

Шаблон отчёта по лабораторной работе

Простейший вариант

Ясиновская Дарья Олеговна

Содержание

| | |
|--------------------------------|----|
| Цель работы | 5 |
| Выполнение лабораторной работы | 6 |
| Ответы на вопросы | 17 |
| Самостоятельная работа | 18 |
| Листинги программ | 20 |
| Выводы | 22 |
| Список литературы | 23 |

Список иллюстраций

| | | |
|----|--|----|
| 1 | Создание каталога и файла в нем | 6 |
| 2 | Ввод текста | 7 |
| 3 | Создание и запуск исполняемого файла | 7 |
| 4 | Изменение текста | 8 |
| 5 | Создание и запуск исполняемого файла | 8 |
| 6 | Создание файла | 9 |
| 7 | Ввод текста | 9 |
| 8 | Создание и запуск исполняемого файла | 10 |
| 9 | Изменение текста | 10 |
| 10 | Создание и запуск исполняемого файла | 11 |
| 11 | Создание каталога и файла в нем | 11 |
| 12 | Создание файла в каталоге | 11 |
| 13 | Ввод текста | 12 |
| 14 | Создание и запуск исполняемого файла | 12 |
| 15 | Изменение текста | 13 |
| 16 | Создание и запуск исполняемого файла | 14 |
| 17 | Создание файла | 14 |
| 18 | Ввод текста | 15 |
| 19 | Создание и запуск исполняемого файла | 16 |
| 1 | Создание файла | 18 |
| 2 | Текст программы | 18 |
| 3 | Создание и запуск исполняемого файла | 19 |
| 4 | Создание и запуск исполняемого файла | 19 |

Список таблиц

Цель работы

Целью лабораторной работы является освоение арифметических инструкций языка ассемблера NASM

sh;@newham:2005:~\$ bash;@zarrelli:2017:~\$ bash;@robbins:2013:~\$ bash;@tannenbaum:arch-pc:ru;@tannenbaum:modern-os:ru].

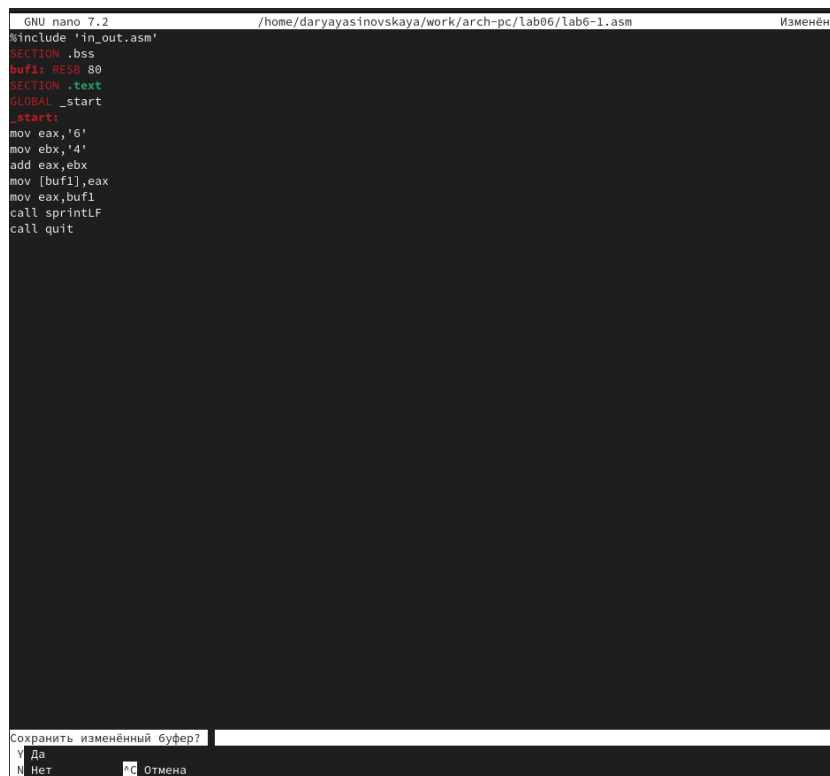
Выполнение лабораторной работы

Создала каталог для выполнения лабораторной работы №6, перешла в него и создала файл lab6-1.asm

```
[daryayasinovskaya@10 ~]$ mkdir ~/work/arch-pc/lab06  
[daryayasinovskaya@10 ~]$ cd ~/work/arch-pc/lab06  
[daryayasinovskaya@10 lab06]$ touch lab6-1.asm  
[daryayasinovskaya@10 lab06]$
```

