

# РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

## ОТЧЕТ

### ПО ЛАБОРАТОРНОЙ РАБОТЕ № 7

дисциплина: Архитектура компьютера

Студент: Фелипе Паласиос

Группа: НКАбд-04-22\_

МОСКВ

2022 г.

## Содержание

1. Цель работы.....	3
2. Задание.....	3
3. Выполнение лабораторной работы.....	3
3.1 Символьные и численные данные в NASM.....	3
Выполнение арифметических операций в NASM.....	10
3.2 Задание для самостоятельной работы.....	14
4. Результаты.....	16

## 1. Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

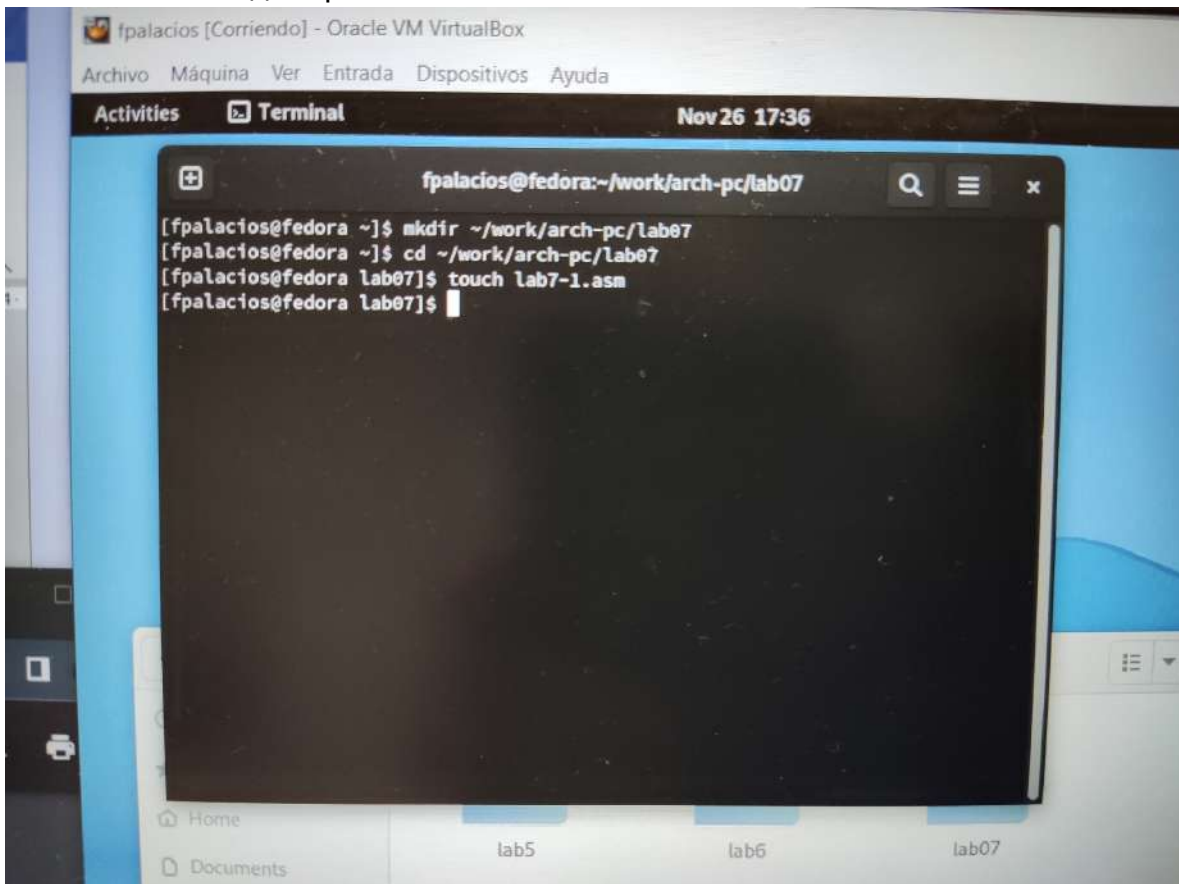
## 2. Задание

1. Работа с символьными и численными данными в NASM
2. Отработка на практике арифметических операций в NASM
3. Написать программу вычисления выражения с входными данными

## 3. Выполнение лабораторной работы

### 3.1 Символьные и численные данные в NASM

1. Создал каталог для программ лабораторной работы № 7, перешел в него и создал файл lab7-1.asm.



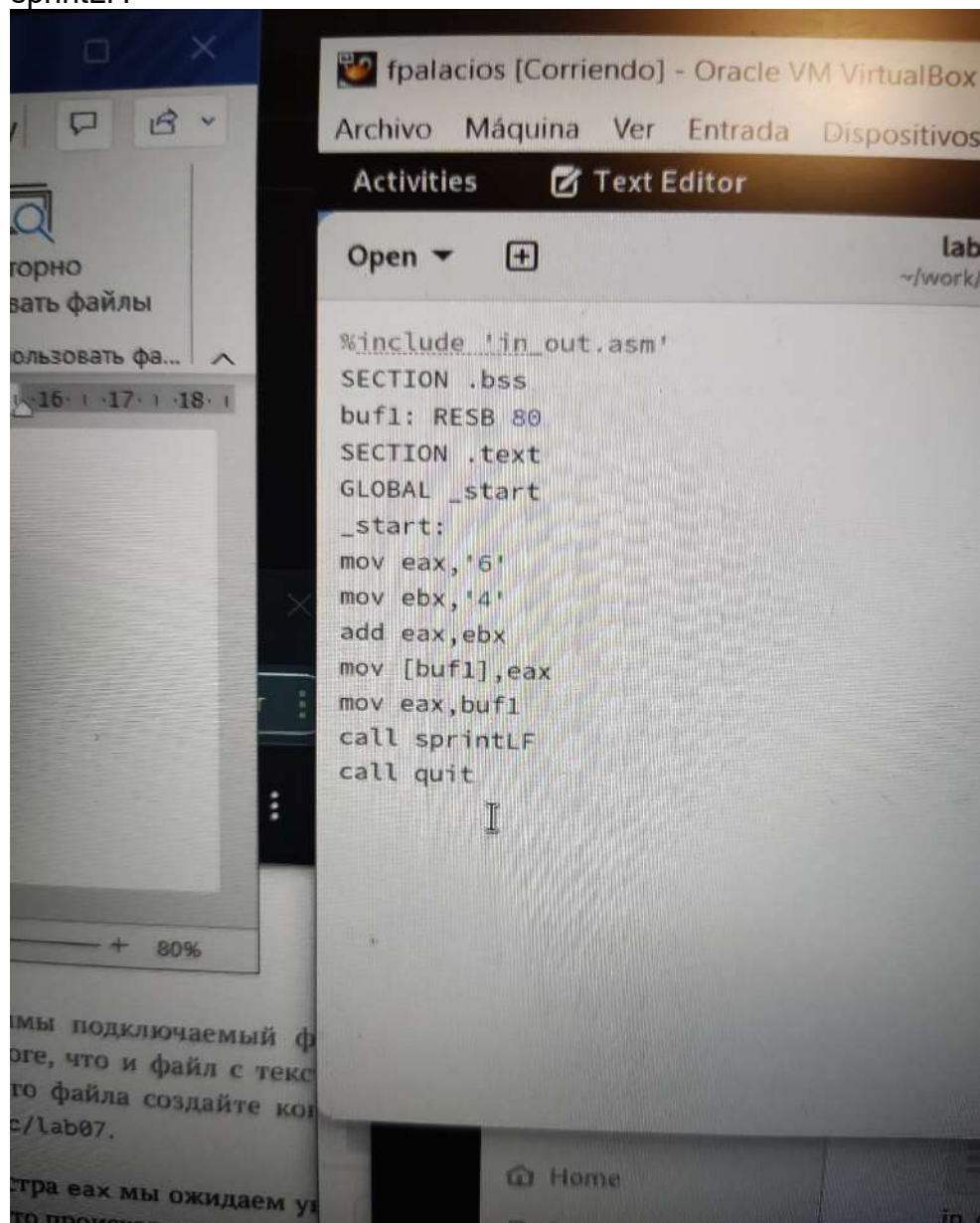
The screenshot shows a terminal window titled "fpalacios [Corriendo] - Oracle VM VirtualBox". The terminal output is as follows:

