

Шаблон отчёта по лабораторной работе

Простейший вариант

Лупупа Чилеше

Содержание

1	Цель работы	5
2	Задание	6
2.1	Выполнение арифметических операций в NASM	12
3	Задание для самостоятельной работы	17
4	Выводы	19

Список иллюстраций

2.1	lab6-1.asm	6
2.2	листинга 6.1.	7
2.3	листинга 6.1	8
2.4	исполняемый файл	8
2.4	исполняемый файл Листинг 6.1	9
2.5	lab6-2.asm	10
2.6	листинга 6.2	10
2.7	исполняемый файл листинга 6.2	10
2.7	iprint	11
2.8	lab6-3.asm	12
2.9	листинга 6.3	12
2.10	исполняемый файл листинга 6.3	13
2.11	исполняемый файл 2	13
2.12	variant.asm	14
2.13	листинга 6.4	14
2.14	исполняемый файл листинга 6.4	15
3.1	Вариант 15	17
3.2	исполняемый файл	18

Список таблиц

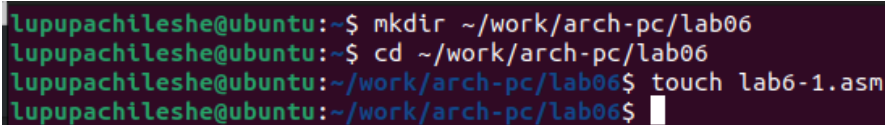
1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Задание

1. Создайте каталог для программ лабораторной работы № 6, перейдите в него и создайте файл lab6-1.asm:

```
mkdir ~/work/arch-pc/lab06 cd ~/work/arch-pc/lab06 touch lab6-1.asm
```



```
lupupachileshe@ubuntu:~$ mkdir ~/work/arch-pc/lab06
lupupachileshe@ubuntu:~$ cd ~/work/arch-pc/lab06
lupupachileshe@ubuntu:~/work/arch-pc/lab06$ touch lab6-1.asm
lupupachileshe@ubuntu:~/work/arch-pc/lab06$
```

Рис. 2.1: lab6-1.asm

КОММЕНТАРИЙ: Создал новый файл с именем lab6-1.asm.

2. Рассмотрим примеры программ вывода символьных и численных значений. Программы будут выводить значения записанные в регистр eax. Введите в файл lab6-1.asm текст программы из листинга 6.1.

```
GNU nano 6.2 /home/lupupachileshe/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 sprintf
call quit
```

Read 17 lines

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location
^X Exit	^R Read File	^_ Replace	^U Paste	^J Justify	^_ Go To Line

Рис. 2.2: листинга 6.1.

