

Отчет по лабораторной работе номер 7

Сафин Андрей Алексеевич

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Ответы на вопросы к лабораторной работе	19
5	Самостоятельная работа	20
6	Выводы	23

Список иллюстраций

3.1	Создание файла lab7-1.asm; перемещение файла in_out.asm . . .	7
3.2	Текст первоначальной программы в lab7-1.asm	8
3.3	Трансляция и компоновка lab7-1.asm (далее данный шаг в тексте подчеркиваться не будет, так как его выполнение подразумевается), результат работы исполняемого файла	8
3.4	Программа lab7-1.asm с измененными операндами сложения . .	9
3.5	Результат выполнения lab7-1.asm после редактирования	9
3.6	Первоначальный текст lab7-2.asm	10
3.7	Создание файла lab7-2.asm; результат выполнения написанной программы	10
3.8	Изменения в lab7-2.asm	11
3.9	Повторное выполнение программы из lab7-2.asm	11
3.10	Замена iprintLF на iprint	12
3.11	Результат выполнения измененной программы	12
3.12	Текст lab7-3.asm	13
3.13	Создание файла lab7-3.asm; результат выполнения написанной программы	14
3.14	Новый текст lab7-3.asm	15
3.15	Результат выполнения измененной программы	16
3.16	Текст variant.asm	17
3.17	Создание файла variant.asm; результат выполнения написанной программы	18
5.1	Текст sr.asm	21
5.2	Создание файла sr.asm; результат выполнения написанной программы	22

Список таблиц

1 Цель работы

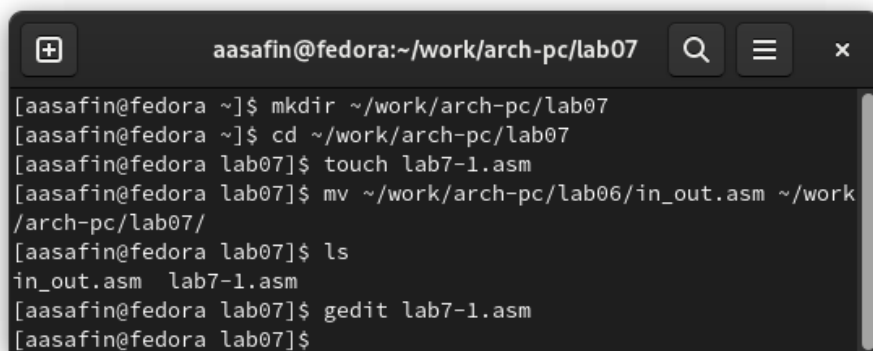
Освоение арифметических инструкций языка ассемблера NASM.

2 Задание

Написать ряд программ, производящих арифметические операции (в том числе с введенными значениями) и выводящие результаты на экран. В процессе написания прийти к корректному вводу и выводу численных данных с помощью команд, обращающихся к коду ASCII.

