

Лабораторная работа №6

Отчет

Зубов Иван Александрович

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	16

Список иллюстраций

2.1	Создаем каталог с помощью команды <code>mkdir</code> и файл с помощью команды <code>touch</code>	6
2.2	Запускаем файл и смотрим на его работу	7
2.3	Изменяем файл	7
2.4	Запускаем файл и смотрим на его работу	7
2.5	Заполняем файл	8
2.6	Смотрим на работу программы	8
2.7	Изменяем файл	9
2.8	Смотрим на работу программы	9
2.9	Изменяем файл	10
2.10	Смотрим, как сработала программа	10
2.11	Заполняем файл	11
2.12	Смотрим на результат работы программы	11
2.13	Редактируем файл	12
2.14	Заполняем файл	12
2.15	Проверяем правильность написания программы	13
2.16	Проверяем результат работы программы	13
2.17	Заполняем этот файл, чтобы решалось уравнение $(8x-6)/2$	14
2.18	Смотрим как работает программа	15

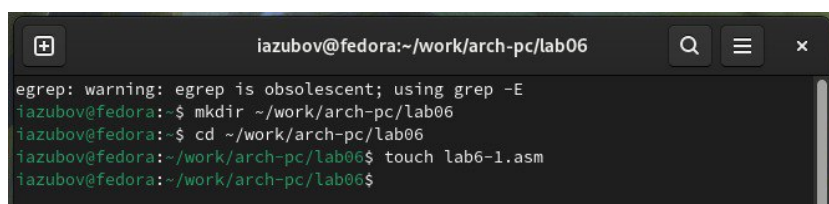
Список таблиц

1 Цель работы

Освоить арифметических инструкций языка ассемблера NASM и написать программы для вычисления арифметических выражений с неизвестной.

2 Выполнение лабораторной работы

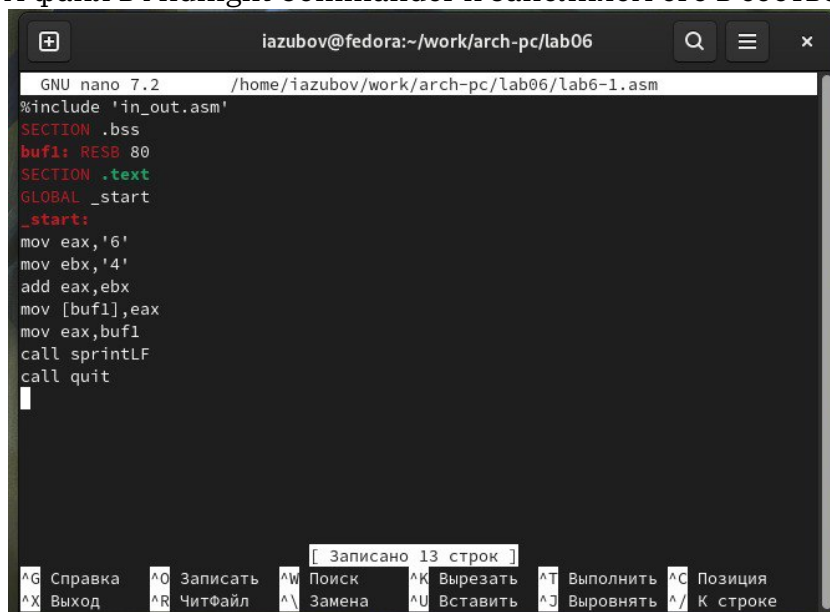
Создаем каталог для программ ЛБ6, в нем создаем файл и с помощью команды touch создаем файл



```
iazubov@fedora:~/work/arch-pc/lab06
egrep: warning: egrep is obsolescent; using grep -E
iazubov@fedora:~$ mkdir ~/work/arch-pc/lab06
iazubov@fedora:~$ cd ~/work/arch-pc/lab06
iazubov@fedora:~/work/arch-pc/lab06$ touch lab6-1.asm
iazubov@fedora:~/work/arch-pc/lab06$
```

Рис. 2.1: Создаем каталог с помощью команды mkdir и файл с помощью команды touch

Открываем файл в Midnight Commander и заполняем его в соответствии с ли-



```
GNU nano 7.2 /home/iazubov/work/arch-pc/lab06/lab6-1.asm
#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 sprintLF
call quit
```

[Записано 13 строк]

^G Справка	^O Записать	^W Поиск	^K Вырезать	^T Выполнить	^C Позиция
^X Выход	^R ЧитФайл	^_ Замена	^U Вставить	^J Выводить	^/ К строке