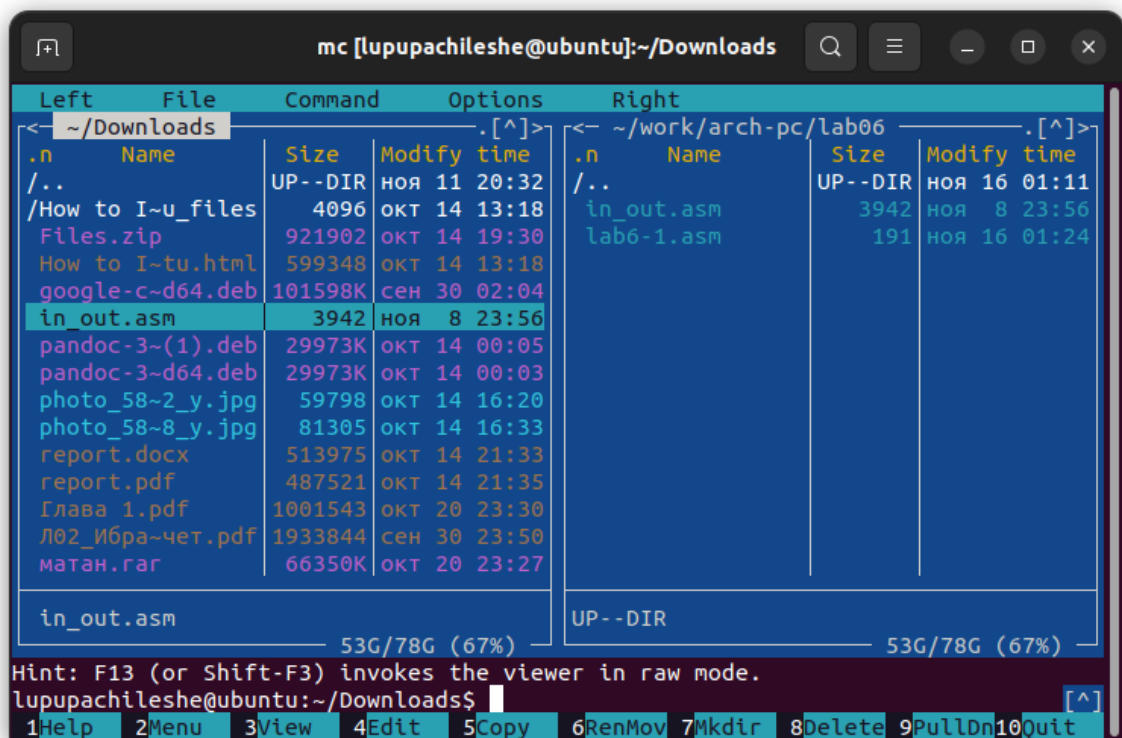


Рис. 2.3: листинга 6.1

__КОММЕНТАРИЙ:__ В файл, который я сделал, я вставил текст из листинга 6.1. Мне ну

Создайте исполняемый файл и запустите его.

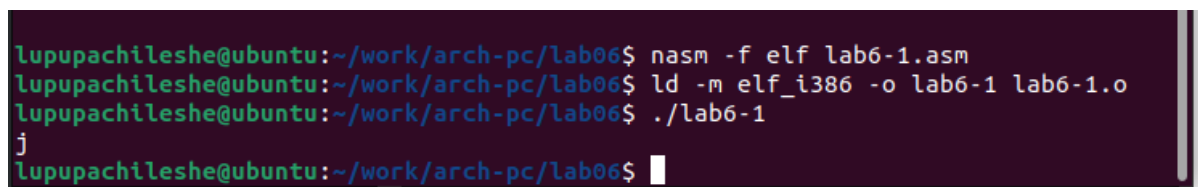


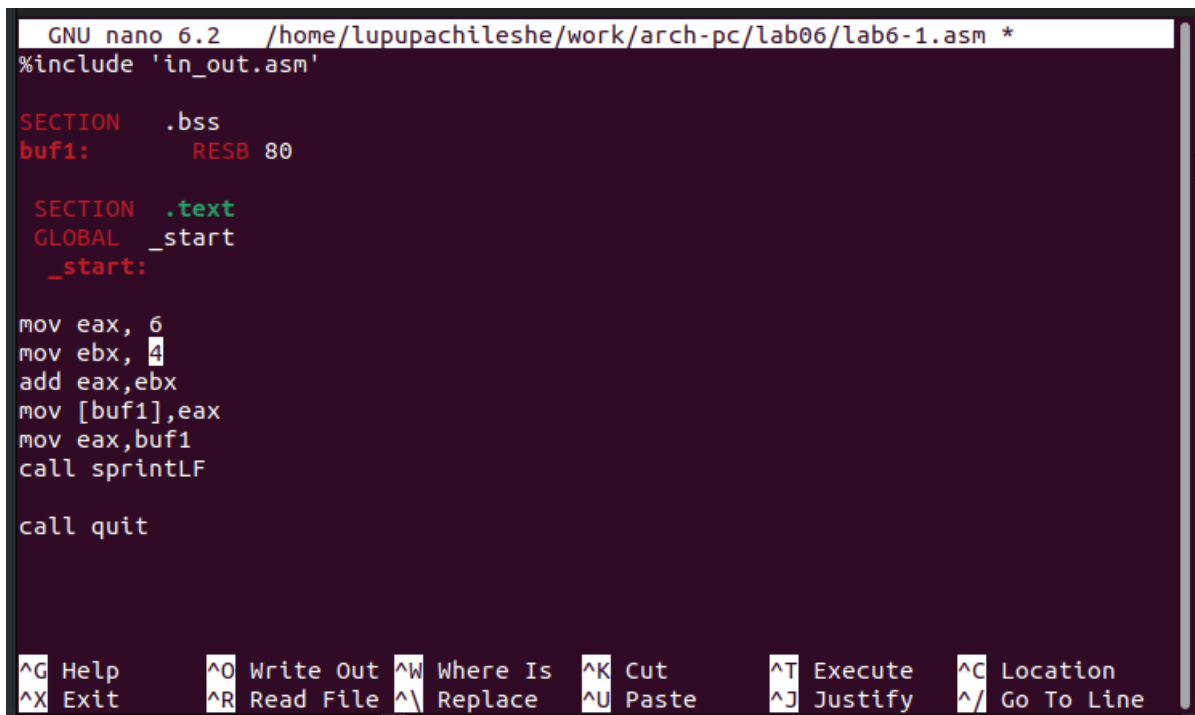
Рис. 2.4: исполняемый файл

- Далее изменим текст программы и вместо символов, запишем в регистры числа. Исправьте текст программы (Листинг 6.1) следующим образом: замените строки

```
mov eax,'6' mov ebx,'4'
```