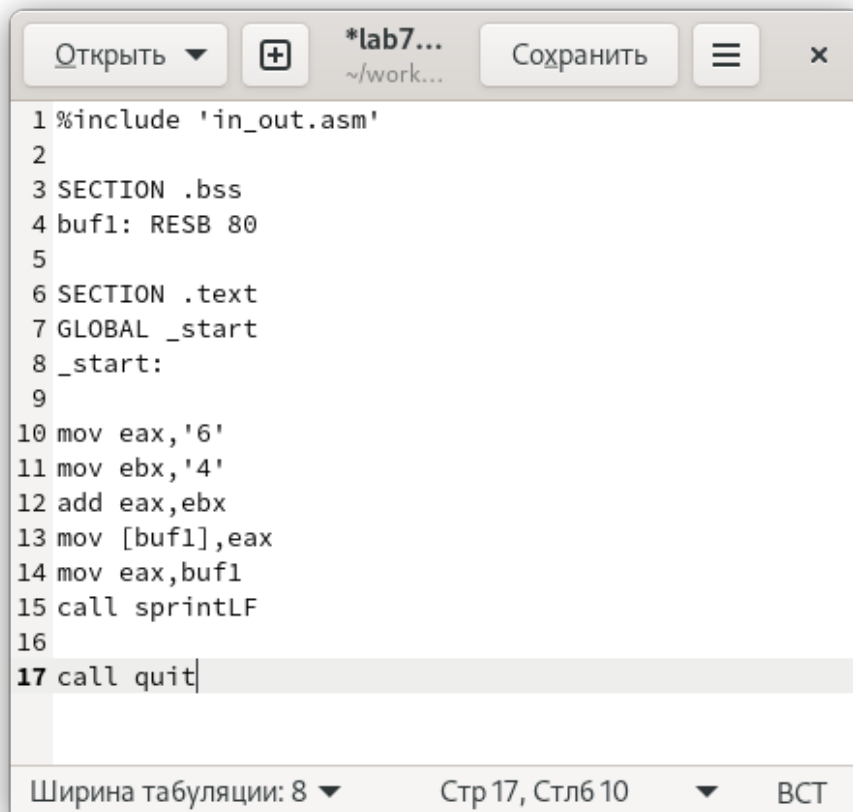
3 Выполнение лабораторной работы

1. Создан файл lab7-1.asm. В соответствующую директорию перемещен файл in_out.asm (рис. 3.1).
2. В файл lab7-1.asm записана программа, выполняющая вычисления (рис. 3.2). Однако результат выполнения такой программы не корректен (рис.3.2), так как все вычисления выполнены с кодами символов '6' и '4'.
3. Программа переписана так, чтоб в качестве операндов сложения выступали именно числа 6 и 4 (рис. 3.4), однако на экран выводится всё ещё неверный результат (рис. 3.5), так как вместо, собственно, числа выводится символ, код которого соответствует этому числу (переводящий на следующую строку символ).



```
aasafin@fedora:~/work/arch-pc/lab07
[aasafin@fedora ~]$ mkdir ~/work/arch-pc/lab07
[aasafin@fedora ~]$ cd ~/work/arch-pc/lab07
[aasafin@fedora lab07]$ touch lab7-1.asm
[aasafin@fedora lab07]$ mv ~/work/arch-pc/lab06/in_out.asm ~/work/arch-pc/lab07/
[aasafin@fedora lab07]$ ls
in_out.asm  lab7-1.asm
[aasafin@fedora lab07]$ gedit lab7-1.asm
[aasafin@fedora lab07]$
```

Рис. 3.1: Создание файла lab7-1.asm; перемещение файла in_out.asm

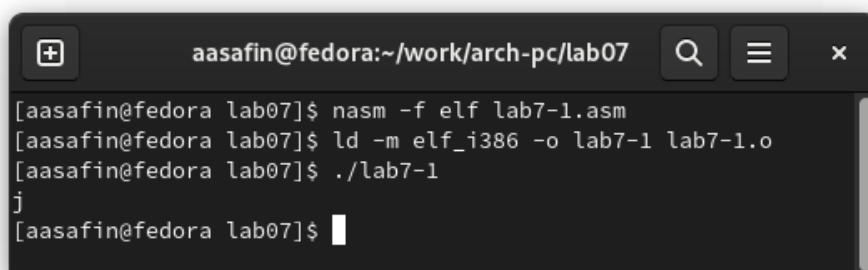


The screenshot shows a text editor window with a title bar containing buttons for 'Открыть', '+', '*lab7...', 'Сохранить', and window controls. The main text area contains assembly code for a program named lab7-1.asm. The code includes a preprocessor directive, section declarations for .bss and .text, global variable declarations, and a series of assembly instructions. The status bar at the bottom indicates 'Ширина табуляции: 8', 'Стр 17, Стлб 10', and 'ВСТ'.

```
1 %include 'in_out.asm'
2
3 SECTION .bss
4 buf1: RESB 80
5
6 SECTION .text
7 GLOBAL _start
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
17 call quit
```

Ширина табуляции: 8 Стр 17, Стлб 10 ВСТ

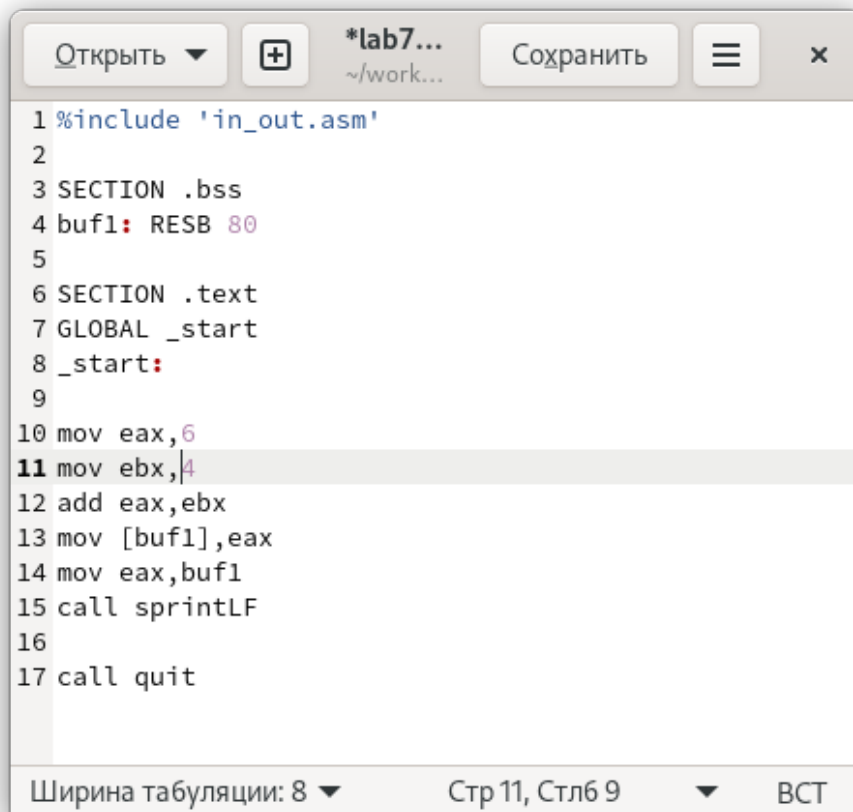
Рис. 3.2: Текст первоначальной программы в lab7-1.asm



The screenshot shows a terminal window with the title 'aasafin@fedora:~/work/arch-pc/lab07'. The terminal displays the commands used to compile and execute the assembly program: nasm -f elf lab7-1.asm, ld -m elf_i386 -o lab7-1 lab7-1.o, and ./lab7-1. The output of the program is the character 'j'.

