

Лабораторная работа №7

Архитектура компьютера

Матвей Эдуардович Косолапов

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	18
	Список литературы	19

Список иллюстраций

4.1	Создание каталога и файла lab7-1.asm	8
4.2	Программа №1.1	8
4.3	Создание исполняемого файла и результат выполнения программы №1.1	9
4.4	Программа №1.2	9
4.5	Создание исполняемого файла и результат выполнения программы №1.2	9
4.6	Программа №2.1	10
4.7	Создание исполняемого файла и результат выполнения программы №2.1	10
4.8	Программа №2.2	10
4.9	Создание исполняемого файла и результат выполнения программы №2.2	11
4.10	Программа №2.3	11
4.11	Создание исполняемого файла и результат выполнения программы №2.3	11
4.12	Создание файла lab7-3.asm	11
4.13	Программа №4.1 (вычисление значения функции)	12
4.14	Создание исполняемого файла и результат выполнения программы №4.1	12
4.15	Программа №4.2	13
4.16	Создание исполняемого файла и результат выполнения программы №4.2	13
4.17	Создание файла variant.asm	13
4.18	Программа variant	14
4.19	Создание исполняемого файла и результат выполнения программы variant	14
4.20	Создание файла myprogvar6.asm	15
4.21	Программа myprogvar6	16
4.22	Создание исполняемого файла и результат выполнения программы myprogvar6	17

Список таблиц

1 Цель работы

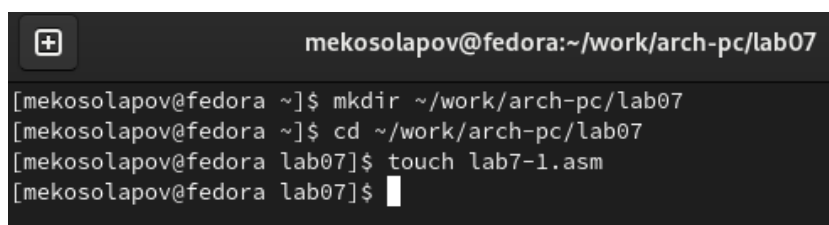
Освоение арифметических инструкций языка ассемблера NASM.

2 Задание

3 Теоретическое введение

4 Выполнение лабораторной работы

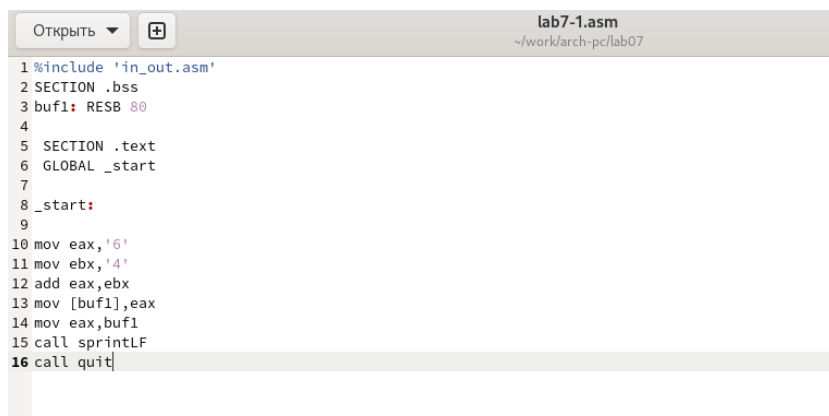
1. Создадим каталог для программ ~/work/arch-pc/lab07, переходим в него и создаём файл lab7-1.asm (рис. 4.1):



```
mekosolapov@fedora:~/work/arch-pc/lab07
[mekosolapov@fedora ~]$ mkdir ~/work/arch-pc/lab07
[mekosolapov@fedora ~]$ cd ~/work/arch-pc/lab07
[mekosolapov@fedora lab07]$ touch lab7-1.asm
[mekosolapov@fedora lab07]$
```