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

The terminal window is part of a desktop environment with a blue background. The top bar shows "Activities" and "Terminal" with the date and time "Nov 26 17:36". The bottom of the screen shows a dock with icons for "Home", "Documents", and three tabs labeled "lab5", "lab6", and "lab07".

2. Рассмотрели примеры программ вывода символьных и численных значений. Программы будут выводить значения записанные в регистр

eax. Ввели в файл lab7-1.asm текст программы из листинга 7.1. (рис. [-@fig:002]) В данной программе в регистр eax записывается символ 6 (mov eax,'6'), в регистр ebx символ 4 (mov ebx,'4'). Далее к значению в регистре eax прибавляется значение регистра ebx (add eax,ebx, результат сложения запишется в регистр eax). Далее выводится результат. Так как для работы функции sprintf в регистр eax должен быть записан адрес, необходимо использовать дополнительную переменную. Для этого записали значение регистра eax в переменную buf1 (mov [buf1],eax), а затем записали адрес переменной buf1 в регистр eax (mov eax,buf1) и вызвали функцию sprintf.



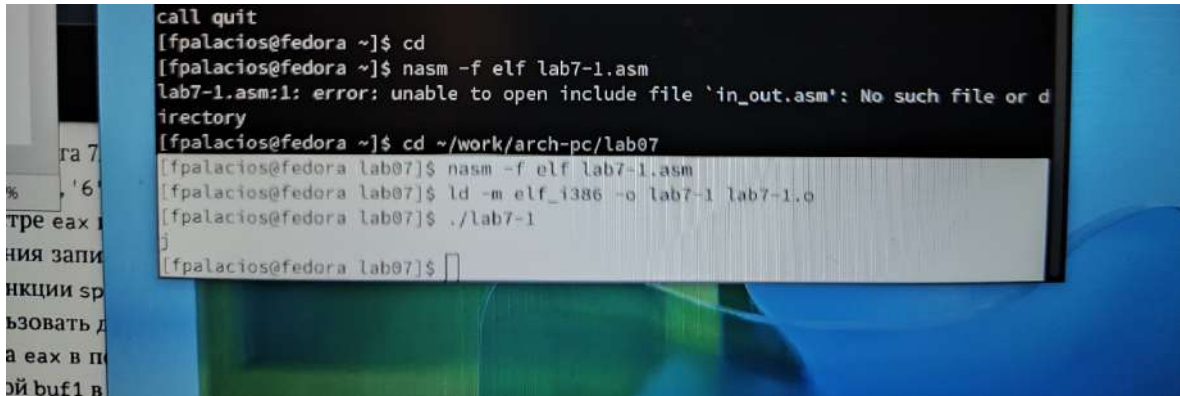
The screenshot shows a text editor window titled "fpalacios [Corriendo] - Oracle VM VirtualBox". The window has a menu bar with "Archivo", "Máquina", "Ver", "Entrada", and "Dispositivos". Below the menu bar is a toolbar with "Activities" and "Text Editor". The text editor shows the following assembly code:

```
%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
```

The code is displayed in a monospaced font. The window title bar includes the name "fpalacios [Corriendo]" and the application "Oracle VM VirtualBox". The menu bar and toolbar are visible above the code area. The code itself is for an assembly program that includes "in\_out.asm", defines a buffer "buf1" of size 80 in the .bss section, and implements a \_start function that moves the characters '6' and '4' into registers eax and ebx, adds them, stores the result in buf1, and then calls sprintf and quit.

### Текст программы из листинга 7.1

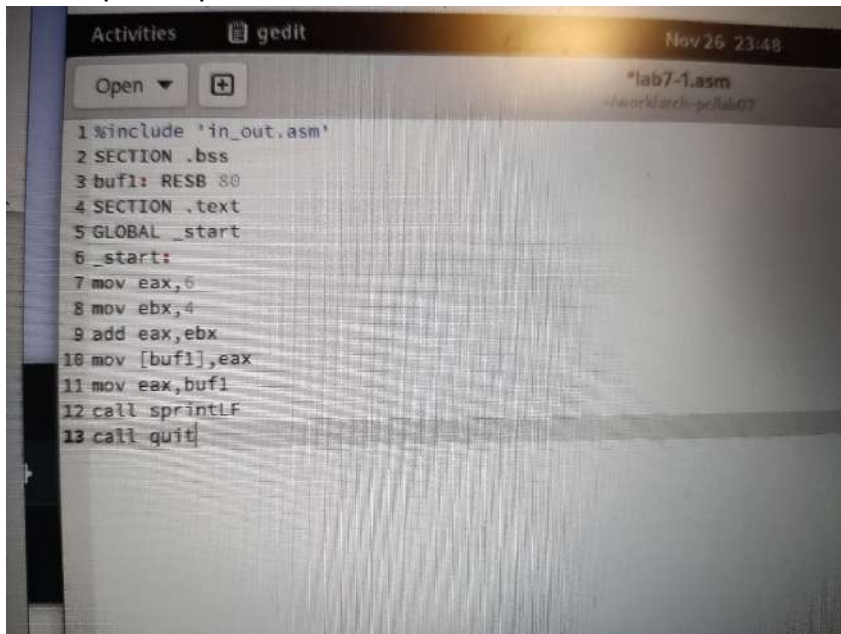
Создали исполняемый файл и запустили его.