на строки


```
mov eax,6 mov ebx,4
```



```
GNU nano 6.2 /home/lupupachileshe/work/arch-pc/lab06/lab6-1.asm *
#include 'in_out.asm'

SECTION .bss
buf1:    RESB 80

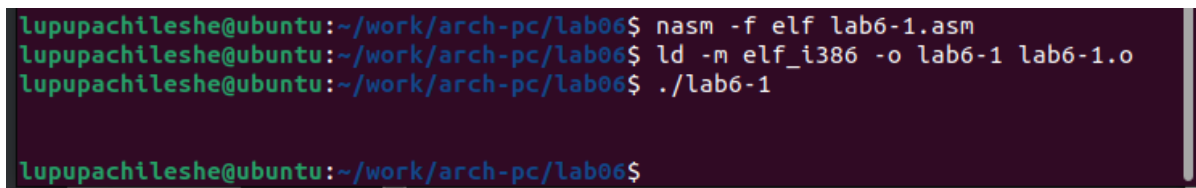
SECTION .text
GLOBAL _start
_start:

mov eax, 6
mov ebx, 4
add eax,ebx
mov [buf1],eax
mov eax,buf1
call sprintLF

call quit

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

Создайте исполняемый файл и запустите его.



```
lupupachileshe@ubuntu:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
lupupachileshe@ubuntu:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
lupupachileshe@ubuntu:~/work/arch-pc/lab06$ ./lab6-1

lupupachileshe@ubuntu:~/work/arch-pc/lab06$
```

Рис. 2.4: исполняемый файл Листинг 6.1

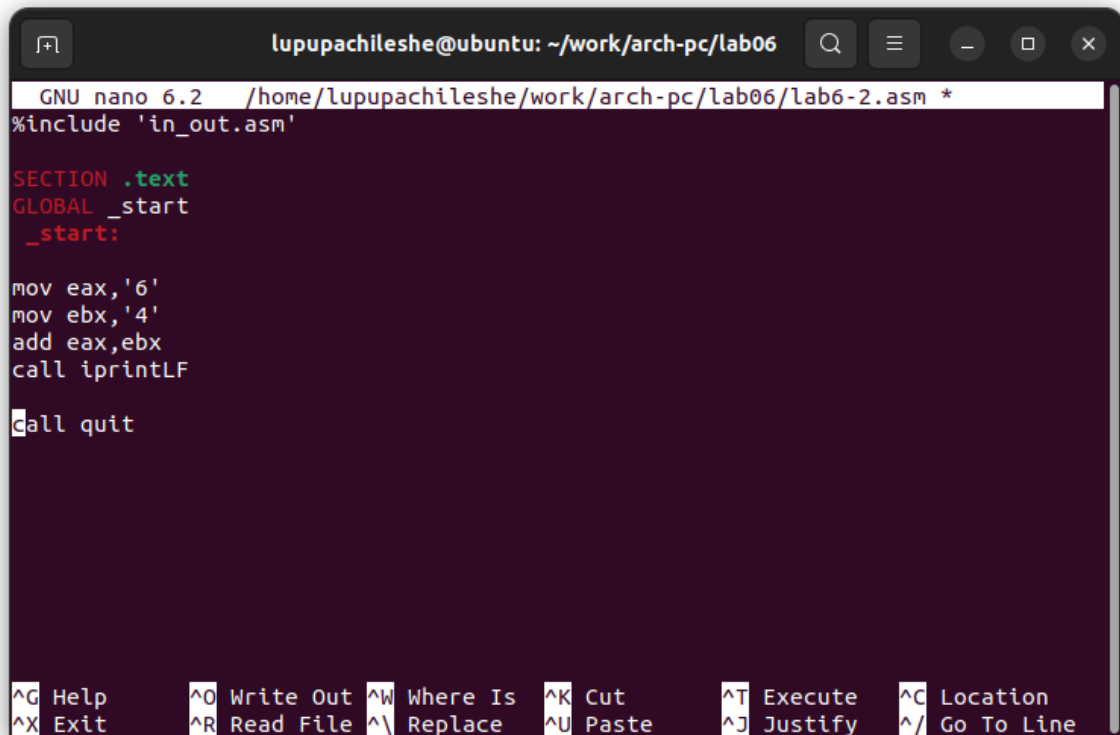
КОММЕНТАРИЙ: Изменил текст в программе на `mov eax,6 mov ebx,4`.И создал исполняемый файл

4. Как отмечалось выше, для работы с числами в файле `in_out.asm` реализованы подпрограммы для преобразования ASCII символов в числа и обратно. Преобразуем текст программы из Листинга 6.1 с использованием этих функций.

Создайте файл `lab6-2.asm` в каталоге `~/work/arch-pc/lab06` и введите в него текст программы из листинга 6.2.