Рис. 4.1: Создание каталога и файла lab7-1.asm

2. Переносим в файл предложенную программу (рис. 4.2):



```
lab7-1.asm
~/work/arch-pc/lab07

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

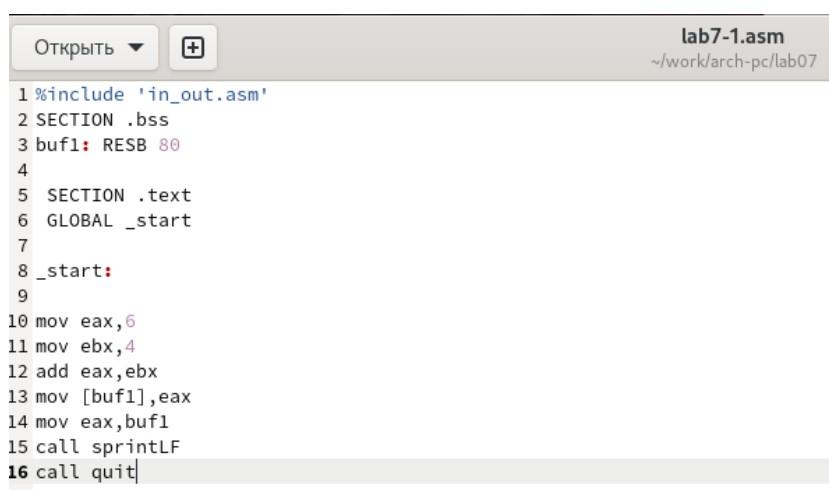
Рис. 4.2: Программа №1.1

3. Создаём и запускаем исполняемый файл (рис. 4.3):


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

Рис. 4.3: Создание исполняемого файла и результат выполнения программы №1.1

4. Изменяем программу, меняя '6','4' на 6,4 (рис. 4.4):



```
lab7-1.asm
~/work/arch-pc/lab07

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

Рис. 4.4: Программа №1.2

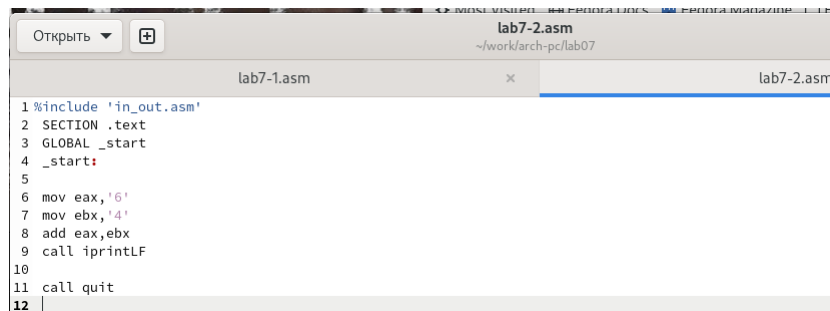
5. Создаём и запускаем исполняемый файл. Получаем символ пустоты (рис. 4.5):

```
[mekosolapov@fedora lab07]$ nasm -f elf lab7-1.asm
[mekosolapov@fedora lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
> nasm -f elf lab7-1.asm
ld: параметр -f нельзя использовать без -shared
[mekosolapov@fedora lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[mekosolapov@fedora lab07]$ ./lab7-1

[mekosolapov@fedora lab07]$
```

Рис. 4.5: Создание исполняемого файла и результат выполнения программы №1.2

6. Создаём файл lab7-2.asm и вводим в него текст предложенной программы (рис. 4.6):



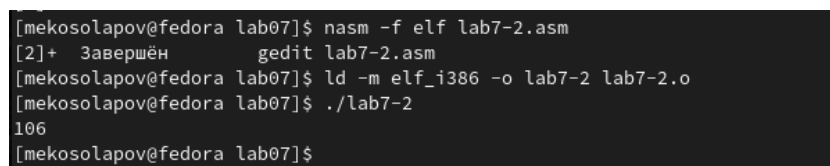
```
lab7-2.asm
~/work/arch-pc/lab07

lab7-1.asm x lab7-2.asm

1 %include 'in_out.asm'
2 SECTION .text
3 GLOBAL _start
4 _start:
5
6 mov eax, '6'
7 mov ebx, '4'
8 add eax, ebx
9 call iprintLF
10
11 call quit
12
```

Рис. 4.6: Программа №2.1

7. Создаём исполняемый файл и запускаем его (рис. 4.7):



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

Рис. 4.7: Создание исполняемого файла и результат выполнения программы №2.1

8. Изменяем программу, меняя '6','4' на 6,4 (рис. 4.8):



```
lab7-2.asm
~/work/arch-pc/la

Открыть ▾ +

1 %include 'in_out.asm'
2 SECTION .text
3 GLOBAL _start
4 _start:
5
6 mov eax,6
7 mov ebx,4
8 add eax,ebx
9 call iprint
10
11 call quit
12
```