A terminal window on a Fedora system. The user is in the directory ~/work/arch-pc/lab07. They run 'nasm -f elf lab7-1.asm' which fails with an error: 'unable to open include file 'in\_out.asm': No such file or directory'. Then they run 'ld -m elf\_i386 -o lab7-1 lab7-1.o' which succeeds. Finally, they run './lab7-1' which also succeeds. The prompt is [fpalacios@fedora lab07]\$.

```
call quit
[fpalacios@fedora ~]$ cd
[fpalacios@fedora ~]$ nasm -f elf lab7-1.asm
lab7-1.asm:1: error: unable to open include file 'in_out.asm': No such file or directory
[fpalacios@fedora ~]$ cd ~/work/arch-pc/lab07
[fpalacios@fedora lab07]$ nasm -f elf lab7-1.asm
[fpalacios@fedora lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[fpalacios@fedora lab07]$ ./lab7-1
[fpalacios@fedora lab07]$
```

### Исполняемый файл

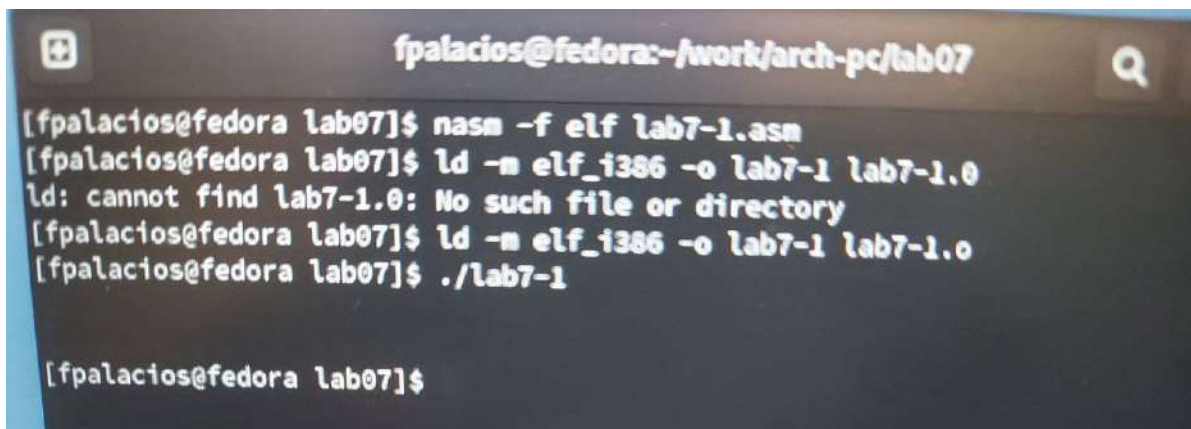
3. Далее изменили текст программы и вместо символов, записали в регистры числа.

A screenshot of a text editor (gedit) showing the assembly code for lab7-1.asm. The code includes a .bss section for a buffer, a .text section with a global \_start, and assembly instructions to move values into registers, add them, and call printf and \_exit. The editor window title is 'lab7-1.asm' and the path is '/work/arch-pc/lab07'. The date and time 'Nov 26 23:48' are shown in the top right corner.

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

### Текст программы

Создали исполняемый файл и запустили его.

A terminal window with a dark background and light-colored text. The title bar at the top shows a window icon, the username 'fpalacios@fedora', the path '-/work/arch-pc/lab07', and a search icon. The terminal contains the following commands and output:

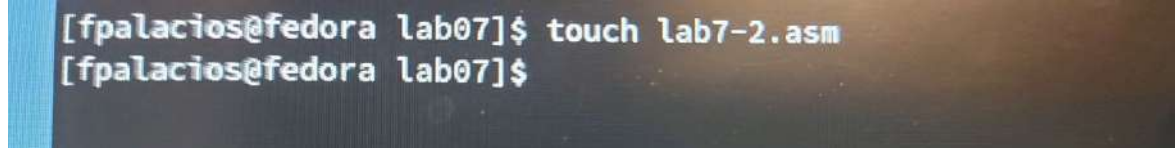
```
[fpalacios@fedora lab07]$ nasm -f elf lab7-1.asm
[fpalacios@fedora lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
ld: cannot find lab7-1.o: No such file or directory
[fpalacios@fedora lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[fpalacios@fedora lab07]$ ./lab7-1

[fpalacios@fedora lab07]$
```

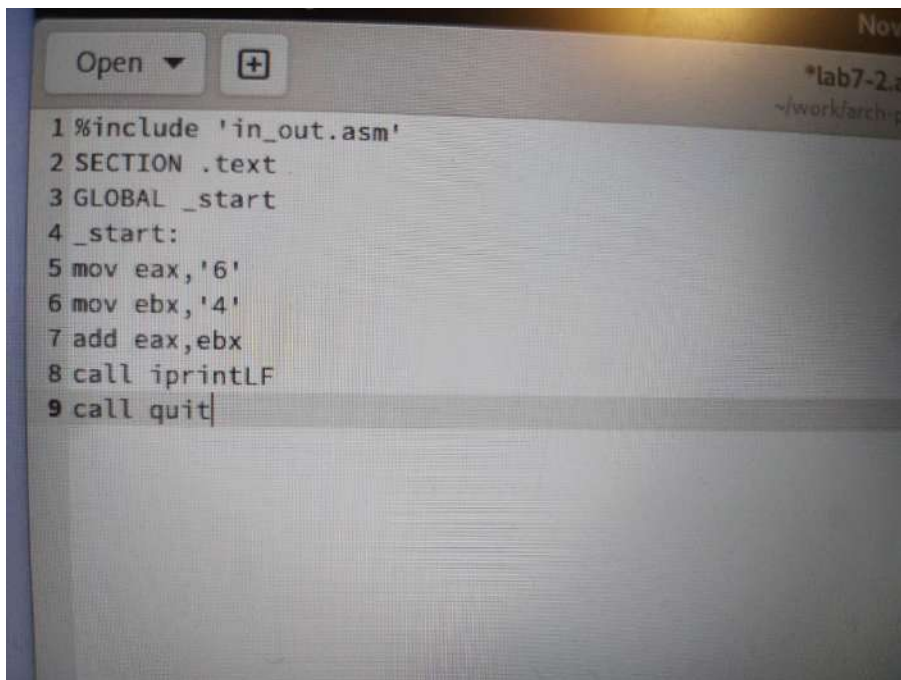
### *Исполняемый файл*

Как и в предыдущем случае при исполнении программы мы не получим число 10. В данном случае выводится символ с кодом 10. Пользуясь таблицей ASCII определили, что код 10 соответствует символ /n. Это символ перевода строки, он не отображается.

4. Как отмечалось выше, для работы с числами в файле in\_out.asm реализованы подпрограммы для преобразования ASCII символов в числа и обратно. Преобразовали текст программы из Листинга 7.1 с использованием этих функций. Создали файл lab7-2.asm в каталоге ~/work/arch-pc/lab07 и ввели в него текст программы из листинга 7.2.