```
lupupachileshe@ubuntu:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-2.asm
lupupachileshe@ubuntu:~/work/arch-pc/lab06$
```

Рис. 2.5: lab6-2.asm



The screenshot shows a terminal window with the title bar "lupupachileshe@ubuntu: ~/work/arch-pc/lab06". The terminal is running the GNU nano 6.2 editor, editing the file "/home/lupupachileshe/work/arch-pc/lab06/lab6-2.asm". The content of the file is as follows:

```
%include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

mov eax,'6'
mov ebx,'4'
add eax,ebx
call iprintLF

call quit
```

At the bottom of the terminal, there is a status bar with various keyboard shortcuts: ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, ^X Exit, ^R Read File, ^\ Replace, ^U Paste, ^J Justify, and ^_ Go To Line.

Рис. 2.6: листинга 6.2

Создайте исполняемый файл и запустите его

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

Рис. 2.7: исполняемый файл листинга 6.2

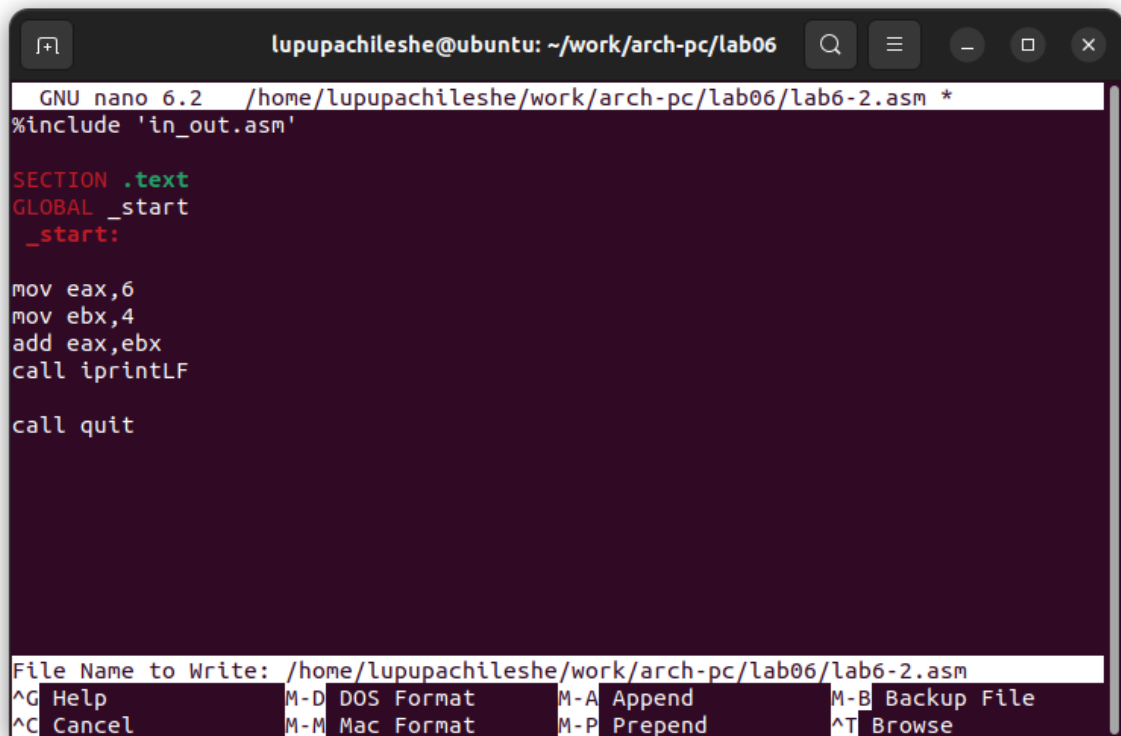
КОММЕНТАРИЙ: я создал файл под названием lab6-2 в каталоге lab06 и в него вставил текст из листинга 6.2.

5. Аналогично предыдущему примеру изменим символы на числа. Замените строки

```
mov eax,'6' mov ebx,'4'
```

на строки

```
mov eax,6 mov ebx,4
```



```
lupupachileshe@ubuntu: ~/work/arch-pc/lab06
GNU nano 6.2 /home/lupupachileshe/work/arch-pc/lab06/lab6-2.asm *
#include 'in_out.asm'