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

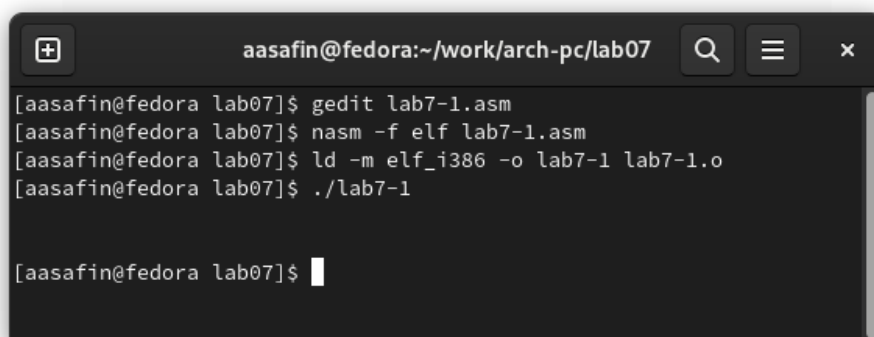
Рис. 3.3: Трансляция и компоновка lab7-1.asm (далее данный шаг в тексте подчеркиваться не будет, так как его выполнение подразумевается), результат работы исполняемого файла



```
1 %include 'in_out.asm'
2
3 SECTION .bss
4 buf1: RESB 80
5
6 SECTION .text
7 GLOBAL _start
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
17 call quit
```

Ширина табуляции: 8 Стр 11, Стлб 9 ВСТ

Рис. 3.4: Программа lab7-1.asm с измененными операндами сложения



```
aasafin@fedora:~/work/arch-pc/lab07
[aasafin@fedora lab07]$ gedit lab7-1.asm
[aasafin@fedora lab07]$ nasm -f elf lab7-1.asm
[aasafin@fedora lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[aasafin@fedora lab07]$ ./lab7-1

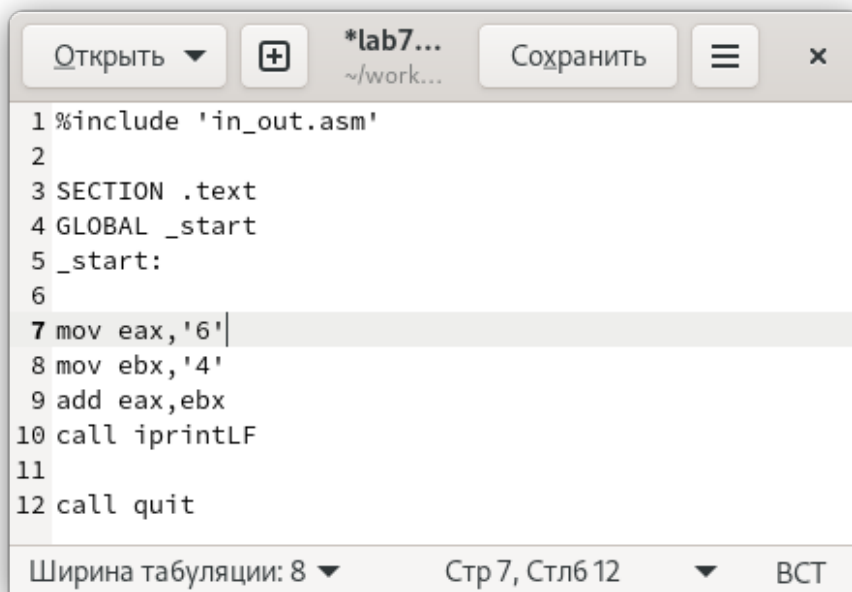
[aasafin@fedora lab07]$
```

Рис. 3.5: Результат выполнения lab7-1.asm после редактирования

4. Создан файл lab7-2.asm (рис. 3.7), в котором написана программа с приведением численной информации к соответствующим значениям кода ASCII

при выводе на экран (рис. 3.6), однако результат выполнения которой всё ещё некорректен, так как в качестве операндов указываются строки '6' и '4' (рис. 3.7).

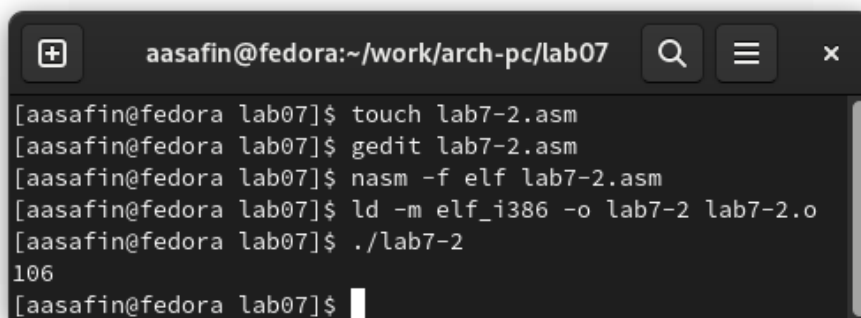
5. Затем в качестве операндов записаны численные значения (рис. 3.8), в результате чего программа выводит на экран верный результат: 10 (рис. 3.9).