Рис. 4.8: Программа №2.2

9. Создаём исполняемый файл и запускаем его. В результате выведено число 10 (рис. 4.9):

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

Рис. 4.9: Создание исполняемого файла и результат выполнения программы №2.2

10. Меняем функцию `iprintLF` на `iprint`, создаём исполняемый файл и проверяем его работу. Видим, что функция `iprintLF` добавляет в конце снос строки (рис. 4.10 - рис. 4.11):

```
1 %include 'in_out.asm'
2 SECTION .text
3 GLOBAL _start
4 _start:
5
6 mov eax,'6'
7 mov ebx,'4'
8 add eax,ebx
9 call iprint
10
11 call quit
12
```

Рис. 4.10: Программа №2.3

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

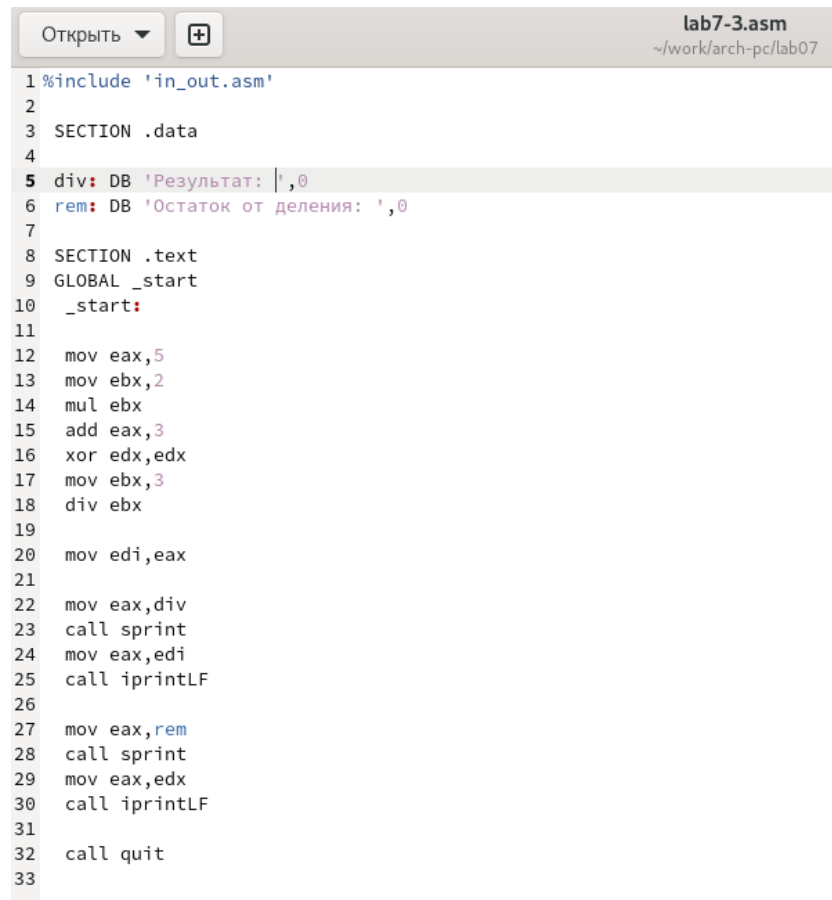
Рис. 4.11: Создание исполняемого файла и результат выполнения программы №2.3

11. Создаём файл в этом же каталоге файл `lab7-3.asm` (рис. 4.12):

```
[mekosolapov@fedora lab07]$ touch lab7-3.asm
[mekosolapov@fedora lab07]$ gedit lab7-3.asm &
```

Рис. 4.12: Создание файла `lab7-3.asm`

12. Переносим в файл программу вычисления значения функции (рис. 4.13):