Рис. 1: Создание каталога и файла в нем

Ввела в файл lab6-1.asm текст из листинга 6.1



```
GNU nano 7.2 /home/daryayasinovskaya/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
```

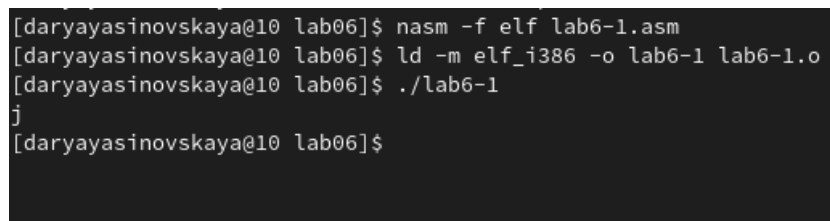
Сохранить изменённый буфер?

Y Да
N Нет

Отмена

Рис. 2: Ввод текста

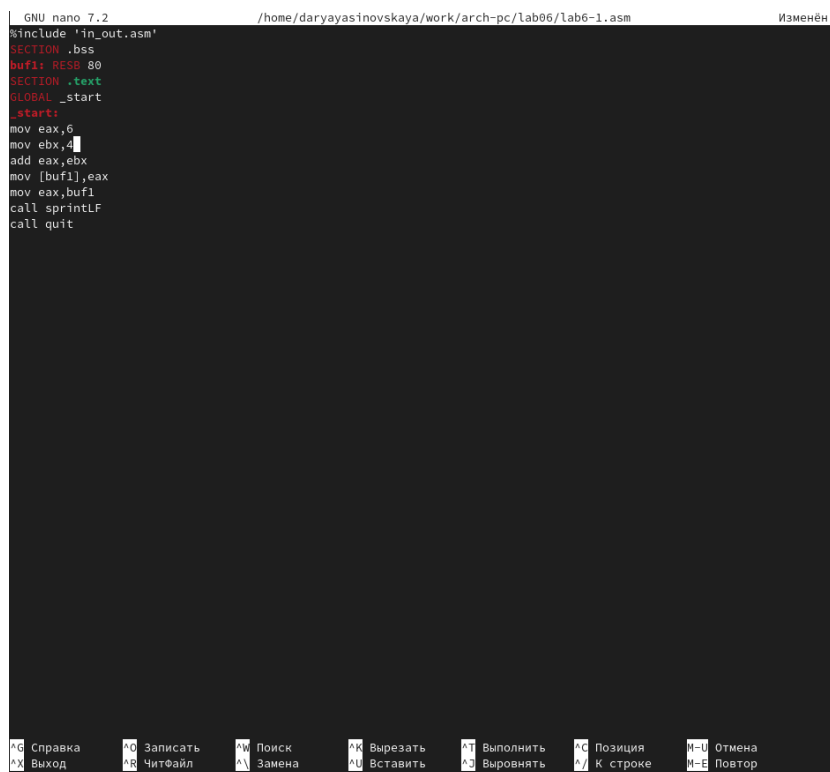
Создала исполняемый файл и запустила его. В выводе получила символ j.



```
[daryayasinovskaya@10 lab06]$ nasm -f elf lab6-1.asm
[daryayasinovskaya@10 lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[daryayasinovskaya@10 lab06]$ ./lab6-1
j
[daryayasinovskaya@10 lab06]$
```