A terminal window showing the creation of a new file. The terminal contains the following commands:

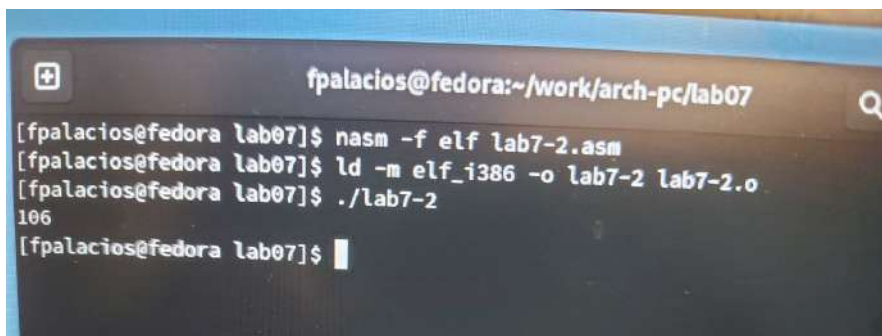
```
[fpalacios@fedora lab07]$ touch lab7-2.asm
[fpalacios@fedora lab07]$
```



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

*Текст программы из листинга 7.2*

Создали исполняемый файл и запустили его.



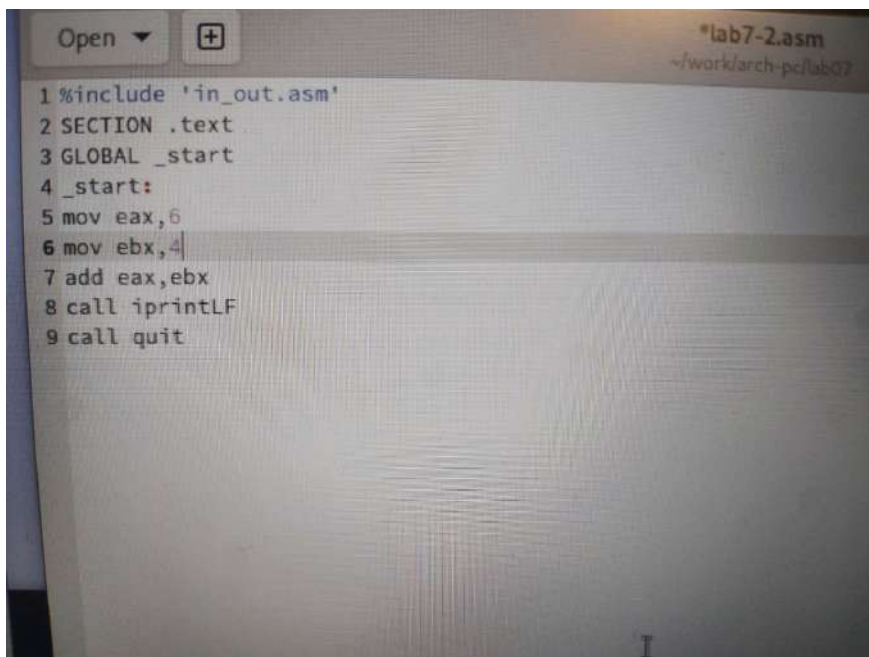
```
fpalacios@fedora:~/work/arch-pc/lab07
[fpalacios@fedora lab07]$ nasm -f elf lab7-2.asm
[fpalacios@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[fpalacios@fedora lab07]$ ./lab7-2
106
[fpalacios@fedora lab07]$
```

*Исполняемый файл*

В результате работы программы мы получили число 106. В данном случае, как и в первом, команда `add` складывает коды символов '6' и '4' ( $54+52=106$ ). Однако, в отличие от программы из листинга 7.1, функция `iprintLF` позволяет вывести число, а не символ, кодом которого является это число.

5. Аналогично предыдущему примеру изменим символы на числа.

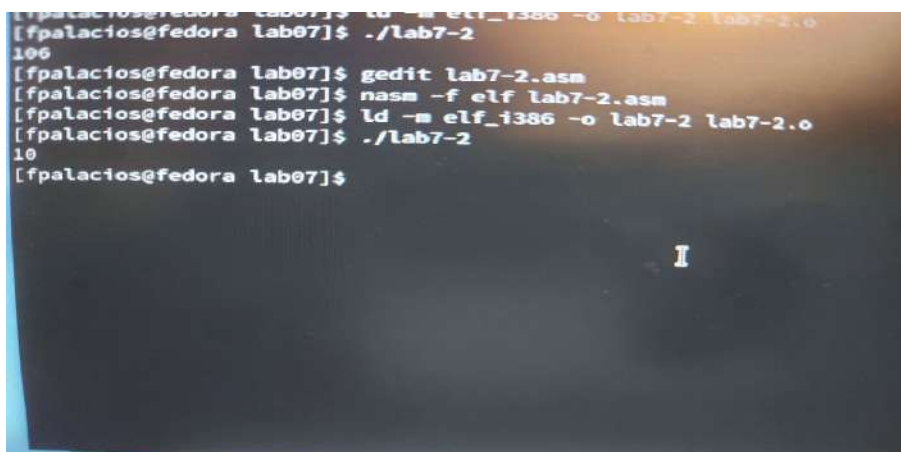


A screenshot of a text editor window titled '\*lab7-2.asm' with a path '~/.work/arch-pc/lab07'. The code is as follows:

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

### *Изменённый код*

Создайте исполняемый файл и запустите его. В результате при исполнении программы получили 10.