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

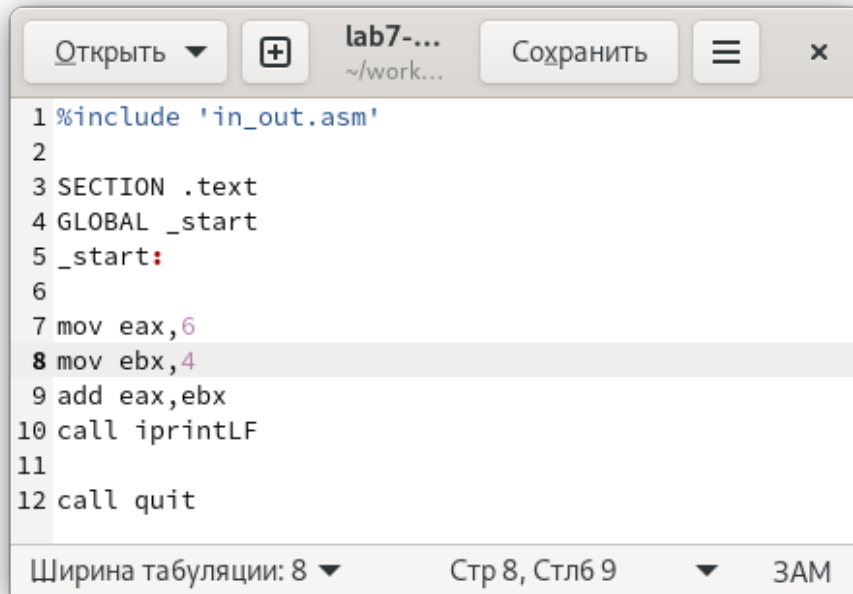
Ширину табуляции: 8 Стр 7, Стл 6 12 ВСТ

Рис. 3.6: Первоначальный текст lab7-2.asm



```
aasafin@fedora: ~/work/arch-pc/lab07
[aasafin@fedora lab07]$ touch lab7-2.asm
[aasafin@fedora lab07]$ gedit lab7-2.asm
[aasafin@fedora lab07]$ nasm -f elf lab7-2.asm
[aasafin@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[aasafin@fedora lab07]$ ./lab7-2
106
[aasafin@fedora lab07]$
```

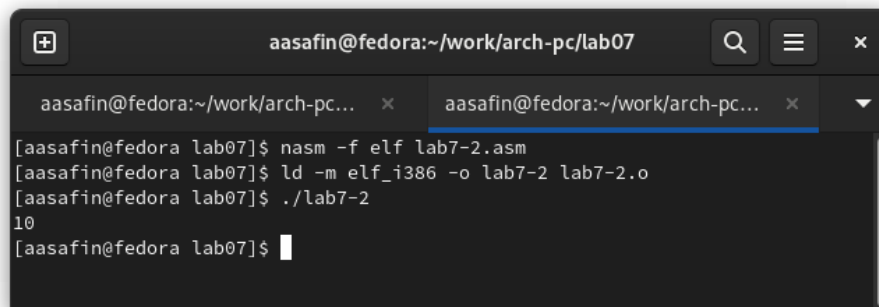
Рис. 3.7: Создание файла lab7-2.asm; результат выполнения написанной программы



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

Ширина табуляции: 8 Стр 8, Стлб 9 ЗАМ

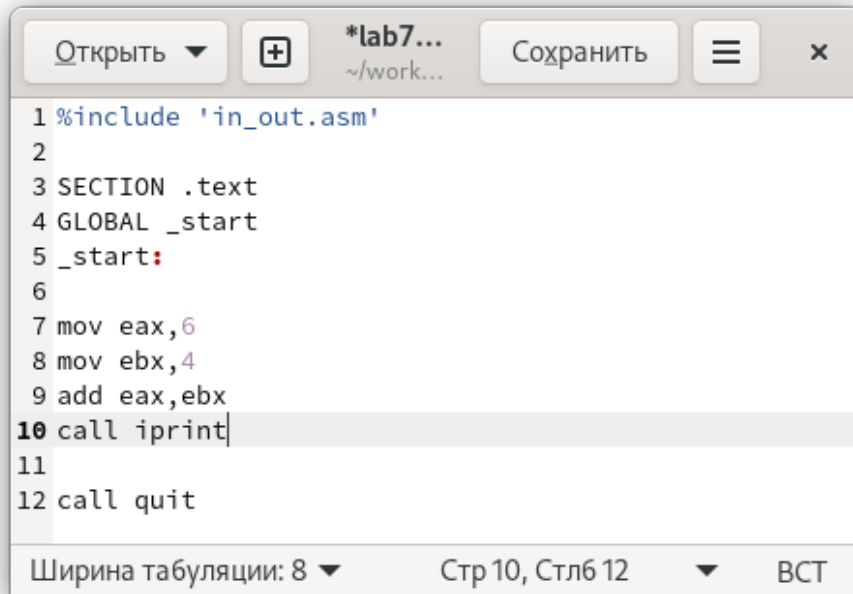
Рис. 3.8: Изменения в lab7-2.asm



```
aasafin@fedora:~/work/arch-pc/lab07
aasafin@fedora:~/work/arch-pc... x aasafin@fedora:~/work/arch-pc... x
[aasafin@fedora lab07]$ nasm -f elf lab7-2.asm
[aasafin@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[aasafin@fedora lab07]$ ./lab7-2
10
[aasafin@fedora lab07]$
```

