

Отчёта по лабораторной работе №5

Основы работы с Midnight Commander

Жозе Рамос Домингуш

Содержание

1	Цель работы	1
2	Задание	1
3	Выполнение лабораторной работы.....	1
3.1	Порядок выполнения лабораторной работы	1
3.2	Задание для самостоятельной работы.....	12
4	Выводы	15

1 Цель работы

Освоить инструкции языка ассемблера mov. Приобрести знания использования Midnight Commander.

2 Задание

Написать 2 программы по примеру и впоследствии изменить их по условию.

3 Выполнение лабораторной работы

3.1 Порядок выполнения лабораторной работы

Открываем Midnight Commander (рис. 1).

mc

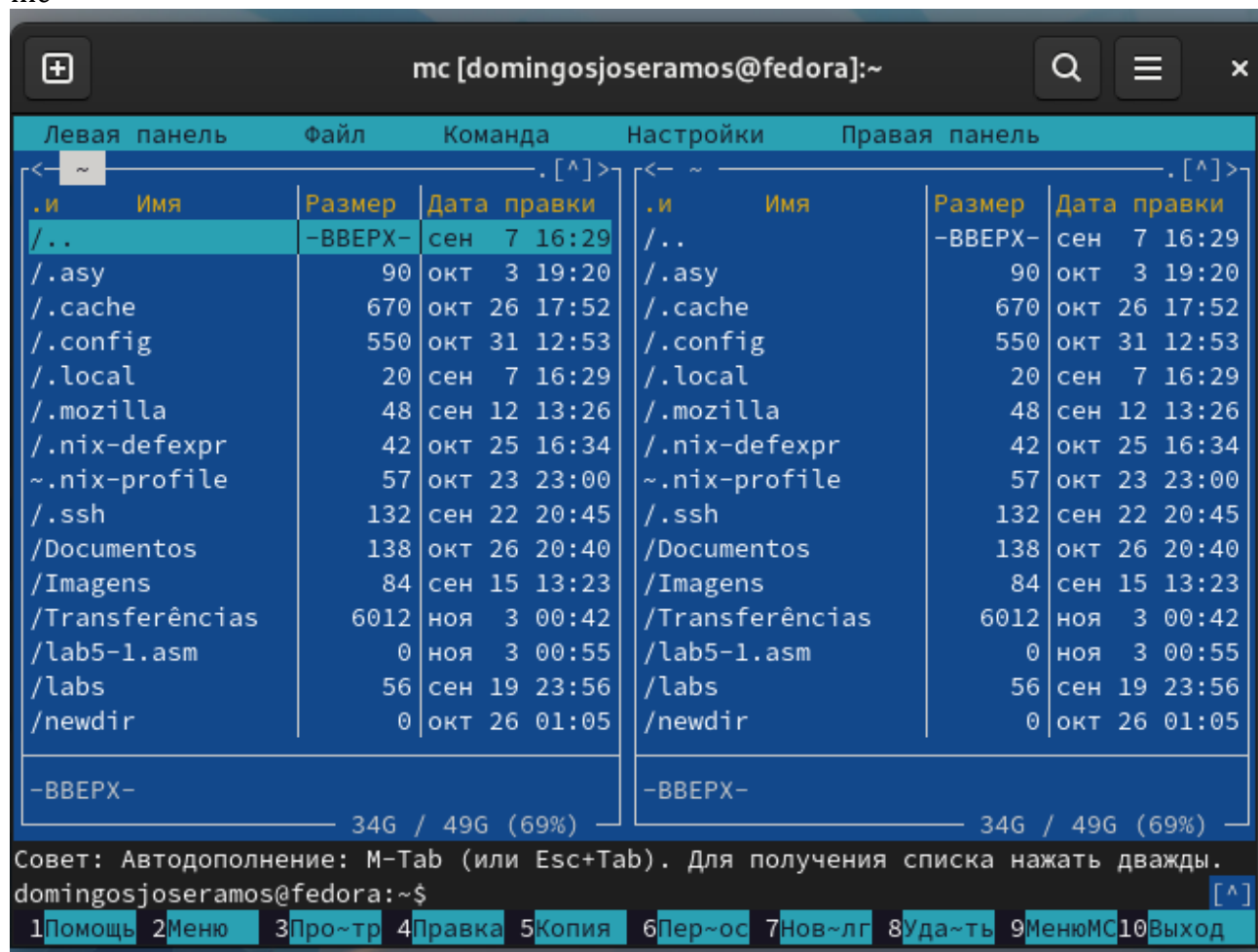


Figure 1: Вводим в консоль команду mc

Переходим в каталог, созданный при выполнении 4 ЛБ (рис. 2).