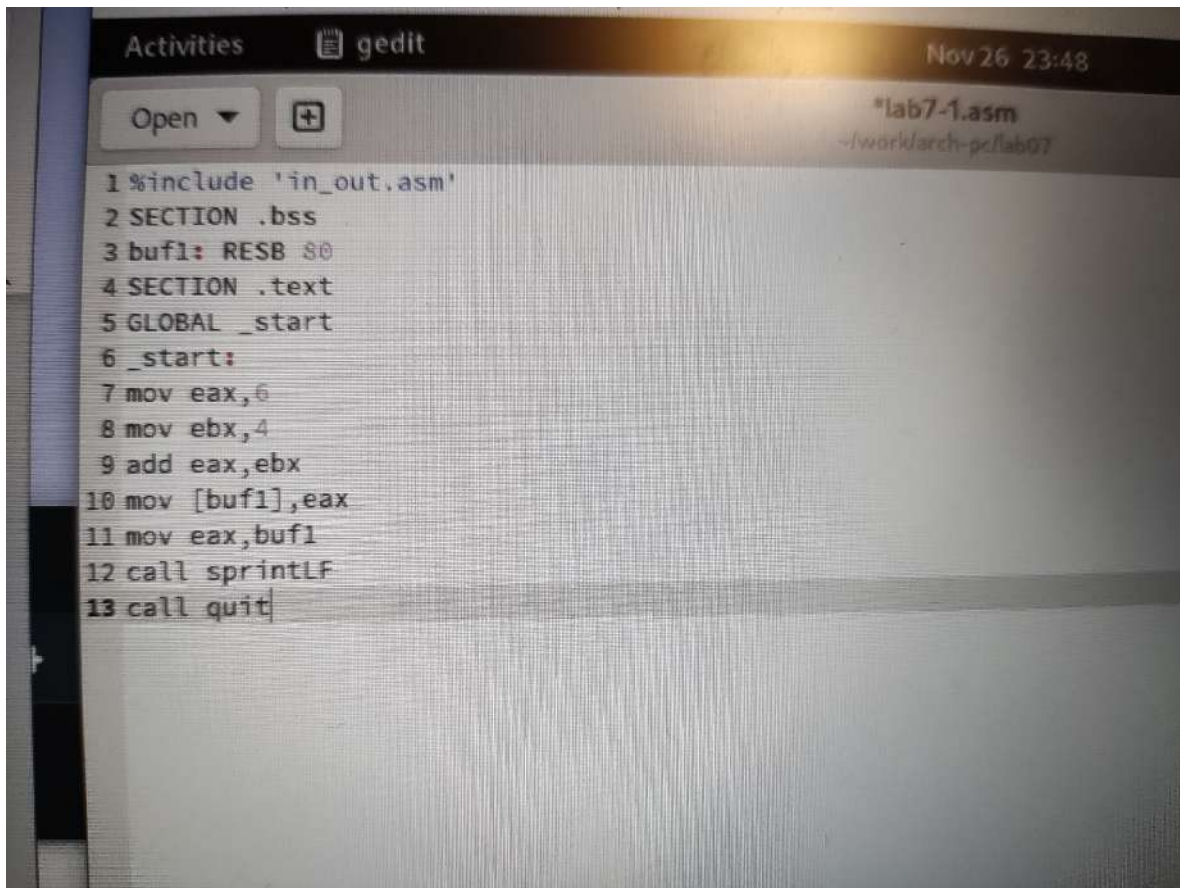
A screenshot of a terminal window showing the following commands and output:

```
[fpalacios@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[fpalacios@fedora lab07]$ ./lab7-2
106
[fpalacios@fedora lab07]$ gedit lab7-2.asm
[fpalacios@fedora lab07]$ nasm -f elf lab7-2.asm
[fpalacios@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[fpalacios@fedora lab07]$ ./lab7-2
10
[fpalacios@fedora lab07]$
```

### *Результат программы*

Заменили функцию `iprintLF` на `iprint`. Создали исполняемый файл и запустили его. Вывод функций `iprintLF` и `iprint` отличается наличием перевода строки после вывода?

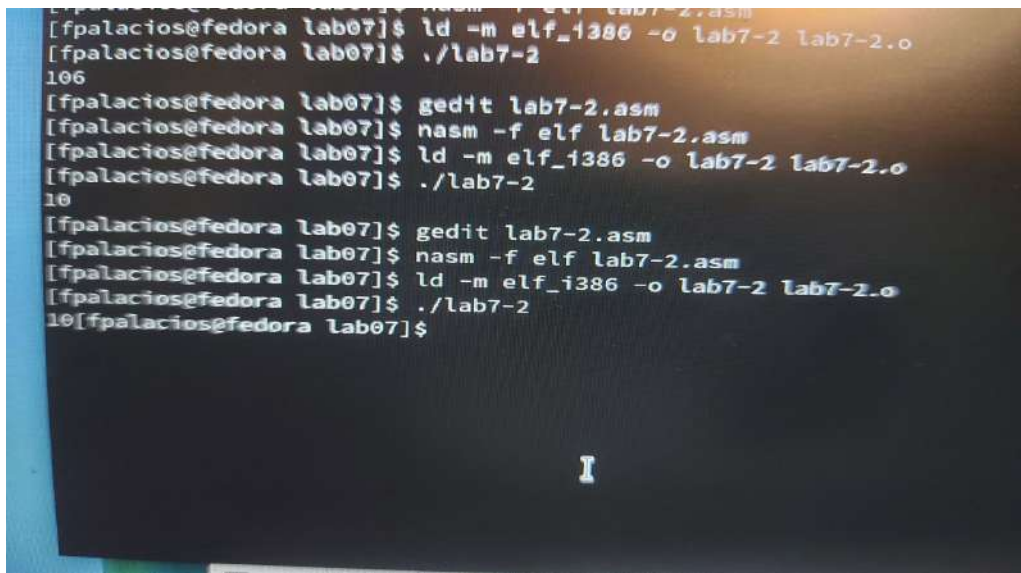




The screenshot shows a gedit editor window with the title bar "Activities gedit" and a timestamp "Nov 26 23:48". The file being edited is "lab7-1.asm" located at "~/.workMarch-pc/lab07". The code in the editor is as follows:

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

*iprintLF на iprint*



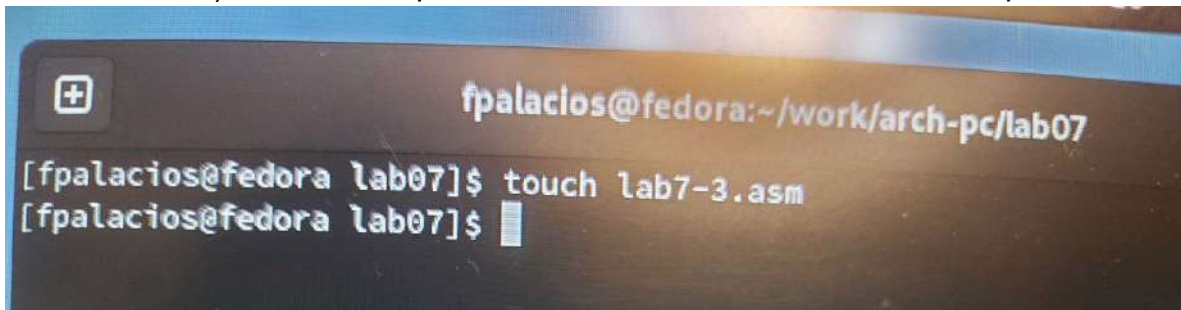
The screenshot shows a terminal window with the following commands and output:

```
[fpalacios@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[fpalacios@fedora lab07]$ ./lab7-2
106
[fpalacios@fedora lab07]$ gedit lab7-2.asm
[fpalacios@fedora lab07]$ nasm -f elf lab7-2.asm
[fpalacios@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[fpalacios@fedora lab07]$ ./lab7-2
10
[fpalacios@fedora lab07]$ gedit lab7-2.asm
[fpalacios@fedora lab07]$ nasm -f elf lab7-2.asm
[fpalacios@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[fpalacios@fedora lab07]$ ./lab7-2
10[fpalacios@fedora lab07]$
```