Рис. 3.9: Повторное выполнение программы из lab7-2.asm

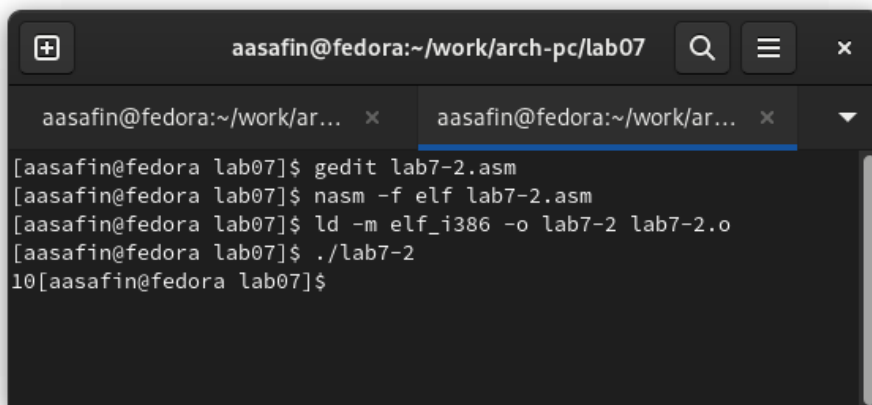
Далее вместо команды `iprintLF` в `lab7-2.asm` записана команда `iprint` (рис. 3.10). Как можно видеть по результату (рис. 3.11), разница состоит в том, что вторая команда не приводит к переходу на следующую строку при выведении результата на экран.



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

Ширина табуляции: 8 Стр 10, Стлб 12 ВСТ

Рис. 3.10: Замена iprintLF на iprint



```
aasafin@fedora:~/work/arch-pc/lab07
aasafin@fedora:~/work/ar... x aasafin@fedora:~/work/ar... x
[aasafin@fedora lab07]$ gedit lab7-2.asm
[aasafin@fedora lab07]$ nasm -f elf lab7-2.asm
[aasafin@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[aasafin@fedora lab07]$ ./lab7-2
10[aasafin@fedora lab07]$
```

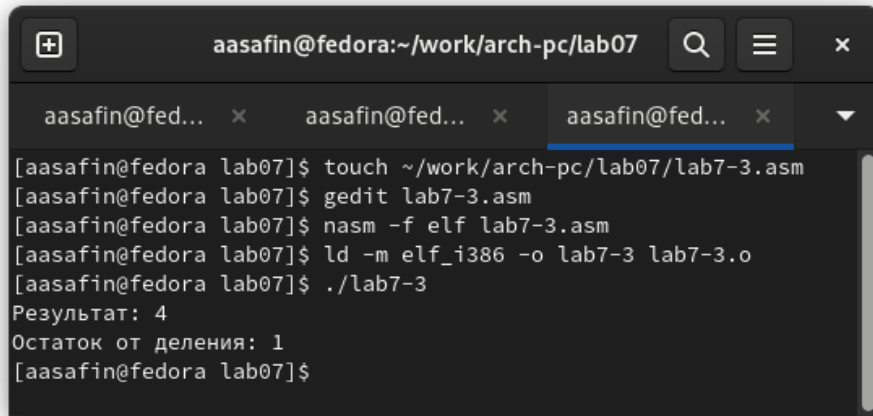
Рис. 3.11: Результат выполнения измененной программы

6. Был создан файл lab7-3.asm (рис. 3.13), в котором была написана программа, выводящая значение выражения $(5*2+3)/3$ вместе с остатком от деления (рис. 3.12, 3.13). Затем текст lab7-3.asm был изменен так (рис. 3.14), чтоб новый исполняемый файл приводил к выведению на экран значения $(4*6+2)/5$ (рис. 3.15).