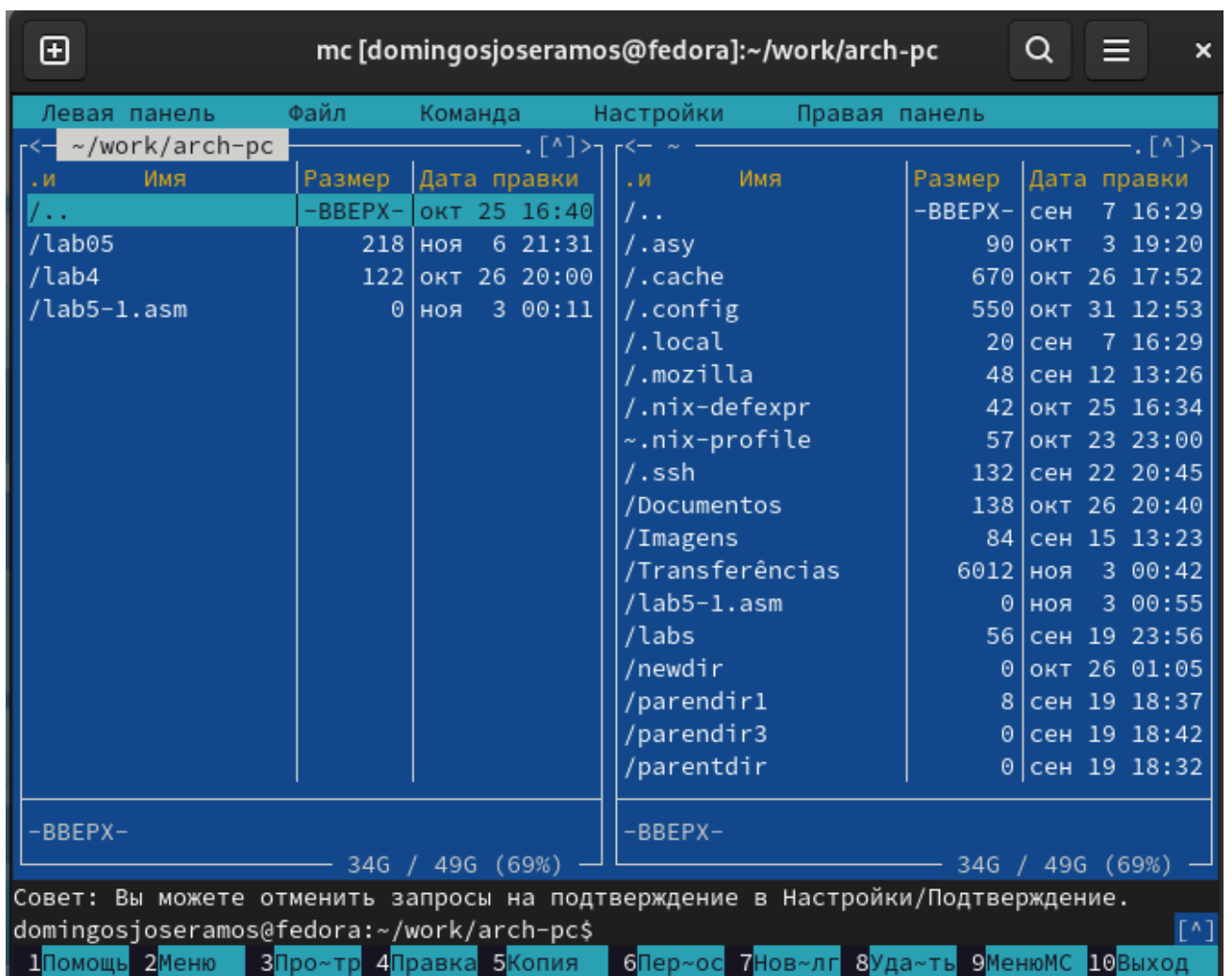


Figure 2: Переходим в каталог

Создаем каталог lab05 (рис. 3).