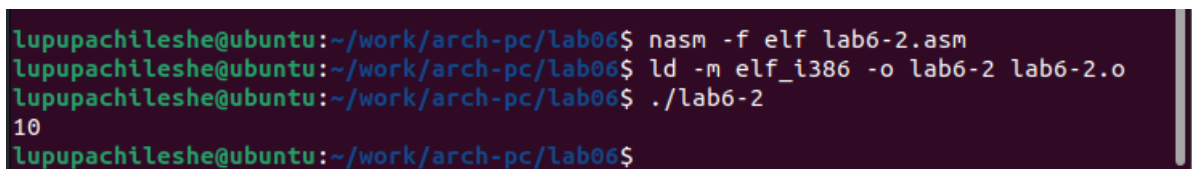
SECTION .text
GLOBAL _start
_start:

mov eax,6
mov ebx,4
add eax,ebx
call iprintLF

call quit

File Name to Write: /home/lupupachileshe/work/arch-pc/lab06/lab6-2.asm
^G Help      M-D DOS Format  M-A Append     M-B Backup File
^C Cancel    M-M Mac Format  M-P Prepend    ^T Browse
```

Замените функцию iprintLF на iprint. Создайте исполняемый файл и запустите его. Чем отличается вывод функций iprintLF и iprint?



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

Рис. 2.7: iprint

__КОММЕНТАРИЙ:__ Я поменял цифры на символы, сменил iprintLF на iprint. Далее я с

2.1 Выполнение арифметических операций в NASM

6. В качестве примера выполнения арифметических операций в NASM приведем программу вычисления арифметического выражения $\text{X}(\text{X}) = (5 \times 2 + 3)/3$.

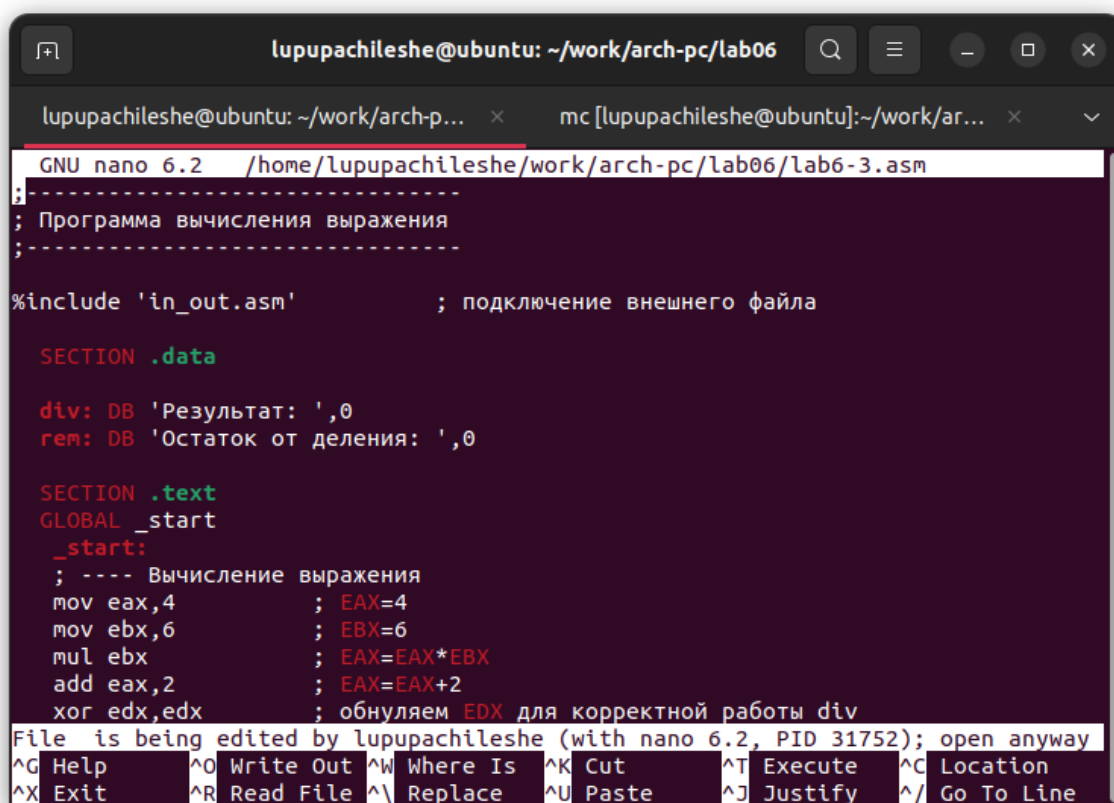
Создайте файл lab6-3.asm в каталоге ~/work/arch-pc/lab06:

`touch ~/work/arch-pc/lab06/lab6-3.asm`

```
lupupachileshe@ubuntu:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-3.asm
lupupachileshe@ubuntu:~/work/arch-pc/lab06$
```

Рис. 2.8: lab6-3.asm

Внимательно изучите текст программы из листинга 6.3 и введите в lab6-3.asm.