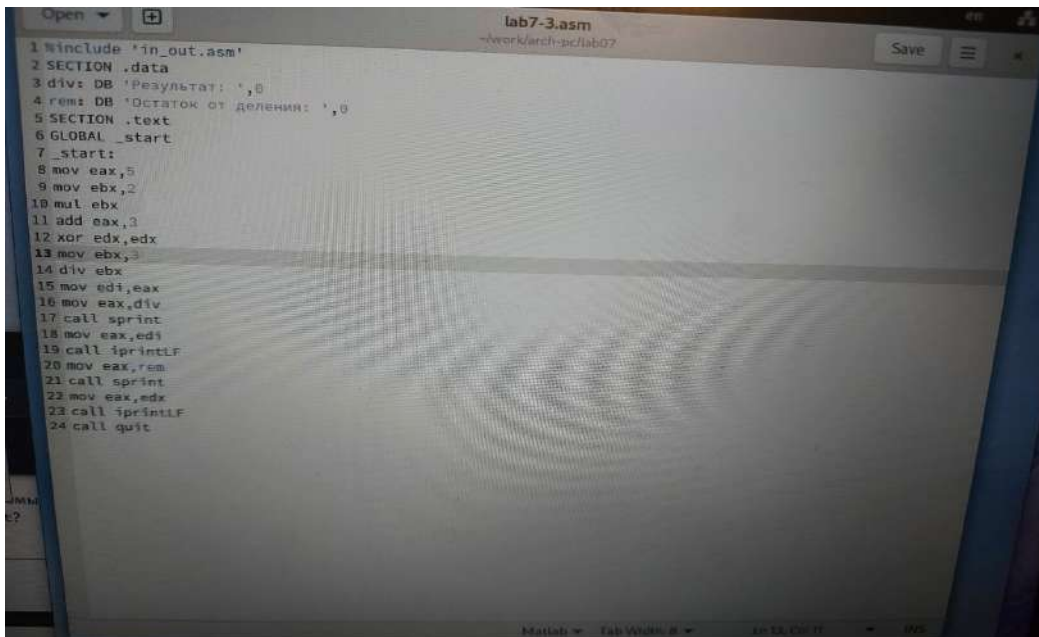
*Работа исполняемого файла*

## Выполнение арифметических операций в NASM

6. В качестве примера выполнения арифметических операций в NASM привели программу вычисления арифметического выражения  $f(x) = (5 * 2 + 3)/3$ . Создали файл lab7-3.asm в каталоге ~/work/arch-pc/lab07.



Внимательно изучили текст программы из листинга 7.3 и ввели в lab7-3.asm.



*Текст программы из листинга 7.3*

Создали исполняемый файл и запустили его. Получили следующий результат.

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

*Исполняемый файл*

Изменили текст программы для вычисления выражения  $f(x) = (4 * 6 + 2)/5$ .

```
*lab7-3.asm
#workdir: /home/palacios

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

*Изменённый текст программы*

Создали исполняемый файл и проверили его работу.

```
[fpalacios@fedora lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[fpalacios@fedora lab07]$ ./lab7-3
Результат: 4
Остаток от деления: 1
[fpalacios@fedora lab07]$ gedit
[fpalacios@fedora lab07]$ gedit lab7-3.asm
[fpalacios@fedora lab07]$ nasm -f elf lab7-3.asm
[fpalacios@fedora lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[fpalacios@fedora lab07]$ ./lab7-3
Результат: 5
Остаток от деления: 1
[fpalacios@fedora lab07]$
```

### Исполняемый файл

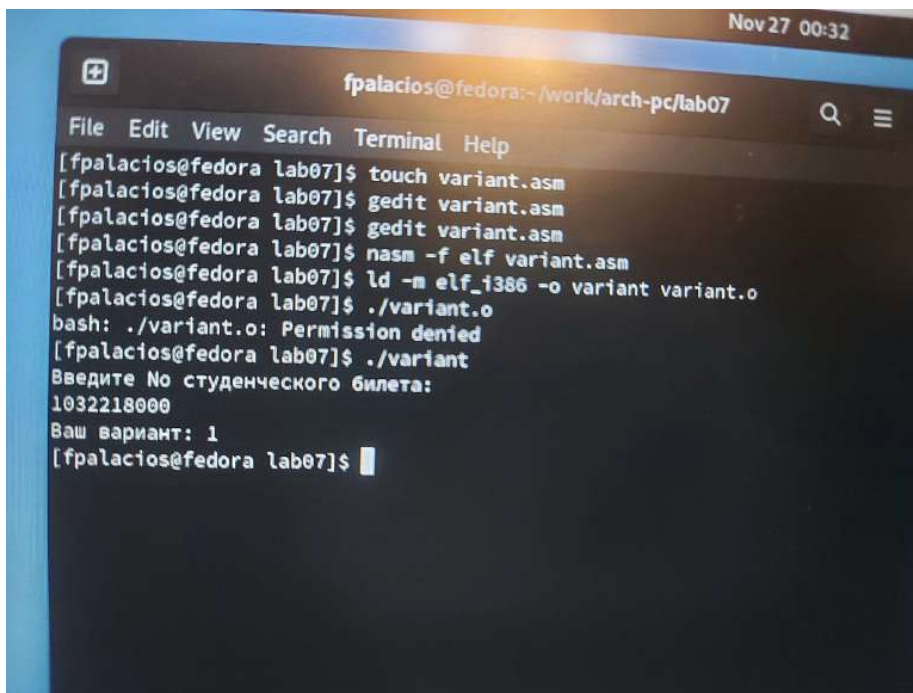
7. В качестве другого примера рассмотрели программу вычисления варианта задания по номеру студенческого билета, работающую по следующему алгоритму:

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

В данном случае число, над которым необходимо проводить арифметические операции, вводится с клавиатуры. Как отмечалось выше, ввод с клавиатуры осуществляется в символьном виде и для корректной работы арифметических операций в NASM символы преобразуются в числа. Для этого использована функция `atoi` из файла `in_out.asm`.

Создали файл `variant.asm` в каталоге `~/work/arch-pc/lab07`.



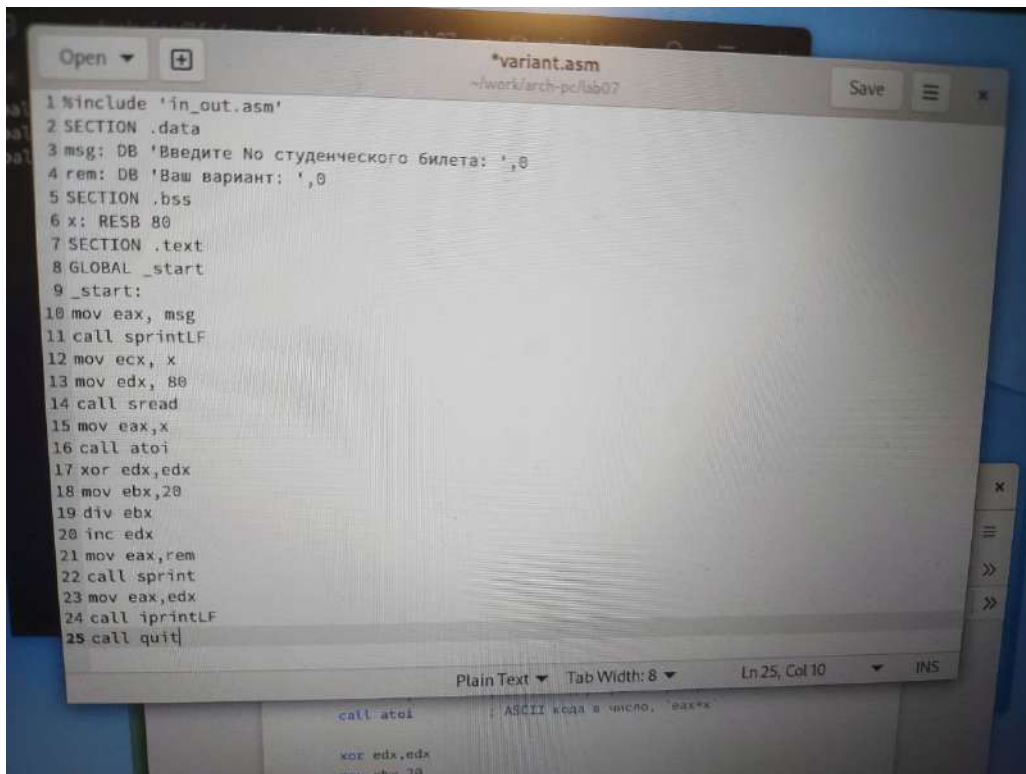


A terminal window titled 'fpalacios@fedora: ~/work/arch-pc/lab07' with a timestamp 'Nov 27 00:32'. The window contains a menu bar (File, Edit, View, Search, Terminal, Help) and a series of commands and their outputs. The commands executed are: 'touch variant.asm', 'gedit variant.asm', 'nasm -f elf variant.asm', and 'ld -m elf\_i386 -o variant variant.o'. The output shows the creation of 'variant.o' and an error message 'Permission denied' when trying to run './variant.o'. The user then runs './variant', which prompts for a student ID and a variant number. The user enters '1032218000' and '1' respectively.