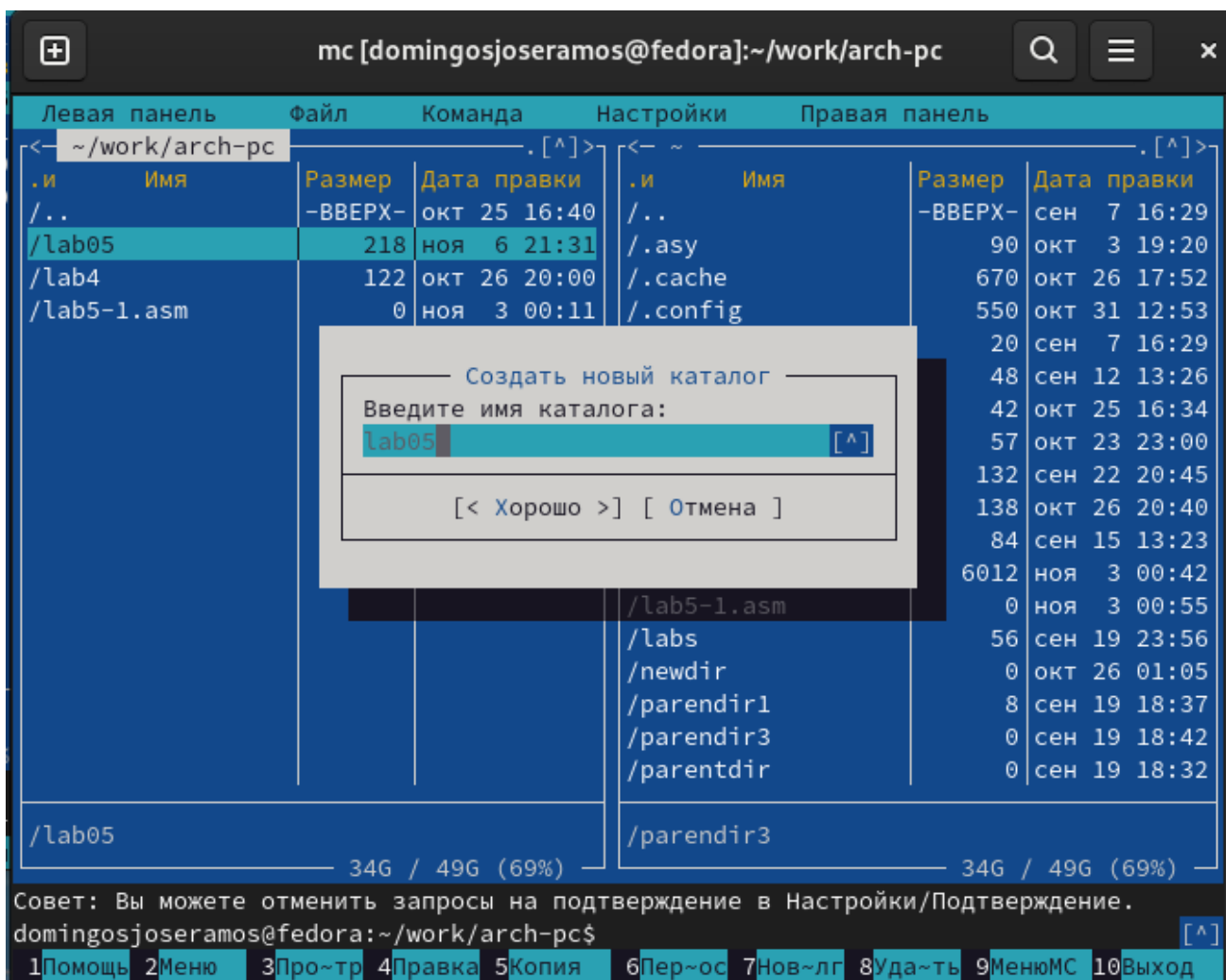