Рис. 3: Создание и запуск исполняемого файла

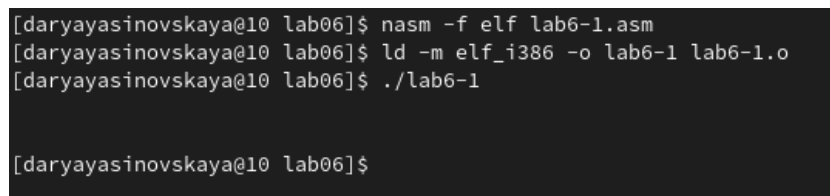
Изменила текст программы, заменив символы на числа



```
GNU nano 7.2 /home/daryayasinovskaya/work/arch-pc/lab06/lab6-1.asm Изменён
#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
```

Рис. 4: Изменение текста

Создала исполняемый файл и запустила его. Теперь вывелся символ с кодом 10, это символ перевода строки. Этот символ не отображается при выводе на экран.



```
[daryayasinovskaya@10 lab06]$ nasm -f elf lab6-1.asm
[daryayasinovskaya@10 lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[daryayasinovskaya@10 lab06]$ ./lab6-1

[daryayasinovskaya@10 lab06]$
```

Рис. 5: Создание и запуск исполняемого файла

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


```
[daryayasinovskaya@10 lab06]$ touch lab6-2.asm  
[daryayasinovskaya@10 lab06]$
```

Рис. 6: Создание файла

A screenshot of the nano text editor interface. The title bar at the top shows a plus icon, the username 'mc [daryayasinovskaya@f', and the file path '/home/daryayasinovskaya/'. The editor content shows assembly code: '%include 'in_out.asm'', 'SECTION .text' (in green), 'GLOBAL _start', and a label '_start:' followed by instructions 'mov eax, '6'', 'mov ebx, '4'', 'add eax, ebx', 'call iprint', and 'call quit' with a cursor at the end of the last line.

```
GNU nano 7.2 /home/daryayasinovskaya/  
%include 'in_out.asm'  
SECTION .text  
GLOBAL _start  
_start:  
mov eax, '6'  
mov ebx, '4'  
add eax, ebx  
call iprint  
call quit
```