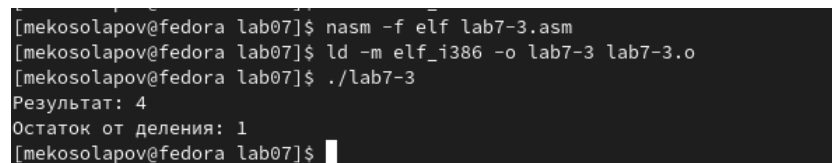


```
lab7-3.asm
~/work/arch-pc/lab07

1 %include 'in_out.asm'
2
3 SECTION .data
4
5 div: DB 'Результат: ',0
6 rem: DB 'Остаток от деления: ',0
7
8 SECTION .text
9 GLOBAL _start
10 _start:
11
12 mov eax,5
13 mov ebx,2
14 mul ebx
15 add eax,3
16 xor edx,edx
17 mov ebx,3
18 div ebx
19
20 mov edi,eax
21
22 mov eax,div
23 call sprint
24 mov eax,edi
25 call iprintLF
26
27 mov eax,rem
28 call sprint
29 mov eax,edx
30 call iprintLF
31
32 call quit
33
```

Рис. 4.13: Программа №4.1 (вычисление значения функции)

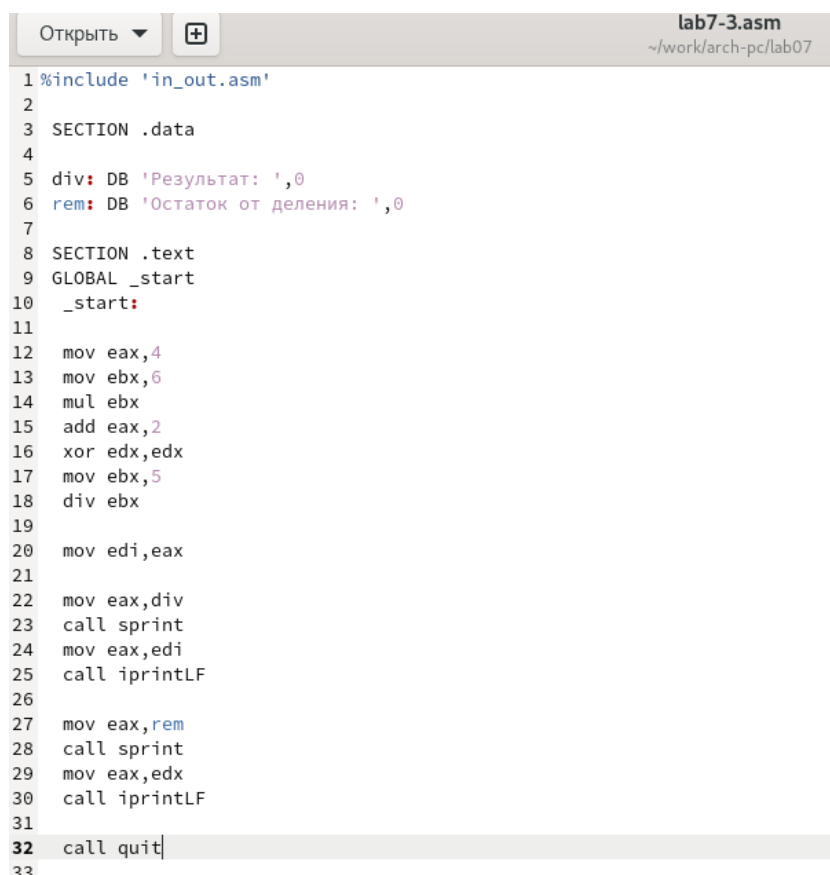
13. Создаём исполняемый файл и запускаем его (рис. 4.14):



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

Рис. 4.14: Создание исполняемого файла и результат выполнения программы №4.1

14. Меняем программу под новую функцию. Создаём исполняемый файл и запускаем его (рис. 4.15 - рис. 4.16):

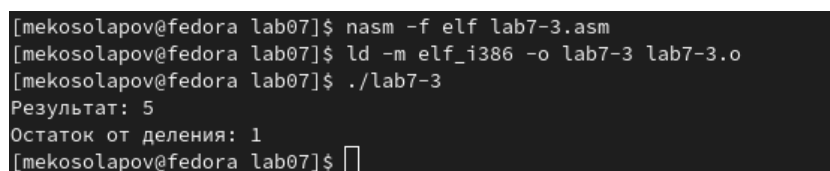


```

lab7-3.asm
~/work/arch-pc/lab07
1 %include 'in_out.asm'
2
3 SECTION .data
4
5 div: DB 'Результат: ',0
6 rem: DB 'Остаток от деления: ',0
7
8 SECTION .text
9 GLOBAL _start
10 _start:
11
12 mov eax,4
13 mov ebx,6
14 mul ebx
15 add eax,2
16 xor edx,edx
17 mov ebx,5
18 div ebx
19
20 mov edi,eax
21
22 mov eax,div
23 call sprint
24 mov eax,edi
25 call iprintLF
26
27 mov eax,rem
28 call sprint
29 mov eax,edx
30 call iprintLF
31
32 call quit