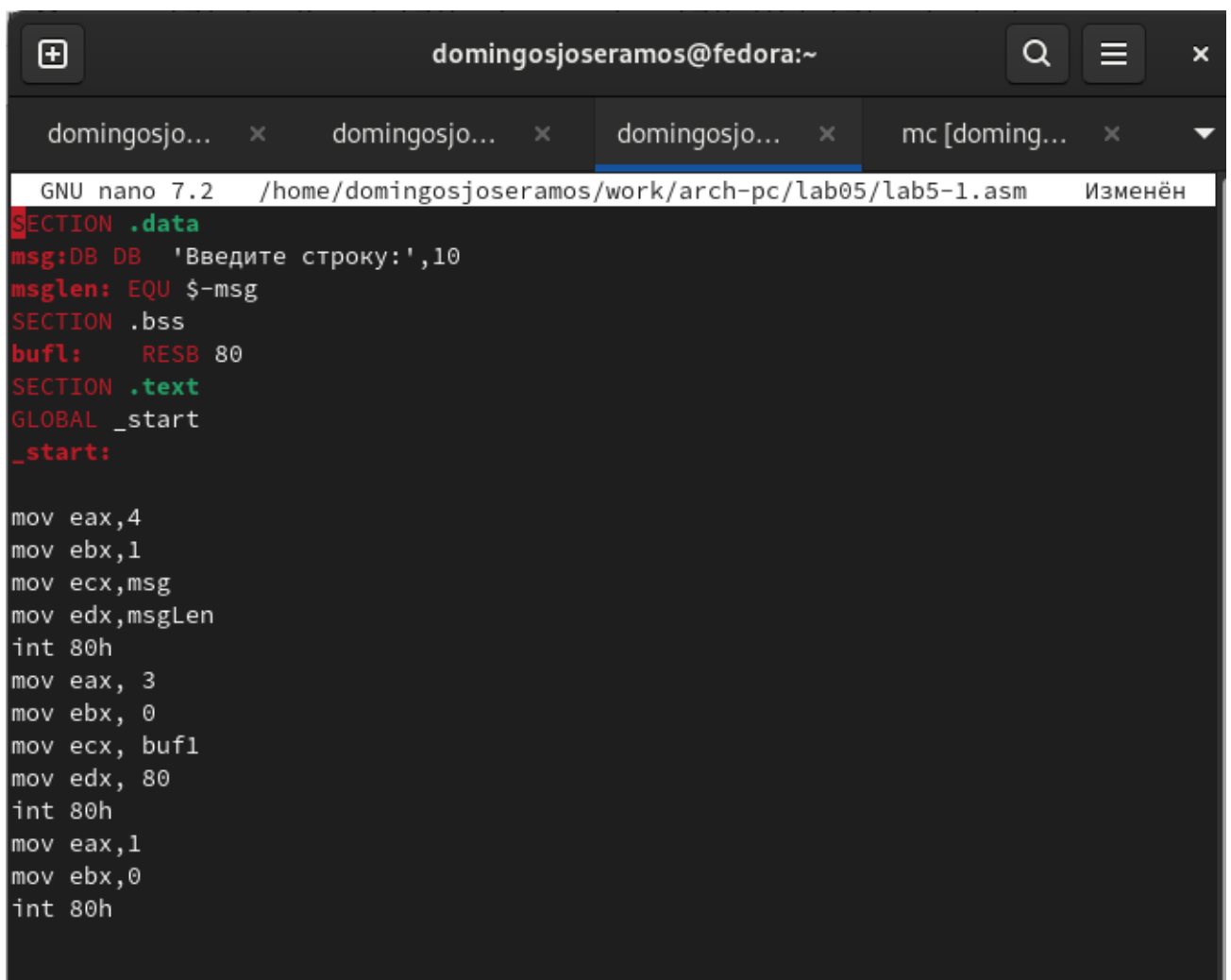


