

lab-07

Арифметические операции в NASM

Владимир Андреевич Баранов

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	13

Список иллюстраций

2.1	Создание файла.	6
2.2	Копирование файла.	7
2.3	Исполнение программы.	7
2.4	Запуск измененной программы.	7
2.5	Исполнение программы.	8
2.6	Запуск измененной программы.	8
2.7	Создание файла и исполнение программы.	8
2.8	Исполнение программы.	9
2.9	Вычисление номера варианта.	9
2.10	Программа.	11
2.11	Вычисление.	12

Список таблиц

1 Цель работы

Целью работы является освоение арифметических инструкций языка ассемблера NASM.

2 Выполнение лабораторной работы

1. Создаю каталог для программ лабораторной работы 7, перехожу в него и создаю файл lab7-1.asm (рис. 2.1).

```
vabaranov@dk8n80 ~ $ mkdir ~/work/arch-pc/lab07
vabaranov@dk8n80 ~ $ cd ~/work/arch-pc/lab07
vabaranov@dk8n80 ~/work/arch-pc/lab07 $ touch lab7-1.asm
vabaranov@dk8n80 ~/work/arch-pc/lab07 $
```

Рис. 2.1: Создание файла.

2. Для корректной работы программы подключаемый файл in_out.asm копирую в каталог ~/work/arch-pc/lab07 (рис. 2.2).

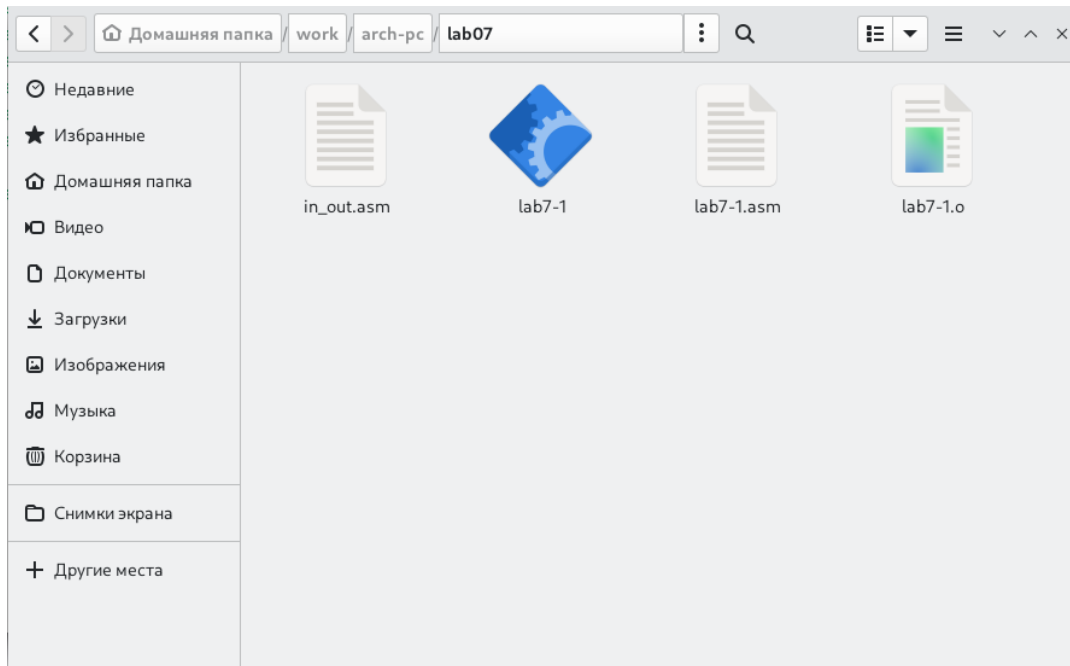


Рис. 2.2: Копирование файла.

3. Создаю исполняемый файл и запускаю его (рис. 2.3).

```
vabaranov@dk8n80 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
vabaranov@dk8n80 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
vabaranov@dk8n80 ~/work/arch-pc/lab07 $ ./lab7-1
j
vabaranov@dk8n80 ~/work/arch-pc/lab07 $
```

Рис. 2.3: Исполнение программы.

4. Изменяю текст программы из листинга 1, создаю исполняемый файл и запускаю его (рис. 2.4).

```
vabaranov@dk8n80 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
vabaranov@dk8n80 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
vabaranov@dk8n80 ~/work/arch-pc/lab07 $ ./lab7-1

vabaranov@dk8n80 ~/work/arch-pc/lab07 $
```

Рис. 2.4: Запуск измененной программы.

5. Создаю файл lab7-2.asm в каталоге ~/work/arch-pc/lab07 и запускаю его (рис. 2.5).

```
vabaranov@dk8n80 ~/work/arch-pc/lab07 $ nasm -f elf lab7-2.asm
vabaranov@dk8n80 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-2 lab7-2.o
vabaranov@dk8n80 ~/work/arch-pc/lab07 $ ./lab7-2
106
vabaranov@dk8n80 ~/work/arch-pc/lab07 $
```

Рис. 2.5: Исполнение программы.