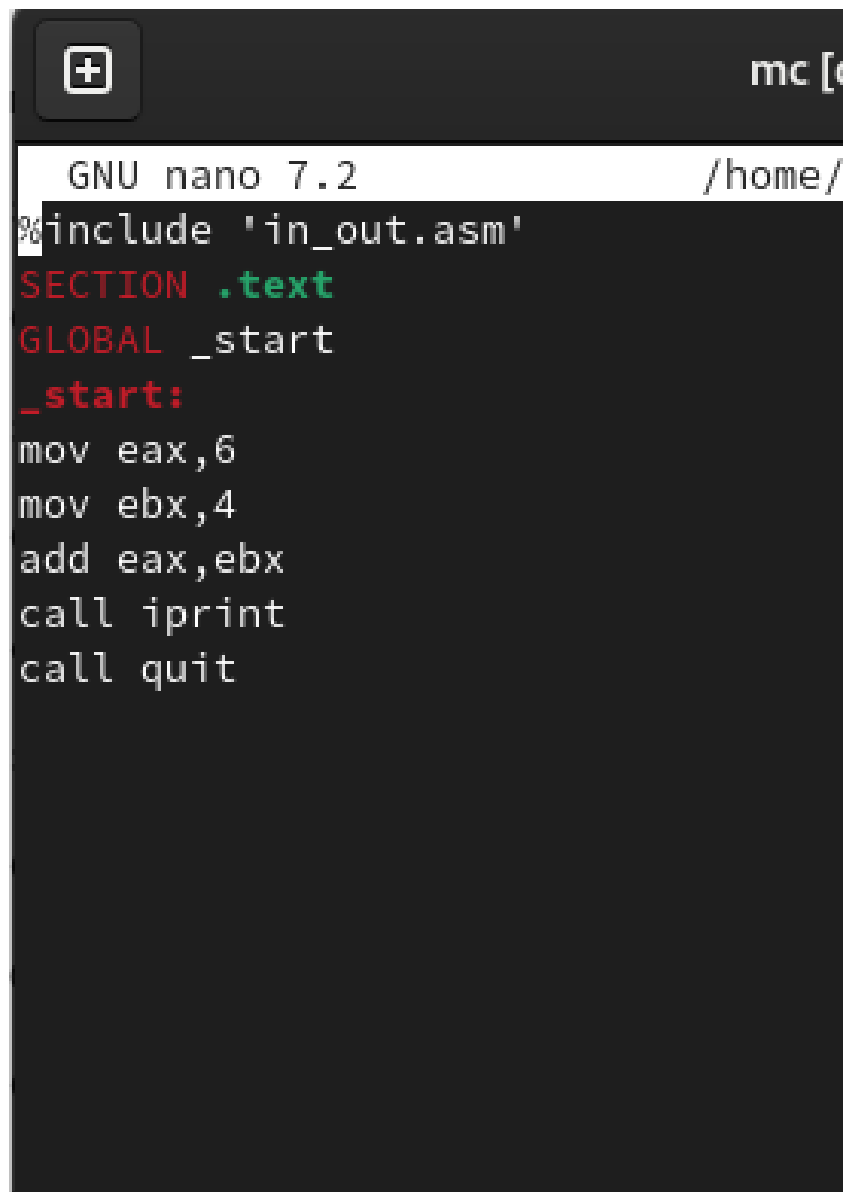
Рис. 7: Ввод текста

Создала исполняемый файл и запустила его. В выводе получила число 106

```
[daryayasinovskaya@10 lab06]$ nasm -f elf lab6-2.asm
[daryayasinovskaya@10 lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[daryayasinovskaya@10 lab06]$ ./lab6-2
106
[daryayasinovskaya@10 lab06]$
```

Рис. 8: Создание и запуск исполняемого файла

Изменила файл, заменив символы на числа.

A screenshot of the GNU nano 7.2 text editor interface. The window title is 'mc [c]'. The editor shows the following assembly code:

```
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprint
call quit
```

Рис. 9: Изменение текста

Создала исполняемый файл и запустила его. В результате выполнения программы получила число 10.

```
[daryayasinovskaya@10 lab06]$ nasm -f elf lab6-2.asm
[daryayasinovskaya@10 lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[daryayasinovskaya@10 lab06]$ ./lab6-2
10
[daryayasinovskaya@10 lab06]$
```

Рис. 10: Создание и запуск исполняемого файла

Заменяла функцию `iprintLF` на `iprint`, затем создала исполняемый файл и запустила его. Теперь следующая команда вводится на той же строке.

```
[daryayasinovskaya@10 lab06]$ nasm -f elf lab6-2.asm
[daryayasinovskaya@10 lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[daryayasinovskaya@10 lab06]$ ./lab6-2
10[daryayasinovskaya@10 lab06]$
```

Рис. 11: Создание каталога и файла в нем

Создала файл `lab6-3.asm` в каталоге `~/work/arch-pc/lab06`

```
[daryayasinovskaya@10 lab06]$ touch lab6-3.asm
[daryayasinovskaya@10 lab06]$
```

Рис. 12: Создание файла в каталоге

Ввела в файл текст из листинга 6.3

```
mc [daryayasinovskaya@fedora]
GNU nano 7.2 /home/daryayasinovskaya/work/
;-----
; 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,5 ; EAX=5
mov ebx,2 ; EBX=2
mul ebx ; EAX=EAX*EBX
add eax,3 ; EAX=EAX+3
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,3 ; EBX=3
div ebx ; EAX=EAX/3, EDX=остаток от деления
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax,div ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов
mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Остаток от деления: '
mov eax,edx ; вызов подпрограммы печати значения
call iprintLF ; из 'edx' (остаток) в виде символов
call quit ; вызов подпрограммы завершения
```