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

Ширина табуляции: 8 ▼ Стр 2, Стлб 1 ▼ ВСТ

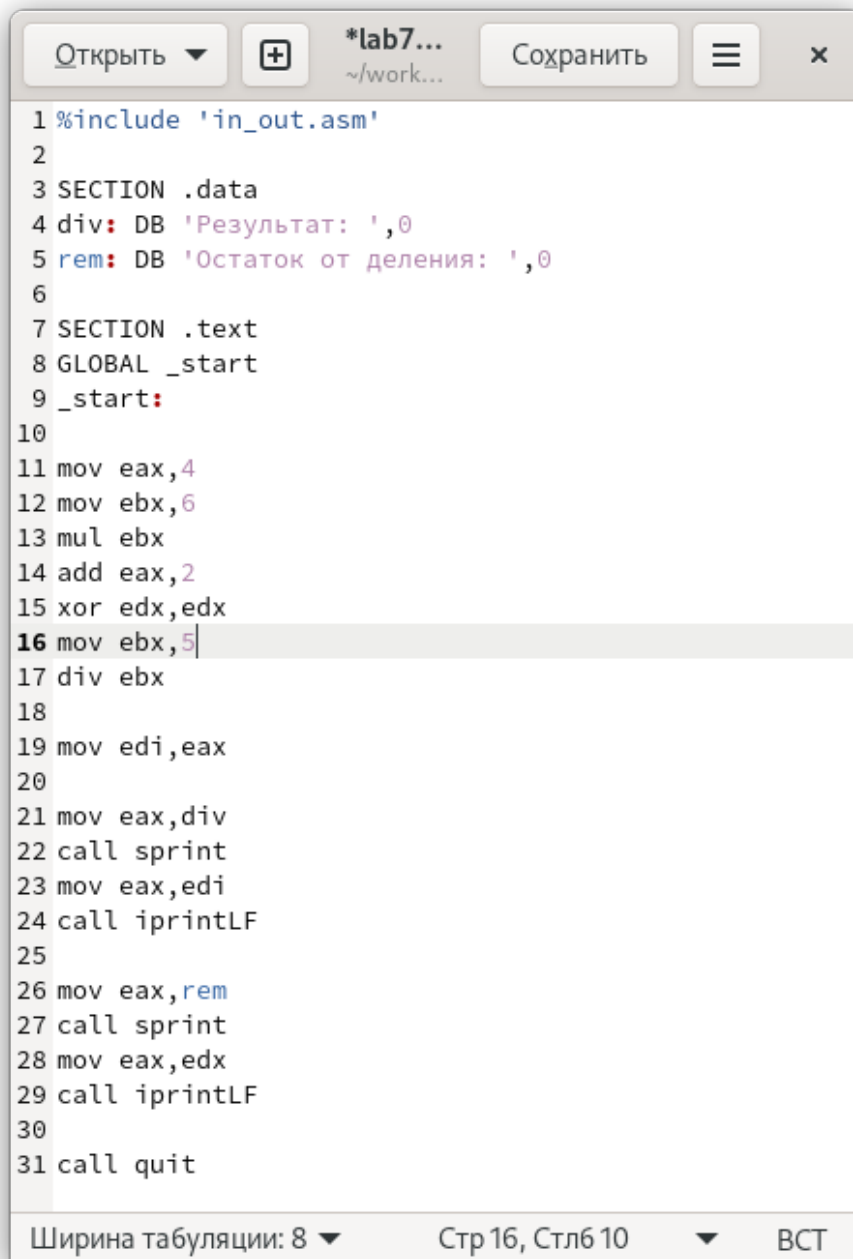
Рис. 3.12: Текст lab7-3.asm



The image shows a terminal window with a dark background. The title bar at the top reads 'aasafin@fedora:~/work/arch-pc/lab07'. There are three tabs open, all with the same title. The terminal content shows a series of commands and their outputs:

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

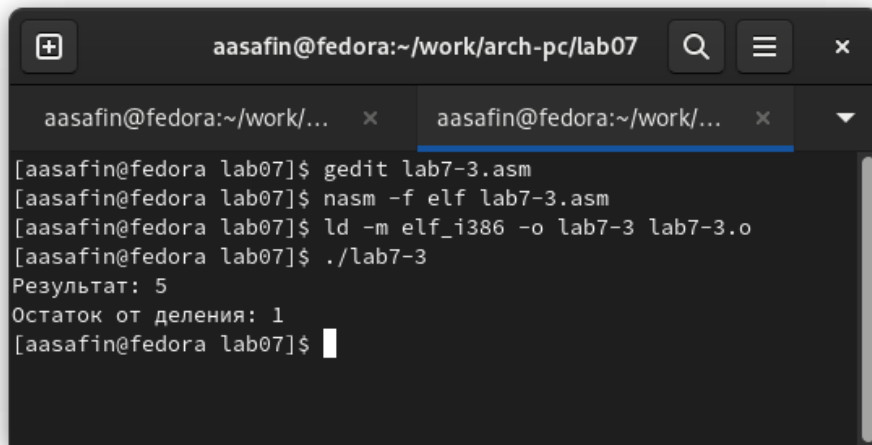
Рис. 3.13: Создание файла lab7-3.asm; результат выполнения написанной программы



