машинный код, называемый объектным; mov ecx, x – запись адреса переменной x в 'ECX'; mov edx, 80 - запись длины вводимого сообщения в EDX; call sread - вызов подпрограммы ввода сообщения.

Ответ на вопрос 3: atoi – функция преобразует ascii-код символа в целое число и запишет результат в регистр eax, перед вызовом atoi в регистр eax необходимо записать число (mov eax,).

Ответ на вопрос 4: строки { xor edx,edx ; mov ebx,20 ; div ebx ; inc edx} отвечают за вычисления варианта.

Ответ на вопрос 5: остаток от деления при выполнении инструкции "div ebx" записывается в регистр edx.

Ответ на вопрос 6: команда inc edx увеличивает значение регистра edx на 1.

Ответ на вопрос 7: строки { mov eax,edx ; call iprintLF} отвечают за вывод на экран результата вычислений.

Задание для самостоятельной работы (мой вариант – 9):

Создание файла для самостоятельной работы (рис. 2.20)

```
[vpkozlov@fedora lab07]$ touch lab7-samost.asm  
[vpkozlov@fedora lab07]$
```

Рис. 2.20: Создание файла

Ввел текст программы в файл (рис. 2.21)

```

%include 'in_out.asm'
SECTION .data
div: DB 'Выражение вида 10+(31+x-5). '
msg: DB 'Введите x: ',0
rem: DB 'Результат: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:

mov eax, div
call sprint

mov ecx, x
mov edx, 80
call sread

mov eax, x
call atoi

mov ecx, eax

```

Рис. 2.21: Ввод текста программы

Создал исполняемый файл (рис. 2.22)

```
[vpkozlov@fedora lab07]$ nasm -f elf lab7-samost.asm  
[vpkozlov@fedora lab07]$ ld -m elf_i386 -o lab7-samost lab7-samost.o
```

Рис. 2.22: Создание исполняемого файла

Проверил работу программы для $x_1 = 3$ (рис. 2.23)

```
[vpkozlov@fedora lab07]$ ./lab7-samost  
Выражение вида  $10+(31*x-5)$ . Введите X: 3  
Результат: 98  
[vpkozlov@fedora lab07]$
```

Рис. 2.23: Проверка работоспособности программы

Проверил работу программы для $x_2 = 1$ (рис. 2.24)

```
[vpkozlov@fedora lab07]$ ./lab7-samost  
Выражение вида  $10+(31*x-5)$ . Введите X: 1  
Результат: 36  
[vpkozlov@fedora lab07]$
```

Рис. 2.24: Проверка работоспособности программы

3 Выводы

Вывод: освоил арифметических инструкций языка ассемблера NASM.

Список литературы