стингом 6.1

Создаем исполняемый файл и запускаем его

```

iazubov@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
iazubov@fedora:~/work/arch-pc/lab06$ ./lab6-1
j
iazubov@fedora:~/work/arch-pc/lab06$

```

Рис. 2.2: Запускаем файл и смотрим на его работу

Снова открываем файл для редактирования и убираем кавычки с числовых значений



```

GNU nano 7.2 /home/iazubov/work/arch-pc/lab06/lab6-1.asm
#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 sprintLF
call quit

```

Прочитано 13 строк

^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
 ^X Выход ^R ЧитФайл ^_ Замена ^U Вставить ^J Выровнять ^/ К строке

Рис. 2.3: Изменяем файл

Создаем исполняемый файл и запускаем его

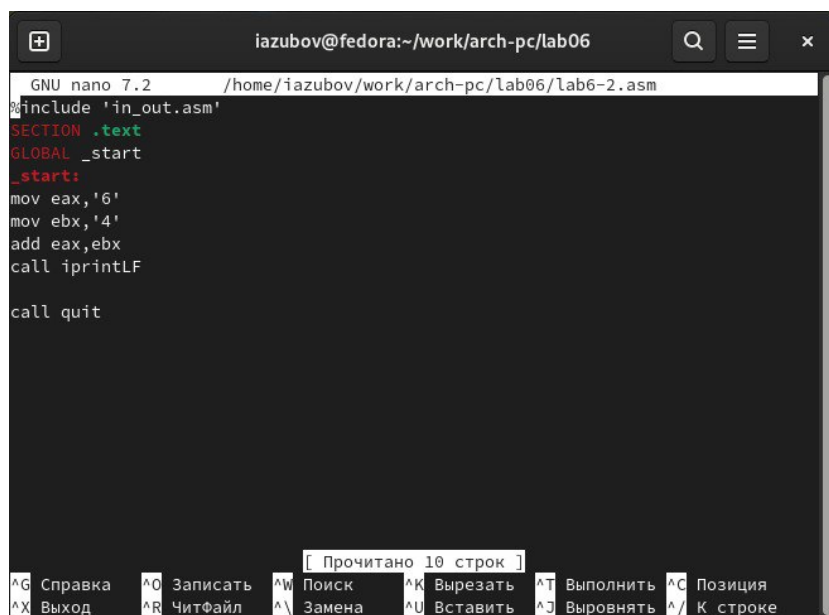
```

iazubov@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
iazubov@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
iazubov@fedora:~/work/arch-pc/lab06$ ./lab6-1
j
iazubov@fedora:~/work/arch-pc/lab06$

```

Рис. 2.4: Запускаем файл и смотрим на его работу

Создаем новый файл в каталоге и заполняем файл в соответствии с листингом 6.2

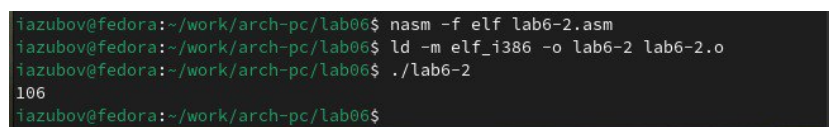


```
GNU nano 7.2 /home/iazubov/work/arch-pc/lab06/lab6-2.asm
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,'6'
mov ebx,'4'
add eax,ebx
call iprintLF

call quit
```

Рис. 2.5: Заполняем файл

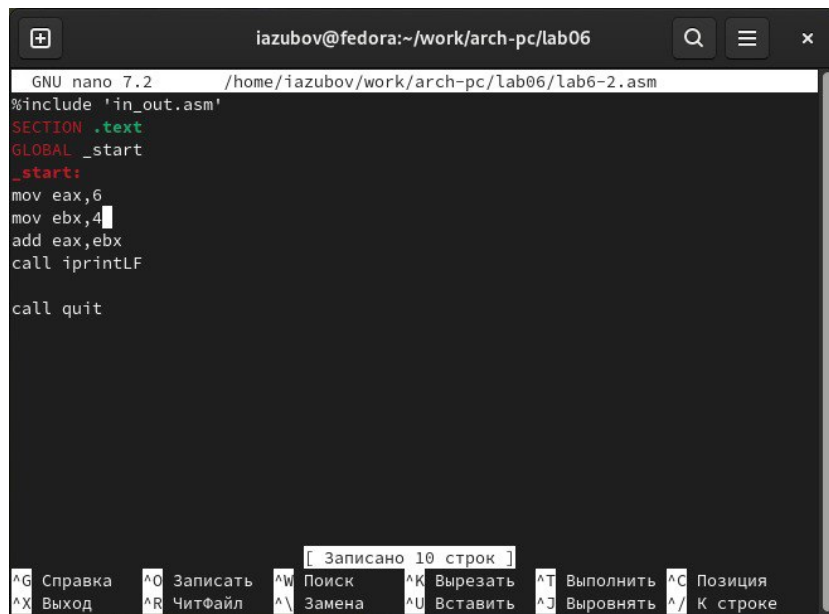
Создаем исполняемый файл и запускаем его



```
iazubov@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
iazubov@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
iazubov@fedora:~/work/arch-pc/lab06$ ./lab6-2
106
iazubov@fedora:~/work/arch-pc/lab06$
```