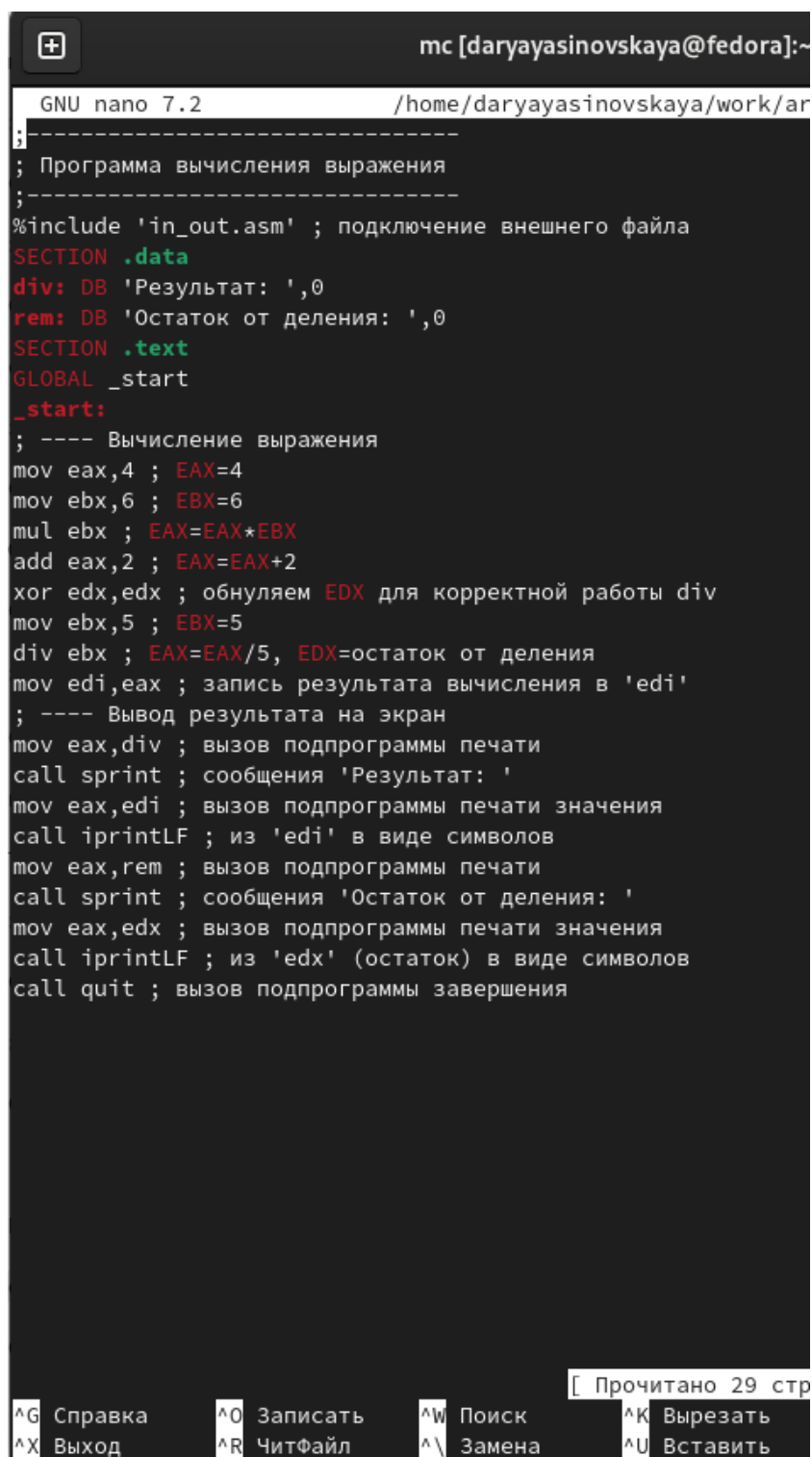
Рис. 13: Ввод текста

Создала исполняемый файл и запустила его

```
[daryayasinovskaya@10 lab06]$ nasm -f elf lab6-3.asm
[daryayasinovskaya@10 lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
[daryayasinovskaya@10 lab06]$ ./lab6-3
Результат: 4
Остаток от деления: 1
[daryayasinovskaya@10 lab06]$
```

Рис. 14: Создание и запуск исполняемого файла

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



```
mc [daryayasinovskaya@fedora]:~
GNU nano 7.2 /home/daryayasinovskaya/work/ar
;-----
; 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
mov ebx,5 ; EBX=5
div ebx ; EAX=EAX/5, EDX=остаток от деления
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax,div ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов
mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Остаток от деления: '
mov eax,edx ; вызов подпрограммы печати значения
call iprintLF ; из 'edx' (остаток) в виде символов
call quit ; вызов подпрограммы завершения

[ Прочитано 29 стр
^G Справка ^O Записать ^W Поиск ^K Вырезать
^X Выход ^R ЧитФайл ^_ Замена ^U Вставить
```