```

Рис. 4.15: Программа №4.2



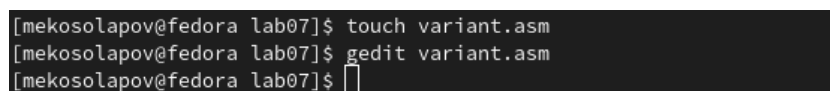
```

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

```

Рис. 4.16: Создание исполняемого файла и результат выполнения программы №4.2

15. Теперь создадим файл variant.asm для определения варианта задания (рис. 4.17):



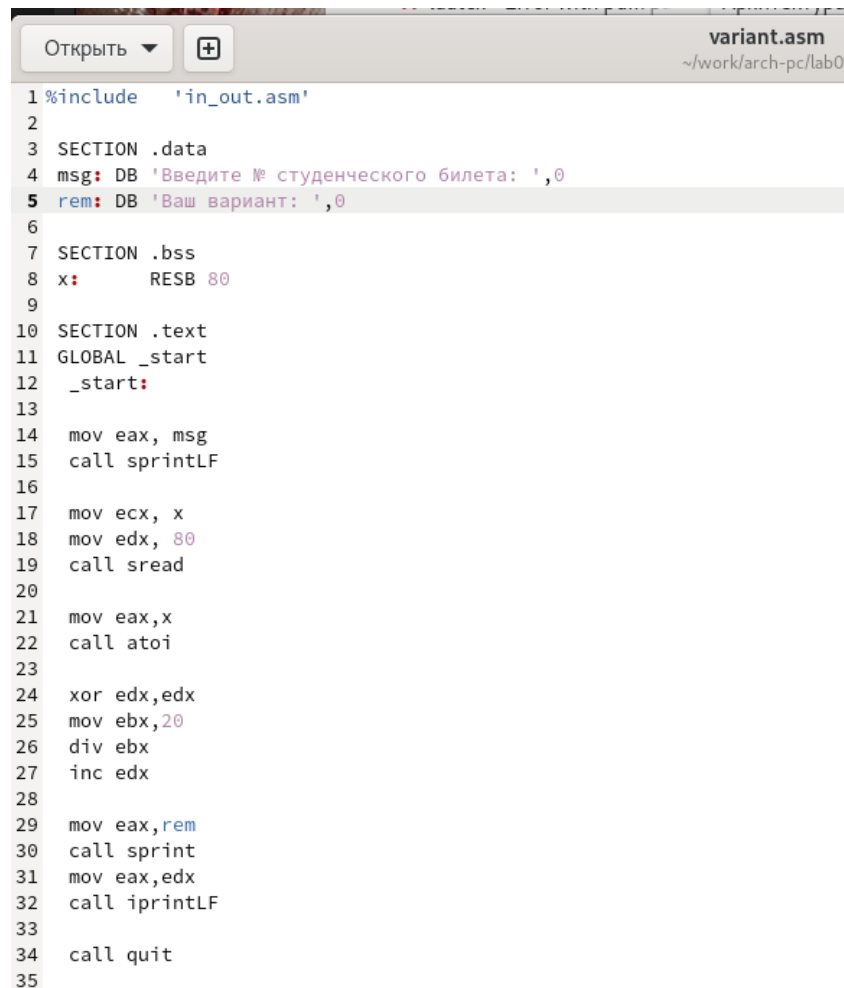
```

[mekosolapov@fedora lab07]$ touch variant.asm
[mekosolapov@fedora lab07]$ gedit variant.asm
[mekosolapov@fedora lab07]$