Рис. 2.6: Смотрим на работу программы

Снова открываем файл для редактирования и убираем кавычки с числовых значений



```
GNU nano 7.2 /home/iazubov/work/arch-pc/lab06/lab6-2.asm
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprintLF

call quit
```

[Записано 10 строк]

^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^J Выводить ^_ К строке

Рис. 2.7: Изменяем файл

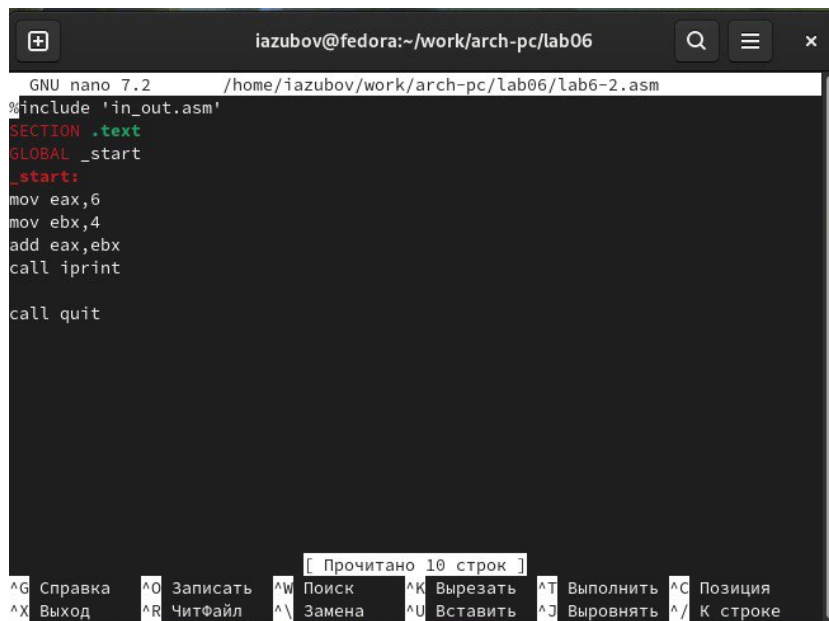
Создаем исполняемый файл и запускаем его



```
iazubov@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
iazubov@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
iazubov@fedora:~/work/arch-pc/lab06$ ./lab6-2
10
iazubov@fedora:~/work/arch-pc/lab06$
```

Рис. 2.8: Смотрим на работу программы

Снова открываем файл для редактирования и меняем iprintLF на iprint



```
GNU nano 7.2 /home/iazubov/work/arch-pc/lab06/lab6-2.asm
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprint

call quit
```