```
GNU nano 6.2 /home/lupupachileshe/work/arch-pc/lab06/lab6-3.asm
;-----
; Программа вычисления выражения
;-----

%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
File is being edited by lupupachileshe (with nano 6.2, PID 31752); open anyway
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

Рис. 2.9: листинга 6.3

Создайте исполняемый файл и запустите его. Результат работы программы должен быть следующим:

user@dk4n31:~\$./lab6-3 Результат: 4 Остаток от деления: 1 user@dk4n31:~\$

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

Рис. 2.10: исполняемый файл листинга 6.3

КОММЕНТАРИЙ: Я создал файл под названием lab6-3.asm и вставил в него текст из листинга 6.3. ДАЛЬШЕ мне пришлось создать исполняемый файл, чтобы увидеть результаты

Измените текст программы для вычисления выражения $\text{X}(\text{X}) = (4 \times 6 + 2)/5$. Создайте исполняемый файл и проверьте его работу.

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

Рис. 2.11: исполняемый файл 2

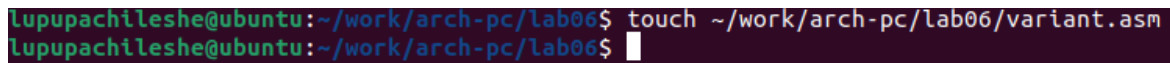
КОММЕНТАРИЙ: Я изменил текст программы для решения выражения

7. В качестве другого примера рассмотрим программу вычисления варианта задания по номеру студенческого билета, работающую по следующему алгоритму:

- вывести запрос на введение № студенческого билета
- вычислить номер варианта по формуле: $(\text{XX} \bmod 20) + 1$, где XX – номер студенческого билета (В данном случае $\text{X} \bmod \text{X}$ – это остаток от деления X на X).
- вывести на экран номер варианта.

Создайте файл variant.asm в каталоге ~/work/arch-pc/lab06:

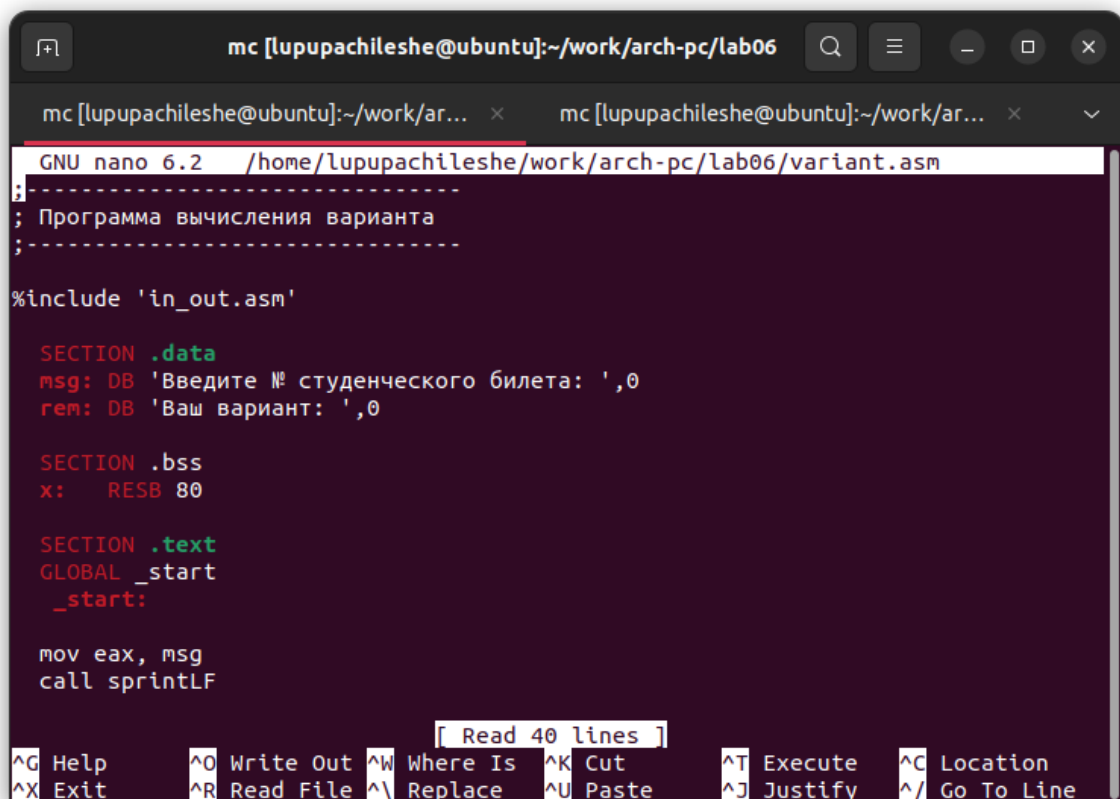
```
touch ~/work/arch-pc/lab06/variant.asm
```



```
lupupachileshe@ubuntu:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/variant.asm
lupupachileshe@ubuntu:~/work/arch-pc/lab06$
```