```

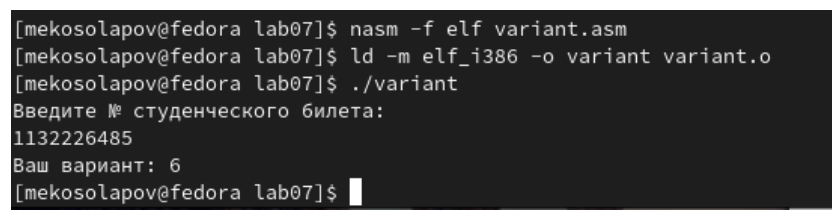
Рис. 4.17: Создание файла variant.asm

16. Переносим текст предложенной программы в файл, создаём исполняемый файл и проверяем его работу (рис. 4.18- рис. 4.19):



```
1 %include 'in_out.asm'
2
3 SECTION .data
4 msg: DB 'Введите № студенческого билета: ',0
5 rem: DB 'Ваш вариант: ',0
6
7 SECTION .bss
8 x:    RESB 80
9
10 SECTION .text
11 GLOBAL _start
12 _start:
13
14 mov eax, msg
15 call sprintLF
16
17 mov ecx, x
18 mov edx, 80
19 call sread
20
21 mov eax, x
22 call atoi
23
24 xor edx, edx
25 mov ebx, 20
26 div ebx
27 inc edx
28
29 mov eax, rem
30 call sprint
31 mov eax, edx
32 call iprintLF
33
34 call quit
35
```

Рис. 4.18: Программа variant



```
[mekosolapov@fedora lab07]$ nasm -f elf variant.asm
[mekosolapov@fedora lab07]$ ld -m elf_i386 -o variant variant.o
[mekosolapov@fedora lab07]$ ./variant
Введите № студенческого билета:
1132226485
Ваш вариант: 6
[mekosolapov@fedora lab07]$
```

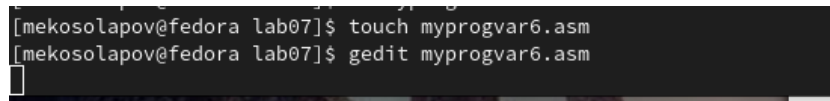
Рис. 4.19: Создание исполняемого файла и результат выполнения программы variant

17. Ответы на вопросы:

1. За вывод сообщения: 'Ваш вариант:' отвечают строки, не считая строки, в которой задаётся сообщение: `"mov eax,ret"`, `"call sprint"`
2. `"nasm"` - создание объектного файла, `"mov ecx,x"` - помещение в `ecx` значения `x`, `"mov edx,80"` - помещение в `edx` значения `80`, `"call spread"` - считывание введённых данных
3. `"call atoi"` используется для преобразования ASCII символов в числа.
4. За вычисления варианта отвечают строки: `"xor edx,edx"`, `"mov ebx,20"`, `"div ebx"`, `"inc edx"`
5. Остаток от деления при выполнении инструкции `"div ebx"` записывается в `edx`
6. Инструкция `"inc edx"` используется для прибавления `1` к `edx`
7. За вывод на экран результата вычислений отвечают строки: `"mov eax,edx"`, `"call iprintLF"`