Figure 3: Создаем каталог функциональной клавишей F7

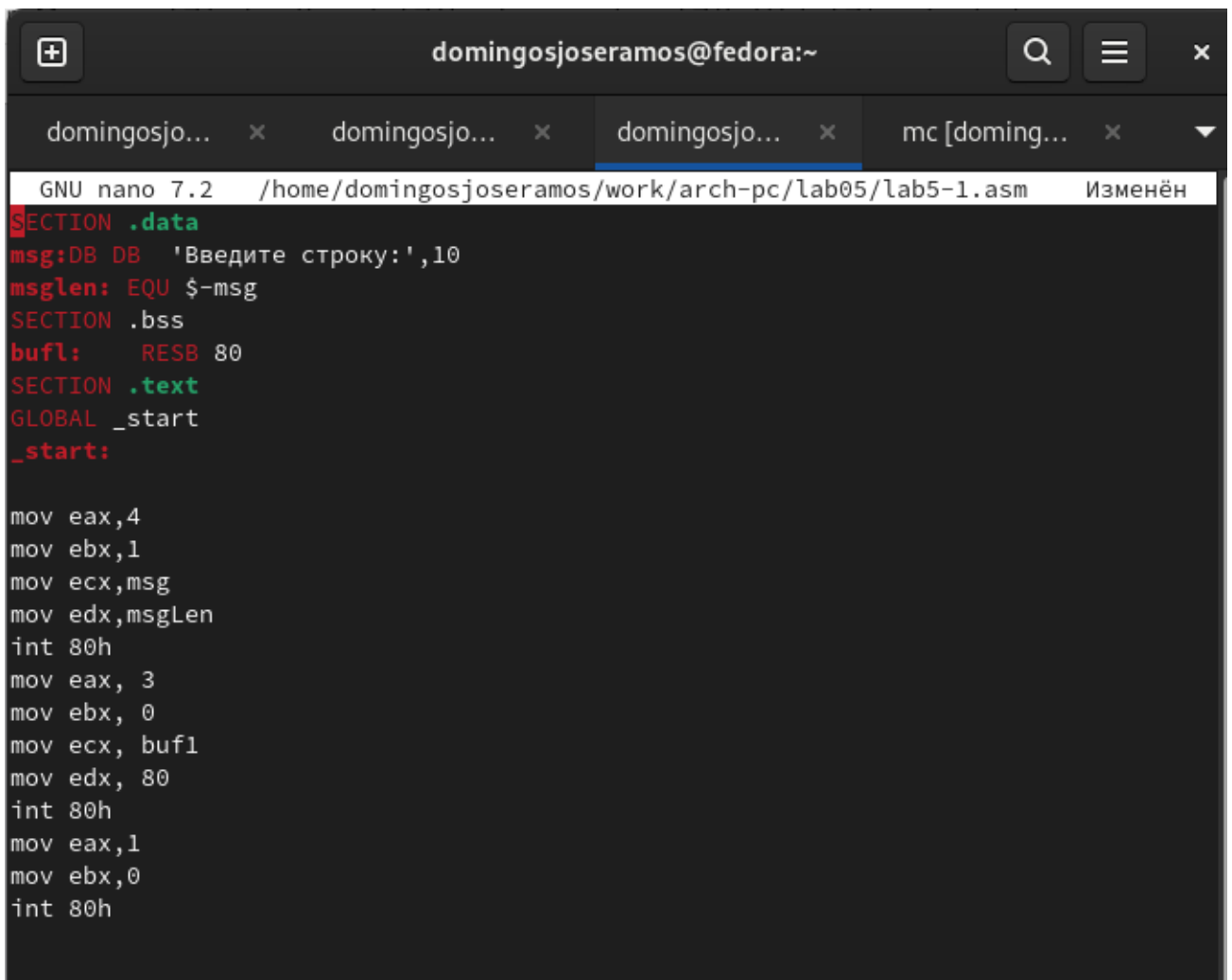
Создаем файл lab5-1.asm (рис. 4).



```
domingosjoseramos@fedora:~  
domingosjo... x domingosjo... x domingosjo... x mc [doming... x  
GNU nano 7.2 /home/domingosjoseramos/work/arch-pc/lab05/lab5-1.asm Изменён  
SECTION .data  
msg:DB DB 'Введите строку:',10  
msglen: EQU $-msg  
SECTION .bss  
buf1: RESB 80  
SECTION .text  
GLOBAL _start  
_start:  
  
mov eax,4  
mov ebx,1  
mov ecx,msg  
mov edx,msglen  
int 80h  
mov eax, 3  
mov ebx, 0  
mov ecx, buf1  
mov edx, 80  
int 80h  
mov eax,1  
mov ebx,0  
int 80h
```