Рис. 15: Изменение текста

Создала исполняемый файл и проверила его работу. Теперь после строки “Результат” выводится 5, а после строки “Остаток от деления” выводится 1

```
[daryayasinovskaya@10 lab06]$ nasm -f elf lab6-3.asm
[daryayasinovskaya@10 lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
[daryayasinovskaya@10 lab06]$ ./lab6-3
Результат: 5
Остаток от деления: 1
[daryayasinovskaya@10 lab06]$
```

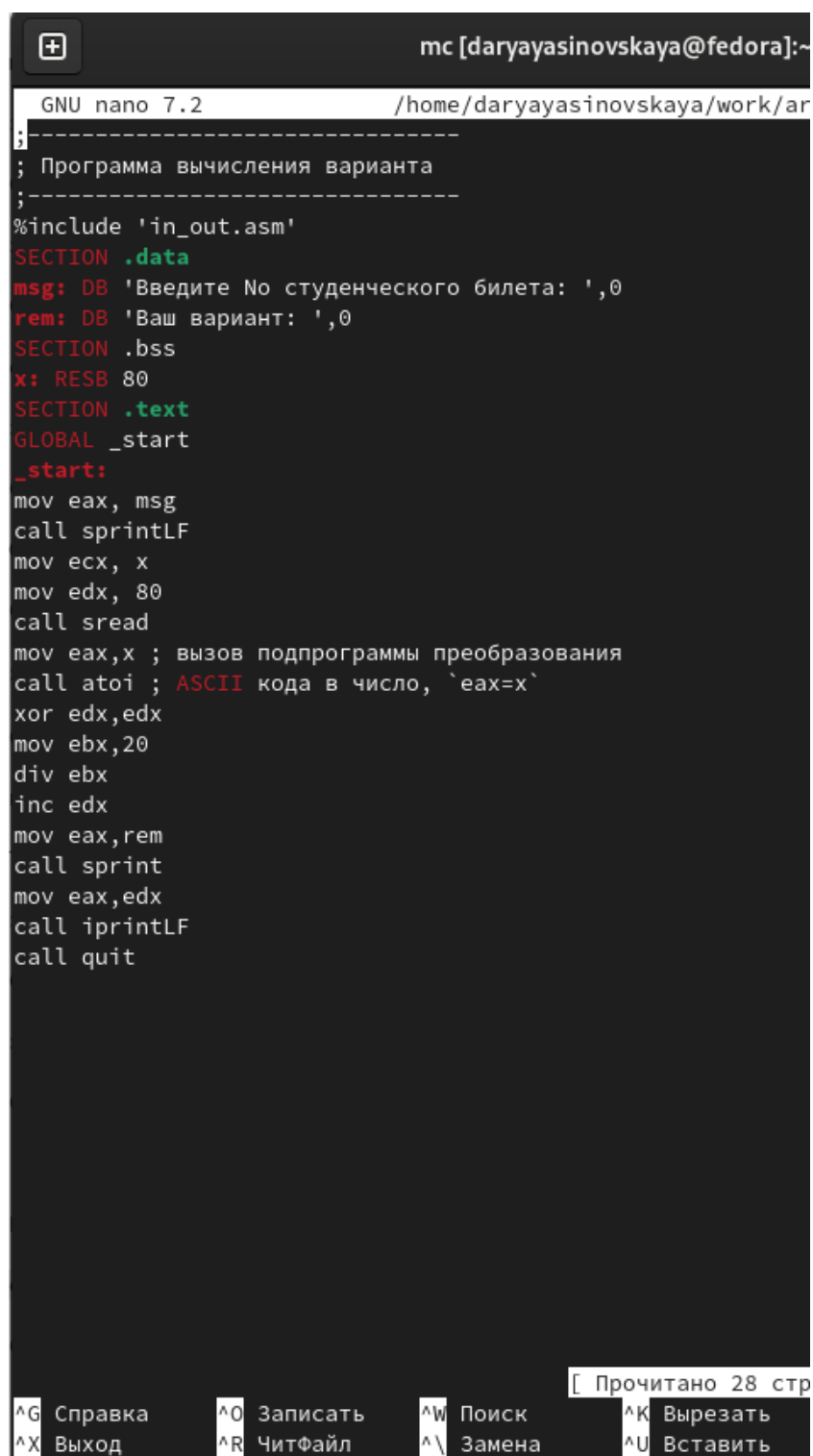
Рис. 16: Создание и запуск исполняемого файла