^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^_ Выводить ^/ К строке

Рис. 2.9: Изменяем файл

Создаем исполняемый файл и запускаем его

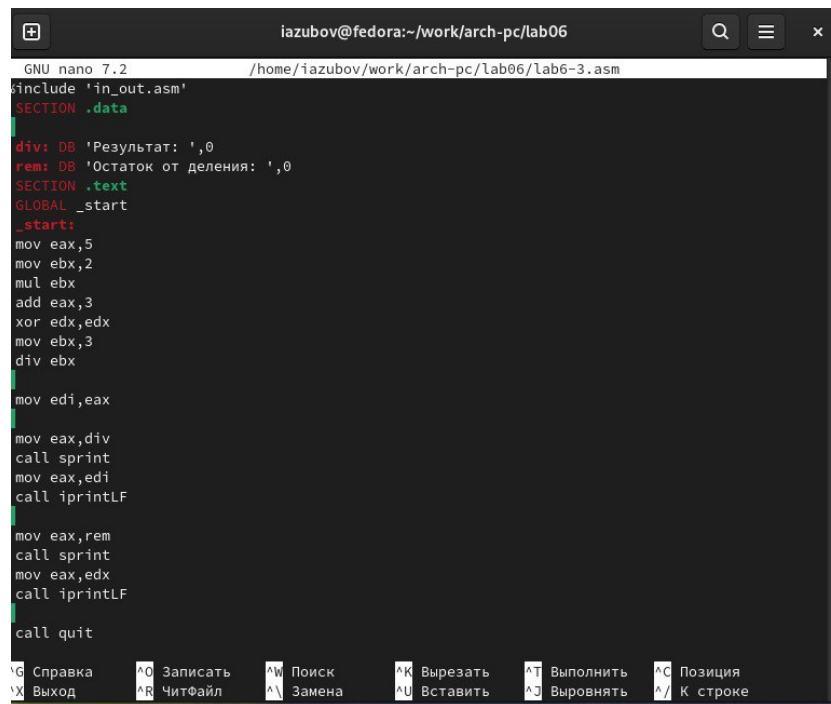


```
iazubov@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
iazubov@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
iazubov@fedora:~/work/arch-pc/lab06$ ./lab6-2
10iazubov@fedora:~/work/arch-pc/lab06$ mc
iazubov@fedora:~/work/arch-pc/lab06$
```

Рис. 2.10: Смотрим, как сработала программа

Вывод функций `iprintLF` и `iprint` отличаются только тем, что `LF` переносит на новую строку.

Создаем новый файл в каталоге, открываем файл и редактируем в соответствии с листингом 6.3



```
GNU nano 7.2 /home/iazubov/work/arch-pc/lab06/lab6-3.asm
#include 'in_out.asm'
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

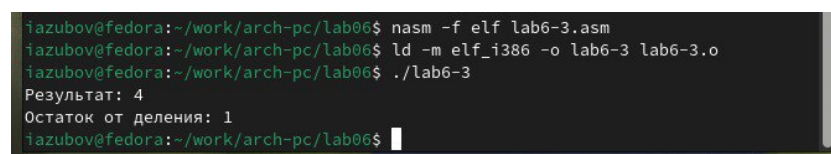
mov eax,div
call sprint
mov eax,edi
call iprintLF

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

call quit
```

Рис. 2.11: Заполняем файл

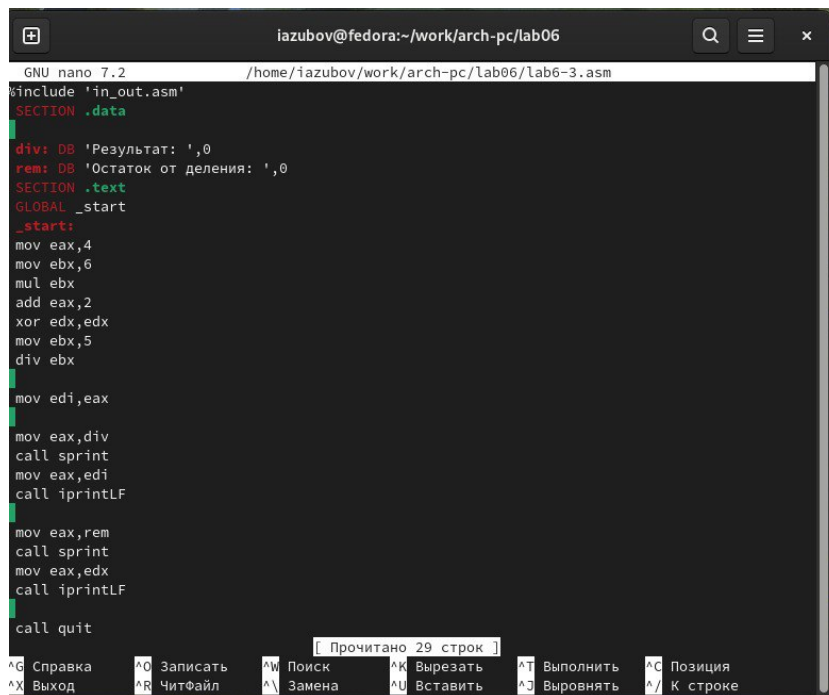
Создаем исполняемый файл и запускаем его



```
iazubov@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
iazubov@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
iazubov@fedora:~/work/arch-pc/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
iazubov@fedora:~/work/arch-pc/lab06$
```