6. Изменяю текст программы из листинга 2, создаю исполняемый файл и запускаю его (рис. 2.6).

```
vabaranov@dk8n80 ~/work/arch-pc/lab07 $ nasm -f elf lab7-2.asm
vabaranov@dk8n80 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-2 lab7-2.o
vabaranov@dk8n80 ~/work/arch-pc/lab07 $ ./lab7-2
10
vabaranov@dk8n80 ~/work/arch-pc/lab07 $
```

Рис. 2.6: Запуск измененной программы.

Разница между функциями в том, что `iprint` просто выводит сообщение на экран, а `iprintLF` добавляет к этому переход на новую строку.

7. Создаю файл lab7-3.asm в каталоге ~/work/arch-pc/lab07 и запускаю его (рис. 2.7).

```
vabaranov@dk8n80 ~/work/arch-pc/lab07 $ touch ~/work/arch-pc/lab07/lab7-3.asm
vabaranov@dk8n80 ~/work/arch-pc/lab07 $ nasm -f elf lab7-3.asm
vabaranov@dk8n80 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-3 lab7-3.o
vabaranov@dk8n80 ~/work/arch-pc/lab07 $ ./lab7-3
Результат: 4
Остаток от деления: 1
vabaranov@dk8n80 ~/work/arch-pc/lab07 $
```

Рис. 2.7: Создание файла и исполнение программы.

8. Изменяю текст программы для вычисления выражения $\boxtimes(\boxtimes) = (4 \boxtimes 6 + 2)/5$, создаю исполняемый файл и запускаю его (рис. 2.8).


```

vabaranov@dk8n80 ~/work/arch-pc/lab07 $ nasm -f elf lab7-3.asm
vabaranov@dk8n80 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-3 lab7-3.o
vabaranov@dk8n80 ~/work/arch-pc/lab07 $ ./lab7-3
Результат: 5
Остаток от деления: 1
vabaranov@dk8n80 ~/work/arch-pc/lab07 $ █

```

Рис. 2.8: Исполнение программы.

9. Создаю файл variant.asm в каталоге ~/work/arch-pc/lab07 и ввожу в этот файл текст программы из листинга 7.4. Создаю исполняемый файл, запускаю его и вывожу номер варианта на экран (рис. 2.9).

```

vabaranov@dk8n80 ~/work/arch-pc/lab07 $ touch ~/work/arch-pc/lab07/variant.asm
vabaranov@dk8n80 ~/work/arch-pc/lab07 $ nasm -f elf variant.asm
vabaranov@dk8n80 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o variant variant.o
vabaranov@dk8n80 ~/work/arch-pc/lab07 $ ./variant
Введите No студенческого билета:
1132226516
Ваш вариант: 17
vabaranov@dk8n80 ~/work/arch-pc/lab07 $ █

```

Рис. 2.9: Вычисление номера варианта.

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? `nasm 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`.

Самостоятельная работа

В результате работы программы `variant.asm` получаю вариант №17. Создаю файл с именем `lab7-5.asm`, пишу там программу для вычисления значения выражения $18(x+1)/6$ (рис. 2.10).

```
lab7-5.asm
1 %include 'in_out.asm'
2
3 SECTION .data
4 a1: DB '18(x+1)/6',0
5 a2: DB 'Введите значение: ',0
6 otv: DB 'Ответ: ',0
7
8 SECTION .bss
9 x: RESB 80
10
11 SECTION .text
12
13 global _start
14 _start:
15     mov eax, a1
16     call sprintf
17
18     mov eax, a2
19     call sprintf
20
21     mov ecx, x
22     mov edx, 80
23     call sread
24
25     mov eax, x
26     call atoi
27
28     inc eax
29     mov ebx, 18
30     mul ebx
31     xor edx, edx
32     mov ebx, 6
33     div ebx
34
35
36     mov edi, eax
37
38     mov eax, otv
39     call sprintf
40     mov eax, edi
41     call iprintLF
42
43     call quit
```

Рис. 2.10: Программа.

Далее запускаю файл. Подставляю заданные значения x , сверяю полученные результаты с вычисленными аналитически (рис. 2.11).

```
vabaranov@dk2n22 ~/work/arch-pc/lab07 $ nasm -f elf lab7-5.asm
vabaranov@dk2n22 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-5 lab7-5.o
vabaranov@dk2n22 ~/work/arch-pc/lab07 $ ./lab7-5
18(x+1)/6
Введите значение:
3
Ответ: 12
vabaranov@dk2n22 ~/work/arch-pc/lab07 $ nasm -f elf lab7-5.asm
vabaranov@dk2n22 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-5 lab7-5.o
vabaranov@dk2n22 ~/work/arch-pc/lab07 $ ./lab7-5
18(x+1)/6
Введите значение:
1
Ответ: 6
vabaranov@dk2n22 ~/work/arch-pc/lab07 $
```

Рис. 2.11: Вычисление.

3 Выводы

В ходе данной лабораторной работы я освоил арифметические инструкции языка ассемблера NASM.