Создаю файл variant.asm

```
[daryayasinovskaya@10 lab06]$ touch variant.asm
[daryayasinovskaya@10 lab06]$
```

Рис. 17: Создание файла

Ввела в файл текст программы из листинга 6.4



```
mc [daryayasinovskaya@fedora]:~
GNU nano 7.2 /home/daryayasinovskaya/work/ar
;-----
; Программа вычисления варианта
;-----
#include 'in_out.asm'
SECTION .data
msg: DB 'Введите No 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 ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, `eax=x`
xor edx, edx
mov ebx, 20
div ebx
inc edx
mov eax, rem
call sprintf
mov eax, edx
call iprintLF
call quit

[ Прочитано 28 стр
^G Справка      ^O Записать     ^W Поиск        ^K Вырезать
^X Выход        ^R ЧитФайл     ^_ Замена       ^U Вставить
```

Рис. 18: Ввод текста

Создала исполняемый файл и запустила его.

```
[daryayasinovskaya@fedora lab06]$  
[daryayasinovskaya@fedora lab06]$ nasm -f elf variant.asm  
[daryayasinovskaya@fedora lab06]$ ld -m elf_i386 -o variant variant.o  
[daryayasinovskaya@fedora lab06]$ ./variant  
Введите No студенческого билета:  
1132236065  
Ваш вариант: 6  
[daryayasinovskaya@fedora lab06]$
```

Рис. 19: Создание и запуск исполняемого файла

Ответы на вопросы

1. За вывод на экран сообщения “Ваш вариант:” отвечают строки:

```
mov eax,rem call sprint
```

2. Инструкция “mov ecx, x” используется, чтобы положить адрес вводимой строки x в регистр ecx, “mov edx, 80” - запись в регистр edx длины вводимой строки, “call sread” - вызов подпрограммы из внешнего файла, обеспечивающей ввод сообщения с клавиатуры
3. “call atoi” используется для вызова подпрограммы из внешнего файла, которая преобразует ascii-код символа в целое число и записывает результат в регистр eax

4. За вычисления варианта отвечают строки:

```
xor edx,edx ; обнуление edx для корректной работы div  
mov ebx,20 ; ebx = 20  
div ebx ; eax = eax/20, edx - остаток от деления  
inc edx ; edx = edx + 1
```

5. При выполнении инструкции div ebx остаток от деления записывается в регистр edx
6. Инструкция inc edx увеличивает значение регистра edx на 1
7. За вывод на экран результатов вычислений отвечают строки:

```
mov eax,edx call iprintLF
```

Самостоятельная работа

Создала файл для программы вычисления выражения $y = f(x)$

```
[daryayasinovskaya@fedora lab06]$ nasm -f elf lab6-4.asm
[daryayasinovskaya@fedora lab06]$ ld -m elf_i386 -o lab6-4 lab6-4.o
[daryayasinovskaya@fedora lab06]$ ./lab6-4
```