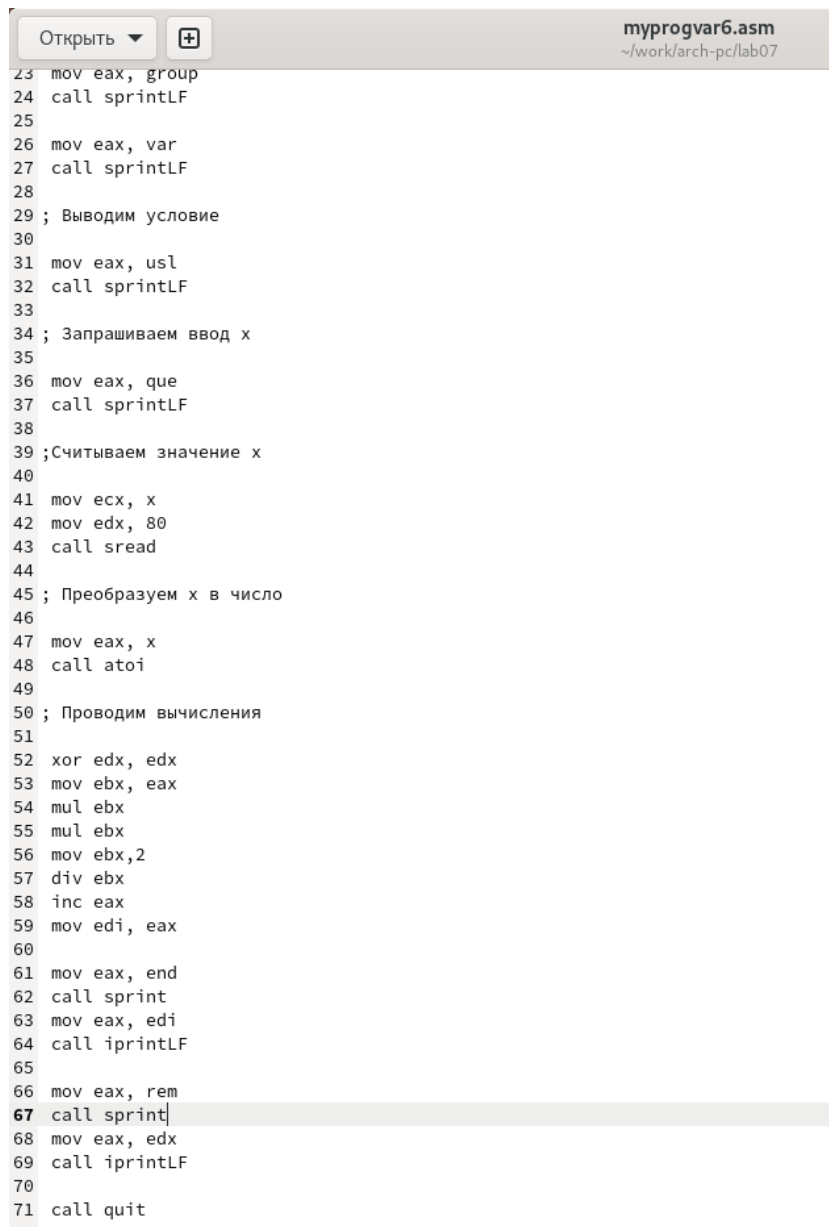
Задания для самостоятельной работы

18. Так как у меня 6 вариант, то делаем программу для 6 уравнения (рис. 4.20 - рис. 4.21):



```
[mekosolapov@fedora lab07]$ touch myprogvar6.asm
[mekosolapov@fedora lab07]$ gedit myprogvar6.asm
```

Рис. 4.20: Создание файла myprogvar6.asm



```
23 mov eax, group
24 call sprintfLF
25
26 mov eax, var
27 call sprintfLF
28
29 ; Выводим условие
30
31 mov eax, usl
32 call sprintfLF
33
34 ; Запрашиваем ввод x
35
36 mov eax, que
37 call sprintfLF
38
39 ;Считываем значение x
40
41 mov ecx, x
42 mov edx, 80
43 call sread
44
45 ; Преобразуем x в число
46
47 mov eax, x
48 call atoi
49
50 ; Проводим вычисления
51
52 xor edx, edx
53 mov ebx, eax
54 mul ebx
55 mul ebx
56 mov ebx, 2
57 div ebx
58 inc eax
59 mov edi, eax
60
61 mov eax, end
62 call sprint
63 mov eax, edi
64 call iprintLF
65
66 mov eax, rem
67 call sprint
68 mov eax, edx
69 call iprintLF
70
71 call quit
72
```

Рис. 4.21: Программа myprogvar6

19. Создаём исполняемый файл и проверяем его работу на числах 2, 5, 3 (рис. 4.22):


```

[mekosolapov@fedora lab07]$ nasm -f elf myprogvar6.asm
[mekosolapov@fedora lab07]$ ld -m elf_i386 -o myprogvar6 myprogvar6.o
[mekosolapov@fedora lab07]$ ./myprogvar6
Косолапов Матвей
НММбд-01-22
Вариант №6
 $f(x) = x^3/2 + 1$ 
Введите значение x:
2
Ответ: 5
Остаток: 0
[mekosolapov@fedora lab07]$ ./myprogvar6
Косолапов Матвей
НММбд-01-22
Вариант №6
 $f(x) = x^3/2 + 1$ 
Введите значение x:
5
Ответ: 63
Остаток: 1
[mekosolapov@fedora lab07]$ ./myprogvar6
Косолапов Матвей
НММбд-01-22
Вариант №6
 $f(x) = x^3/2 + 1$ 
Введите значение x:
3
Ответ: 14
Остаток: 1
[mekosolapov@fedora lab07]$

```

Рис. 4.22: Создание исполняемого файла и результат выполнения программы myprogvar6

Отлично! Всё работает

5 Выводы

В ходе данной работы я научился писать программы на assembler

Список литературы