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

Ширина табуляции: 8 Стр 16, Стлб 10 ВСТ

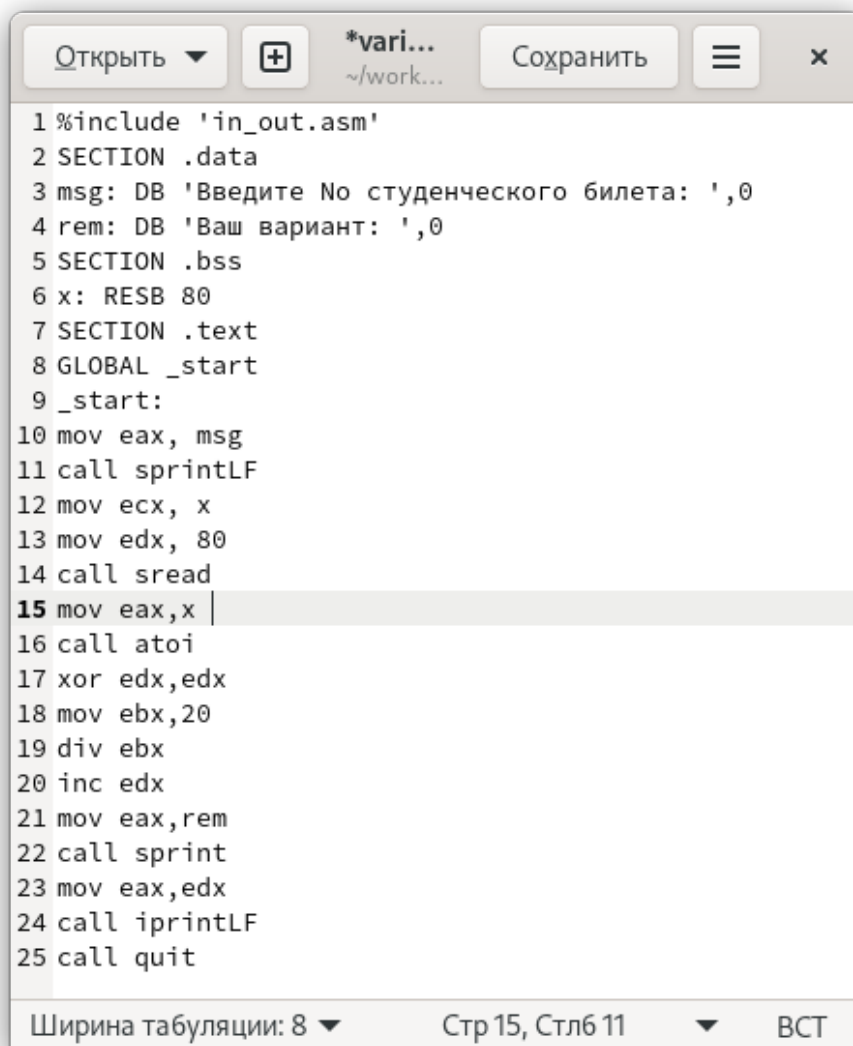
Рис. 3.14: Новый текст lab7-3.asm

A terminal window titled 'aasafin@fedora:~/work/arch-pc/lab07' with search and menu icons. It shows a sequence of commands: 'gedit lab7-3.asm', 'nasm -f elf lab7-3.asm', 'ld -m elf_i386 -o lab7-3 lab7-3.o', and './lab7-3'. The output of the execution is 'Результат: 5' followed by 'Остаток от деления: 1' on a new line. The prompt '[aasafin@fedora lab07]\$' is visible at the bottom.

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

Рис. 3.15: Результат выполнения измененной программы

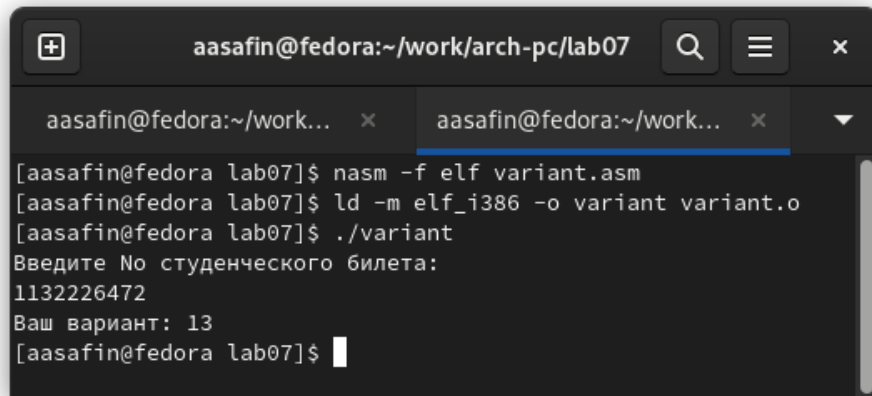
7. Создан файл `variant.asm` (рис. 3.17), в него записана программа (рис. 3.16), вычисляющая вариант задания по введенному номеру студенческого билета по формуле $(x \bmod 20) + 1$ и выводящая номер варианта на экран (рис. 3.17) (для введенного номера билета автора лабораторной работы, очевидно, результат корректен).