```
fpalacios@fedora lab07]$ touch variant.asm
fpalacios@fedora lab07]$ gedit variant.asm
fpalacios@fedora lab07]$ nasm -f elf variant.asm
fpalacios@fedora lab07]$ ld -m elf_i386 -o variant variant.o
fpalacios@fedora lab07]$ ./variant.o
bash: ./variant.o: Permission denied
fpalacios@fedora lab07]$ ./variant
Введите No студенческого билета:
1032218000
Ваш вариант: 1
fpalacios@fedora lab07]$
```

Файл *variant.asm*

Внимательно изучили текст программы из листинга 7.4 и ввели в файл *variant.asm*.

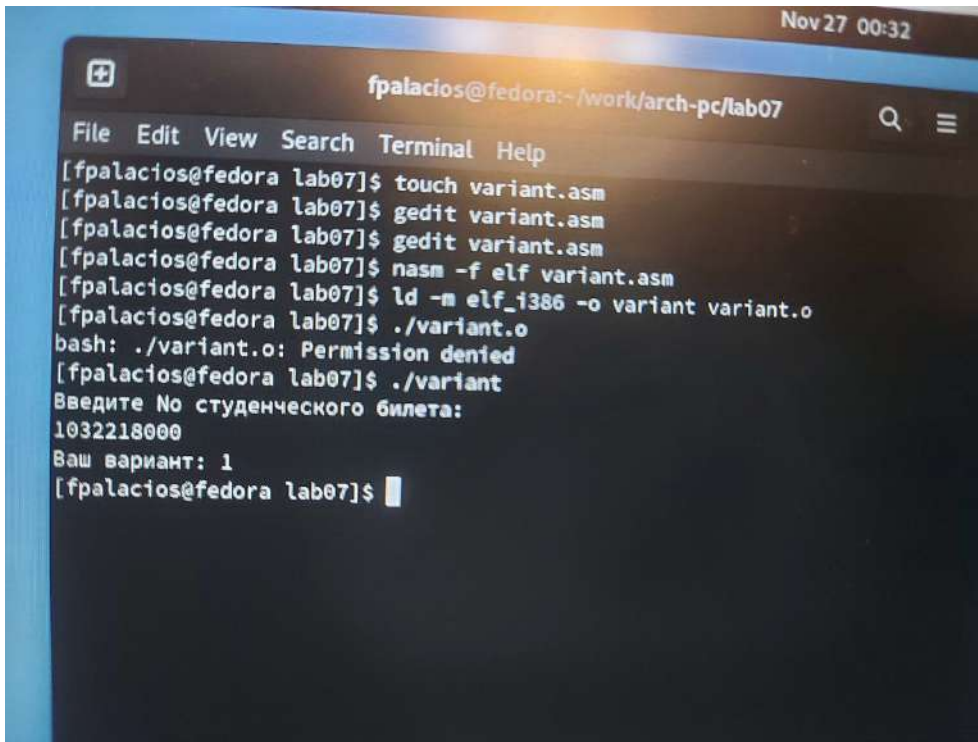


A text editor window titled '\*variant.asm' with a path '~/.work/arch-pc/lab07'. The window contains assembly code for a program that prompts for a student ID and a variant number. The code includes directives for sections (.data, .bss, .text), global symbols, and instructions for moving, printing, and calculating. The code is as follows:

```
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 sprintLF
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 sprint
23 mov eax, edx
24 call iprintLF
25 call quit
```

Текст программы из листинга 7.4

Создали исполняемый файл и запустили его. Проверили результат работы программы вычислив номер варианта аналитически.