Рис. 2.12: variant.asm

Внимательно изучите текст программы из листинга 6.4 и введите в файл variant.asm



```
mc [lupupachileshe@ubuntu]:~/work/arch-pc/lab06
GNU nano 6.2 /home/lupupachileshe/work/arch-pc/lab06/variant.asm
;-----
; Программа вычисления варианта
;-----
#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
```

Рис. 2.13: листинга 6.4

Создайте исполняемый файл и запустите его. Проверьте результат работы программы вычислив номер варианта аналитически.

```

lupupachileshe@ubuntu:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/variant.asm
lupupachileshe@ubuntu:~/work/arch-pc/lab06$ mc

lupupachileshe@ubuntu:~/work/arch-pc/lab06$ nasm -f elf variant.asm
lupupachileshe@ubuntu:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
lupupachileshe@ubuntu:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1032225194
Ваш вариант: 15
lupupachileshe@ubuntu:~/work/arch-pc/lab06$ █

```

Рис. 2.14: исполняемый файл листинга 6.4

КОММЕНТАРИЙ: Я создал файл `variant.asm`. В этот файл я вставил текст из листинга 6.4. Далее я создал исполняемый файл

1. Какие строки листинга 6.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’

```

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

```

2. Для чего используются следующие инструкции?

```

mov ecx, x
mov edx, 80
call sread

```

`mov ecx, x`: Эта команда перемещает значение `x` в регистр `ECX`. Здесь `x` может бы

`mov edx, 80`: Эта команда перемещает значение `80` в регистр `EDX` 2.

`call sread`: Эта команда вызывает функцию `sread`. В контексте вызова функции, з

3. Для чего используется инструкция “call atoi”?

- Функция atoi в языке программирования C используется для преобразования строки

4. Какие строки листинга 6.4 отвечают за вычисления варианта?

```
xor edx,edx  
mov ebx,20  
div ebx  
inc edx
```

5. В какой регистр записывается остаток от деления при выполнении инструкции “div

- Разорвано в регистре eax

6. Для чего используется инструкция “inc edx”?

- В контексте inc edx edx – это 32-битный регистр общего назначения. Эта инструк

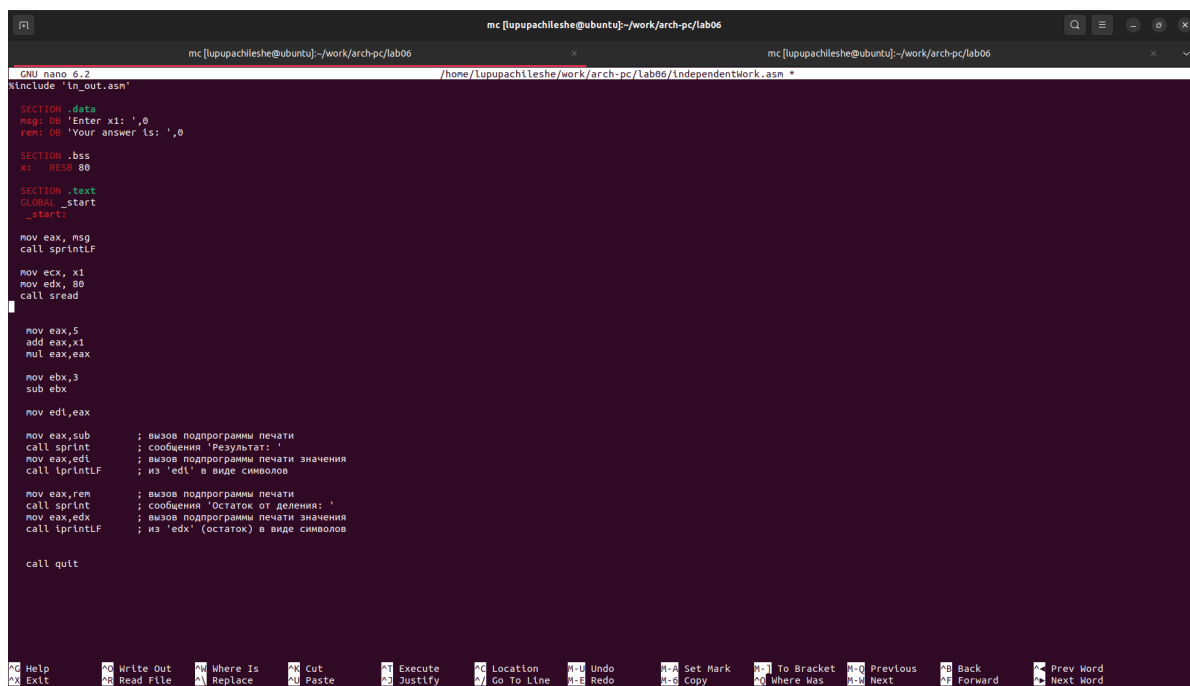
7. Какие строки листинга 6.4 отвечают за вывод на экран результата вычислений?