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

Ширина табуляции: 8 Стр 15, Стлб 11 ВСТ

Рис. 3.16: Текст variant.asm



```
aasafin@fedora:~/work/arch-pc/lab07
[aasafin@fedora lab07]$ nasm -f elf variant.asm
[aasafin@fedora lab07]$ ld -m elf_i386 -o variant variant.o
[aasafin@fedora lab07]$ ./variant
Введите No студенческого билета:
1132226472
Ваш вариант: 13
[aasafin@fedora lab07]$
```

Рис. 3.17: Создание файла variant.asm; результат выполнения написанной программы

4 Ответы на вопросы к лабораторной работе

1. `mov eax,rem; call sprint`
2. `mov ecx, x` - перемещение адреса `x` в регистр `ecx`; `mov edx, 80` - запись в регистр `edx` значения `80`; `call sread` - вызов кода из `in_out.asm`, отвечающего за ввод строки в переменную, адрес которой хранится в `eax` (при этом длина строки должна быть обозначена в `ebx`).
3. `call atoi` преобразует код ASCII в регистре `eax` в целочисленное значение, которое обозначается соответствующими символами.
4. `xor edx,edx; mov ebx,20; div ebx; inc edx`
5. В регистр `edx`
6. Для увеличения значения в `edx` на единицу
7. `mov eax,edx; call iprintLF`

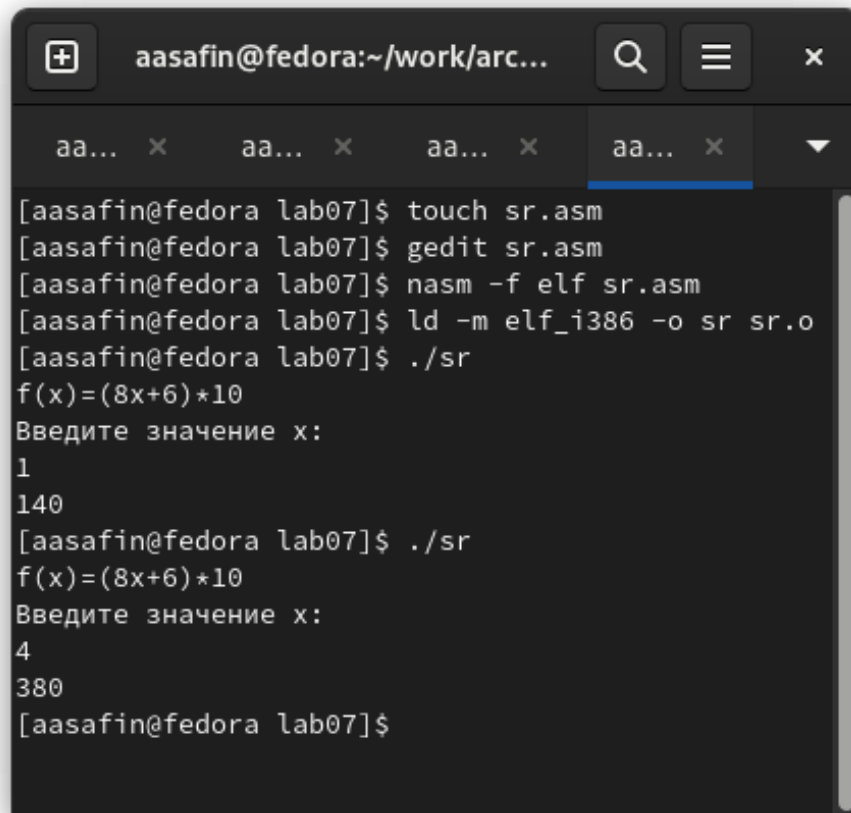
5 Самостоятельная работа

Создан файл `sr.asm` (рис. 5.2). Написана программа, выводящая выражение $f(x)=(8x+6)*10$, выполняющая вычисление значения $f(x)=(8x+6)*10$ по введенному x и выводящая значение на экран (рис. 5.1). Её работа проверена на значениях 1 и 4 (рис. 5.2). Результат корректен.

```
1 %include 'in_out.asm'
2
3 SECTION .data
4
5 msg: DB 'Введите значение x: ',0
6 func: DB 'f(x)=(8x+6)*10',0
7
8 SECTION .bss
9
10 x: RESB 80
11
12 SECTION .text
13 GLOBAL _start
14 _start:
15
16 mov eax, func
17 call sprintf
18
19 mov eax, msg
20 call sprintf
21
22 mov ecx, x
23 mov edx, 80
24 call sread
25
26 mov eax, x
27 call atoi
28
29 mov ebx, 8
30 mul ebx
31
32 mov ebx, 6
33 add eax, ebx
34
35 mov ebx, 10
36 mul ebx
37
38 call iprintLF
39 call quit
```

Ширина табуляции: 8 Стр 1, Стлб 1 ВСТ

Рис. 5.1: Текст sr.asm



The image shows a terminal window with a dark background. The title bar at the top reads 'aasafin@fedora:~/work/arc...'. There are four tabs open, each labeled 'aa...'. The terminal content shows the following sequence of commands and output:

```
[aasafin@fedora lab07]$ touch sr.asm
[aasafin@fedora lab07]$ gedit sr.asm
[aasafin@fedora lab07]$ nasm -f elf sr.asm
[aasafin@fedora lab07]$ ld -m elf_i386 -o sr sr.o
[aasafin@fedora lab07]$ ./sr
f(x)=(8x+6)*10
Введите значение x:
1
140
[aasafin@fedora lab07]$ ./sr
f(x)=(8x+6)*10
Введите значение x:
4
380
[aasafin@fedora lab07]$
```

Рис. 5.2: Создание файла sr.asm; результат выполнения написанной программы

6 Выводы

Все программы созданы и исполнены. Ответы на вопросы даны. В рамках самостоятельной работы предоставленная информация использована для написания программы с соответствующей функцией.