Рис. 2.12: Смотрим на результат работы программы

Открываем файл и редактируем его для вычисления выражения $f(\diamond) = (4 \diamond 6 + 2)/5$



```
GNU nano 7.2 /home/iazubov/work/arch-pc/lab06/lab6-3.asm
#include 'in_out.asm'
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
```

[Прочитано 29 строк]

^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^_ Выводить ^/_ К строке

Рис. 2.13: Редактируем файл

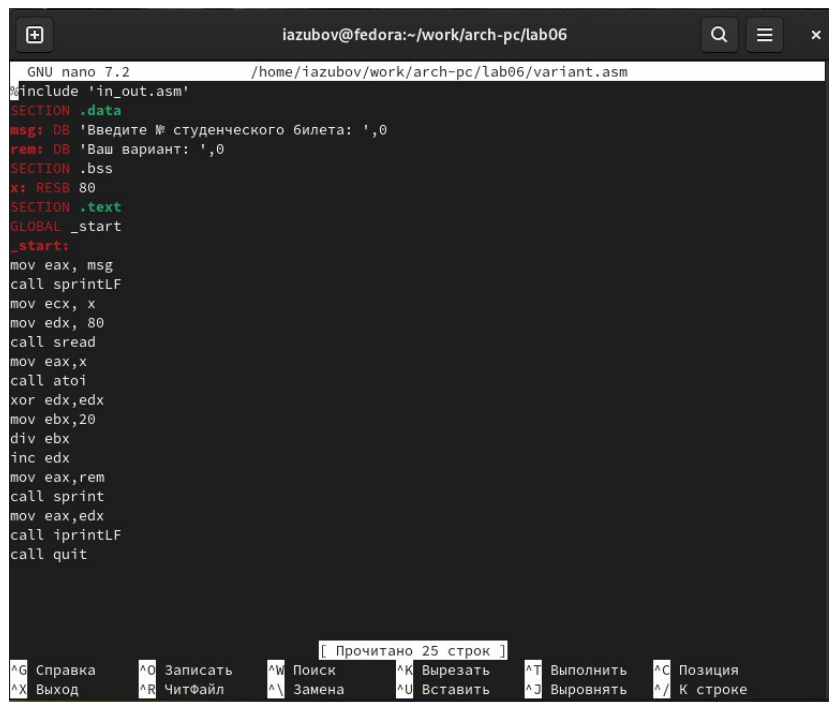
Компилируем файл и запускаем программу



```
iazubov@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
iazubov@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
iazubov@fedora:~/work/arch-pc/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1
iazubov@fedora:~/work/arch-pc/lab06$
```

Рис. 2.14: Заполняем файл

Создаем новый файл в каталоге, открываем и редактируем в соответствии с листингом 6.4



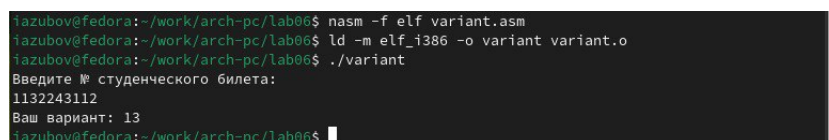
```
GNU nano 7.2 /home/iazubov/work/arch-pc/lab06/variant.asm
#include 'in_out.asm'
SECTION .data
msg: DB 'Введите № студенческого билета: ',0
rem: DB 'Ваш вариант: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintf
mov ecx, x
mov edx, 80
call sread
mov eax, x
call atoi
xor edx, edx
mov ebx, 20
div ebx
inc edx
mov eax, rem
call sprintf
mov eax, edx
call iprintf
call quit
```

Прочитано 25 строк

Справка Записать Поиск Вырезать Выполнить Позиция
Выход ЧитФайл Замена Вставить Выводить К строке

Рис. 2.15: Провераем правильность написания программы

Компилируем файл и запускаем его



```
iazubov@fedora:~/work/arch-pc/lab06$ nasm -f elf variant.asm
iazubov@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
iazubov@fedora:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132243112
Ваш вариант: 13
iazubov@fedora:~/work/arch-pc/lab06$
```

Рис. 2.16: Провераем результат работы программы

#Ответы на вопросы 1 Строка “mov eax,rem” и строка “call sprint” отвечают за вывод на экран сообщения ‘Ваш вариант:’.

2 Эти инструкции используются для чтения строки с вводом данных от пользователя. Начальный адрес строки сохраняется в регистре ecx, а количество символов в строке (максимальное количество символов, которое может быть считано) сохраняется в регистре edx. Затем вызывается процедура sread, которая выполняет чтение строки.