```
Nov 27 00:32
fpalacios@fedora:~/work/arch-pc/lab07
File Edit View Search Terminal Help
[fpalacios@fedora lab07]$ touch variant.asm
[fpalacios@fedora lab07]$ gedit variant.asm
[fpalacios@fedora lab07]$ gedit variant.asm
[fpalacios@fedora lab07]$ nasm -f elf variant.asm
[fpalacios@fedora lab07]$ ld -m elf_i386 -o variant variant.o
[fpalacios@fedora lab07]$ ./variant.o
bash: ./variant.o: Permission denied
[fpalacios@fedora lab07]$ ./variant
Введите No студенческого билета:
1032218000
Ваш вариант: 1
[fpalacios@fedora lab07]$
```

### Работа исполняемого файла

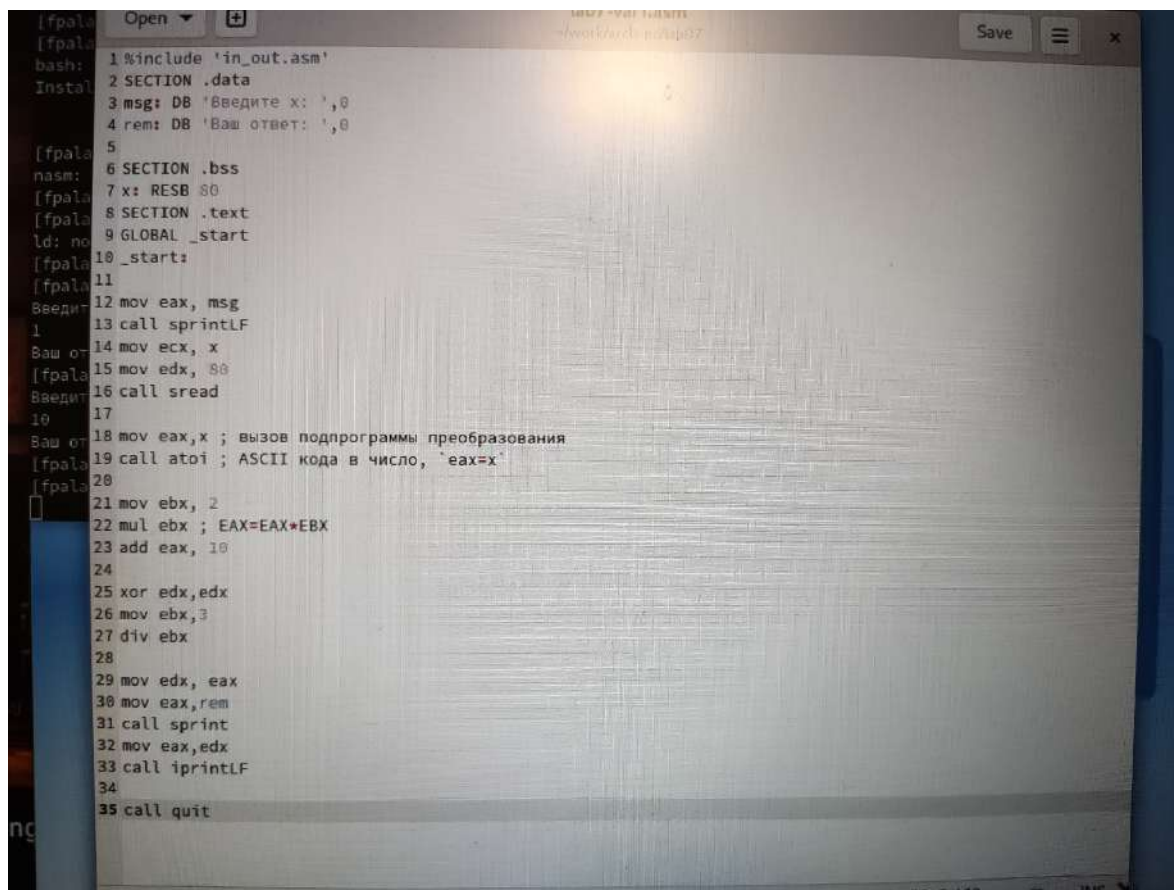
Ответы на вопросы лабораторной работы:

1. Какие строки листинга 7.4 отвечают за вывод на экран сообщения 'Ваш вариант: '? `mov eax,rem call sprint` 2. Для чего используются следующие инструкции? `mov ecx, x` - запись входной переменной в регистр `ecx`; `mov edx, 80` - запись размера переменной в регистр `edx`; `call sread` - вызов процедуры чтения данных; 3. Для чего используется инструкция "call atoi"? Вызов функции преобразующей `ascii`-код символа в целое число и записывающий результат в регистр `eax`. 4. Какие строки листинга 7.4 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 ebx"? В регистр `ebx`. 6. Для чего используется инструкция "inc edx"? Увеличивает значение `edx` на 1. 7. Какие строки листинга 7.4 отвечают за вывод на экран результата вычислений? `mov eax,rem call sprint mov eax,edx call iprintLF`

### 3.2 Задание для самостоятельной работы

Написали программу вычисления выражения  $y = f(x)$ . Программа выводит выражение для вычисления, выводит запрос на ввод значения  $x$ , вычисляет заданное выражение в зависимости от введенного  $x$ , выводит результат вычислений. Вид функции  $f(x)$  выбрали из таблицы 7.3 вариантов заданий/

Создали исполняемый файл и проверили его работу для значений  $x = 1$  и  $x = 10$  из 7.3.



```
1 %include 'in_out.asm'
2 SECTION .data
3 msg: DB 'Введите x: ',0
4 rem: DB 'Ваш ответ: ',0
5
6 SECTION .bss
7 x: RESB 80
8 SECTION .text
9 GLOBAL _start
10 _start:
11
12 mov eax, msg
13 call sprintf
14 mov ecx, x
15 mov edx, 80
16 call sread
17
18 mov eax, x ; вызов подпрограммы преобразования
19 call atoi ; ASCII кода в число, 'eax=x'
20
21 mov ebx, 2
22 mul ebx ; EAX=EAX*EBX
23 add eax, 10
24
25 xor edx, edx
26 mov ebx, 3
27 div ebx
28
29 mov edx, eax
30 mov eax, rem
31 call sprintf
32 mov eax, edx
33 call iprintLF
34
35 call quit
```

Файл *variant.asm*



The screenshot shows a terminal window titled 'fpalacios [Corriendo] - Oracle VM VirtualBox'. The window has a menu bar with 'Archivo', 'Máquina', 'Ver', 'Entrada', 'Dispositivos', and 'Ayuda'. Below the menu bar, there are tabs for 'Activities' and 'Terminal'. The terminal shows the following commands and output:

```
[fpalacios@fedora ~]$ cd
[fpalacios@fedora ~]$ cd ~/work/arch-pc/lab07
[fpalacios@fedora lab07]$ touch lab7-var1.asm
[fpalacios@fedora lab07]$ gedit lab7-var1.asm
[fpalacios@fedora lab07]$ nas -f elf lab-var1.asm
bash: nas: command not found...
Install package 'perl-Net-Appliance-Session' to provide command 'nas'? [N/y] n

[fpalacios@fedora lab07]$ nasm -f elf lab-var1.asm
nasm: fatal: unable to open input file `lab-var1.asm' No such file or directory
[fpalacios@fedora lab07]$ nasm -f elf lab7-var1.asm
[fpalacios@fedora lab07]$ ld -m elf_i386 -o lab7-var1.asm
ld: no input files
[fpalacios@fedora lab07]$ ld -m elf_i386 -o lab7-var1 lab7-var1.o
[fpalacios@fedora lab07]$ ./lab7-var1
Введите x:
1
Ваш ответ: 4
[fpalacios@fedora lab07]$ ./lab7-var1
Введите x:
10
Ваш ответ: 10
[fpalacios@fedora lab07]$
```

At the bottom of the terminal window, the text 'Архитектура ЭВМ' is visible.

*Работа исполняемого файла*

## 4. Результаты

В результате выполнения лабораторной работы были освоены арифметические инструкции языка ассемблера NASM.