Рис. 1: Создание файла

Написала программу для вычисления функции

```
GNU nano 7.2 /home/daryayasinovskaya/work/arch-pc/lab06/lab6-4.asm
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; секция иницированных данных
msg: DB 'Введите значение переменной x: ',0
res: DB 'Результат: ',0
SECTION .bss ; секция не иницированных данных
x: RESB 80 ; Переменная, значение к-рой будем вводить с клавиатуры
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
; ---- Вычисление выражения
mov eax, msg ; запись адреса выводимого сообщения в eax
call sprint ; вызов подпрограммы печати сообщения
mov ecx, x ; запись адреса переменной в ecx
mov edx, 80 ; запись длины вводимого значения в edx
call sread ; вызов подпрограммы ввода сообщения
mov eax, x ; вызов подпрограммы преобразования
; ---- функция ----
call atoi ; ASCII кода в число, eax=x
mov ebx, eax ;
mul ebx ;
mul ebx ;
mov ebx, 2 ;
div ebx ;
add eax, 1 ;
mov edi, eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax, res ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax, edi ; вызов подпрограммы печати значения
call iprintf ; из 'edi' в виде символов
call quit ; вызов подпрограммы завершения

[ Прочитана 31 строка ]
^G Справка      ^O Записать     ^M Поиск        ^K Вырезать     ^T Выполнить    ^C Позиция      ^-U Отмена
^X Выход        ^R ЧитФайл     ^N Замена       ^U Вставить     ^J Выводить     ^/ К строке    ^-E Повтор
```

Рис. 2: Текст программы

Создала исполняемый файл и проверила его работу для x1

```
[daryayasinovskaya@fedora lab06]$ nasm -f elf lab6-4.asm
[daryayasinovskaya@fedora lab06]$ ld -m elf_i386 -o lab6-4 lab6-4.o
[daryayasinovskaya@fedora lab06]$ ./lab6-4
Введите значение переменной x: 2
Результат: 5
[daryayasinovskaya@fedora lab06]$
```

Рис. 3: Создание и запуск исполняемого файла

Создала исполняемый файл и проверила его работу для x2

```
[daryayasinovskaya@fedora lab06]$ nasm -f elf lab6-4.asm
[daryayasinovskaya@fedora lab06]$ ld -m elf_i386 -o lab6-4 lab6-4.o
[daryayasinovskaya@fedora lab06]$ ./lab6-4
Введите значение переменной x: 2
Результат: 5
[daryayasinovskaya@fedora lab06]$ ./lab6-4
Введите значение переменной x: 5
Результат: 63
[daryayasinovskaya@fedora lab06]$
```

Рис. 4: Создание и запуск исполняемого файла

Листинги программ

Листинг 6.1. Программа вывода значения регистра `eax` `%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`

Листинг 6.2. Программа вывода значения регистра `eax` `%include 'in_out.asm'`
`SECTION .text GLOBAL _start _start: mov eax,'6' mov ebx,'4' add eax,ebx call iprintLF`
`call quit`

Листинг 6.3. Программа вычисления выражения $f(x) = (5 * 2 + 3)/3$
;----- ; 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,5 ; EAX=5 mov ebx,2 ; EBX=2 mul ebx ;`
`EAX=EAX*EBX add eax,3 ; EAX=EAX+3 xor edx,edx ; обнуляем EDX для корректной`
`работы div mov ebx,3 ; EBX=3 div ebx ; EAX=EAX/3, EDX=остаток от деления mov`
`edi,eax ; запись результата вычисления в 'edi' ; -- Вывод результата на экран`
`mov eax,div ; вызов подпрограммы печати call sprint ; сообщения 'Результат:'`
`mov eax,edi ; вызов подпрограммы печати значения call iprintLF ; из 'edi' в виде`
`символов mov eax,rem ; вызов подпрограммы печати call sprint ; сообщения`
`'Остаток от деления:' mov eax,edx ; вызов подпрограммы печати значения call`
`iprintLF ; из 'edx' (остаток) в виде символов call quit ; вызов подпрограммы`
`завершения`

Листинг 6.4. Программа вычисления вычисления варианта задания по номеру
студенческого билета ;----- ; Программа вычисления варианта

```
;----- %include 'in_out.asm' SECTION .data msg: DB 'Введите No 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 ; вызов подпрограммы преобразования call atoi ; ASCII кода в число, eax=x xor edx,edx mov ebx,20 div ebx inc edx mov eax,rem call sprint mov eax,edx call iprintLF call quit
```

Выводы

Я освоила арифметические инструкции языка ассемблера NASM

Список литературы