3 Инструкция “call atoi” используется для преобразования строки в целое число. Она принимает адрес строки в регистре eax и возвращает полученное число в

регистре eax.

4 Строка “xor edx,edx” обнуляет регистр edx перед выполнением деления. Строка “mov ebx,20” загружает значение 20 в регистр ebx. Строка “div ebx” выполняет деление регистра eax на значение регистра ebx с сохранением частного в регистре eax и остатка в регистре edx.

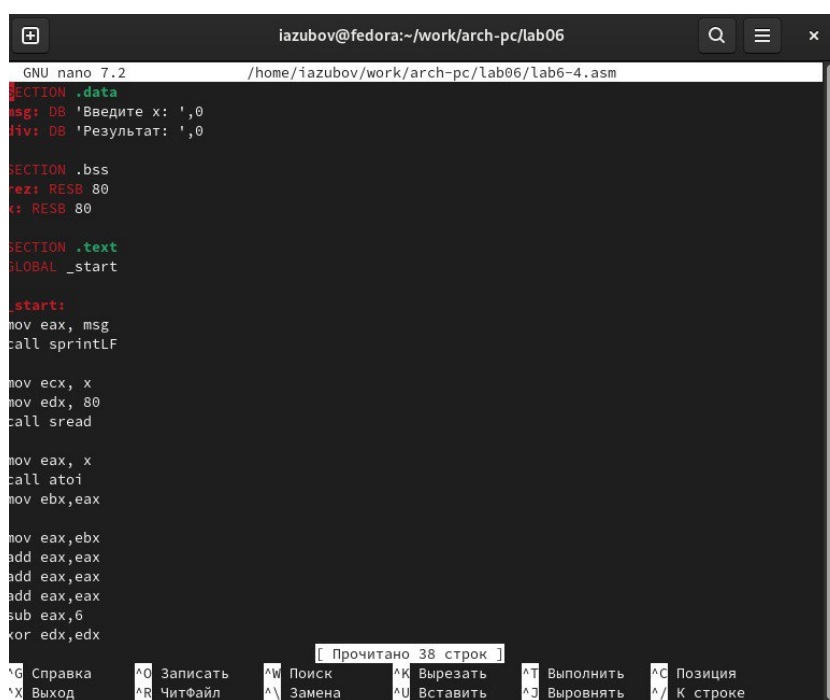
5 Остаток от деления записывается в регистр edx.

6 Инструкция “inc edx” используется для увеличения значения в регистре edx на 1 В данном случае, она увеличивает остаток от деления на 1

7 Строка “mov eax,edx” передает значение остатка от деления в регистр eax. Строка “call iprintLF” вызывает процедуру iprintLF для вывода значения на экран вместе с переводом строки.

#Самостоятельная работа

Создаем новый файл в каталоге,открываем файл и редактируем в соответствии с листингом 6.3



The screenshot shows a terminal window with the nano text editor. The title bar indicates the user is iazubov@fedora in the directory ~/work/arch-pc/lab06. The editor is open to the file /home/iazubov/work/arch-pc/lab06/lab6-4.asm. The code is written in assembly language and includes sections for data, bss, and text. The text section contains instructions for reading input, performing calculations, and printing the result. The status bar at the bottom shows 'Прочитано: 38 строк' and various keyboard shortcuts for file operations.

```
GNU nano 7.2 /home/iazubov/work/arch-pc/lab06/lab6-4.asm
SECTION .data
msg: DB 'Введите x: ',0
div: DB 'Результат: ',0

SECTION .bss
res: RESB 80
x: RESB 80

SECTION .text
GLOBAL _start

_start:
mov eax, msg
call sprintf

mov ecx, x
mov edx, 80
call sread

mov eax, x
call atoi
mov ebx, eax

mov eax, ebx
add eax, eax
add eax, eax
add eax, eax
sub eax, 6
xor edx, edx
```

Рис. 2.17: Заполняем этот файл, чтобы решалось уравнение $(8x-6)/2$

Проверяем программу для $x=1$

```
iazubov@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-4.asm
iazubov@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-4 lab6-4.o
iazubov@fedora:~/work/arch-pc/lab06$ ./lab6-4
Введите x:
1
Результат: 2147483647
```

Рис. 2.18: Смотрим как работает программа

3 Выводы

Мы приобрели навыки создания исполнительных файлов для решения выражений и освоили арифметические инструкции в NASM.