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


3 Задание для самостоятельной работы

Вариант 15

Написать программу вычисления выражения $x = x(x)$. Программа должна выводить выражение для вычисления, выводить запрос на ввод значения x , вычислять заданное выражение в зависимости от введенного x , выводить результат вычислений. Вид функции $x(x)$ выбрать из таблицы 6.3 вариантов заданий в соответствии с номером полученным при выполнении лабораторной работы. Создайте исполняемый файл и проверьте его работу для значений x_1 и x_2 из 6.3.



```
GNU nano 6.2 /home/lupupachleshe/work/arch-pc/lab06/independentWork.asm
#include "in_out.asm"

SECTION .data
msg: DB "Enter x1: ",0
ren: DB "Your answer is: ",0

SECTION .bss
x: RESB 80

SECTION .text
GLOBAL start
_start:

mov eax, msg
call sprintf

mov ecx, x1
mov edx, 80
call sread

mov eax, 5
add eax, x1
mul eax, eax

mov ebx, 3
sub ebx

mov edi, eax

mov eax, sub
call sprintf
; вызов подпрограммы печати
; сообщения "Результат: "
mov eax, edi
call iprintf
; из 'edi' в виде символов

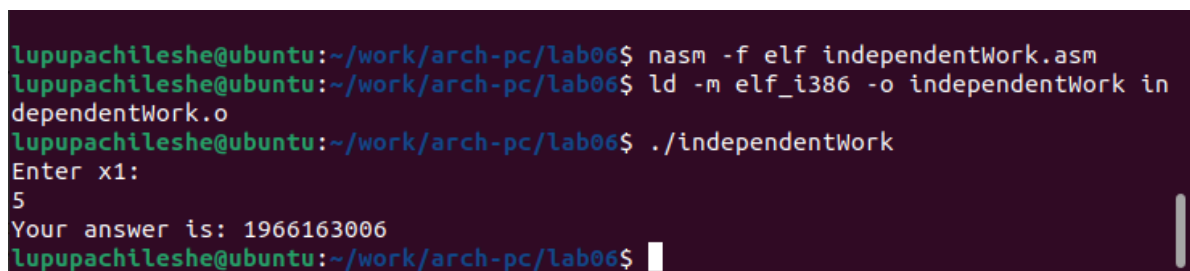
mov eax, ren
call sprintf
; вызов подпрограммы печати
; сообщения "Остаток от деления: "
mov eax, edx
call iprintf
; вызов подпрограммы печати значения
; из 'edx' (остаток) в виде символов

call quit
```

Рис. 3.1: Вариант 15

КОММЕНТАРИЙ: Создал файл под названием independentWork.asm В этом

файле я написал код, позволяющий решить выражение для варианта 15.

A terminal window with a dark purple background and green text. The user is at the prompt 'lupupachileshe@ubuntu:~/work/arch-pc/lab06\$'. They enter 'nasm -f elf independentWork.asm', then 'ld -m elf_i386 -o independentWork independentWork.o', and finally './independentWork'. The program prompts 'Enter x1:' and the user enters '5'. The program outputs 'Your answer is: 1966163006'. The prompt returns to 'lupupachileshe@ubuntu:~/work/arch-pc/lab06\$'.

```
lupupachileshe@ubuntu:~/work/arch-pc/lab06$ nasm -f elf independentWork.asm
lupupachileshe@ubuntu:~/work/arch-pc/lab06$ ld -m elf_i386 -o independentWork independentWork.o
lupupachileshe@ubuntu:~/work/arch-pc/lab06$ ./independentWork
Enter x1:
5
Your answer is: 1966163006
lupupachileshe@ubuntu:~/work/arch-pc/lab06$
```

Рис. 3.2: исполняемый файл

КОММЕНТАРИЙ: Создал исполняемый файл.

4 Выводы

Я приобрел навыки арифметических инструкций на языке ассемблера `nasm`.