Figure 5: Открываем файл функциональной клавишей, заполняем и сохраняем

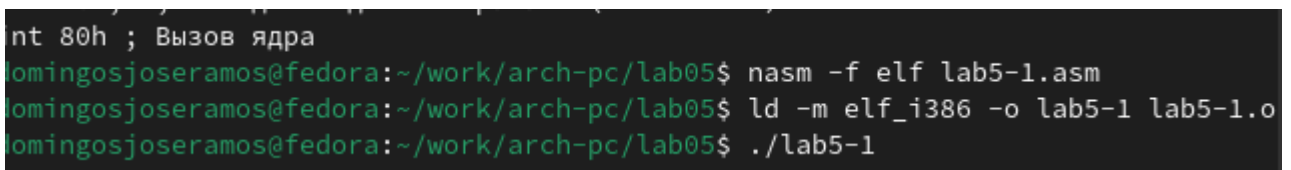
Открываем файл для просмотра (рис. 6).



```
domingosjoseramos@fedora:~  
domingosjo... x domingosjo... x domingosjo... x mc [doming... x  
GNU nano 7.2 /home/domingosjoseramos/work/arch-pc/lab05/lab5-1.asm Изменён  
SECTION .data  
msg:DB DB 'Введите строку:',10  
msglen: EQU $-msg  
SECTION .bss  
buf1: RESB 80  
SECTION .text  
GLOBAL _start  
_start:  
  
mov eax,4  
mov ebx,1  
mov ecx,msg  
mov edx,msglen  
int 80h  
mov eax, 3  
mov ebx, 0  
mov ecx, buf1  
mov edx, 80  
int 80h  
mov eax,1  
mov ebx,0  
int 80h
```

Figure 6: Открываем файл и убеждаемся, что файл содержит текст программы

Транслируем текст программы и запускаем исполняемый файл (рис. 7).



```
nt 80h ; Вызов ядра  
domingosjoseramos@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm  
domingosjoseramos@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o  
domingosjoseramos@fedora:~/work/arch-pc/lab05$ ./lab5-1
```

Figure 7: Проверяем, как работает данная программа

Скачиваем файл со страницы курса (рис. 8).

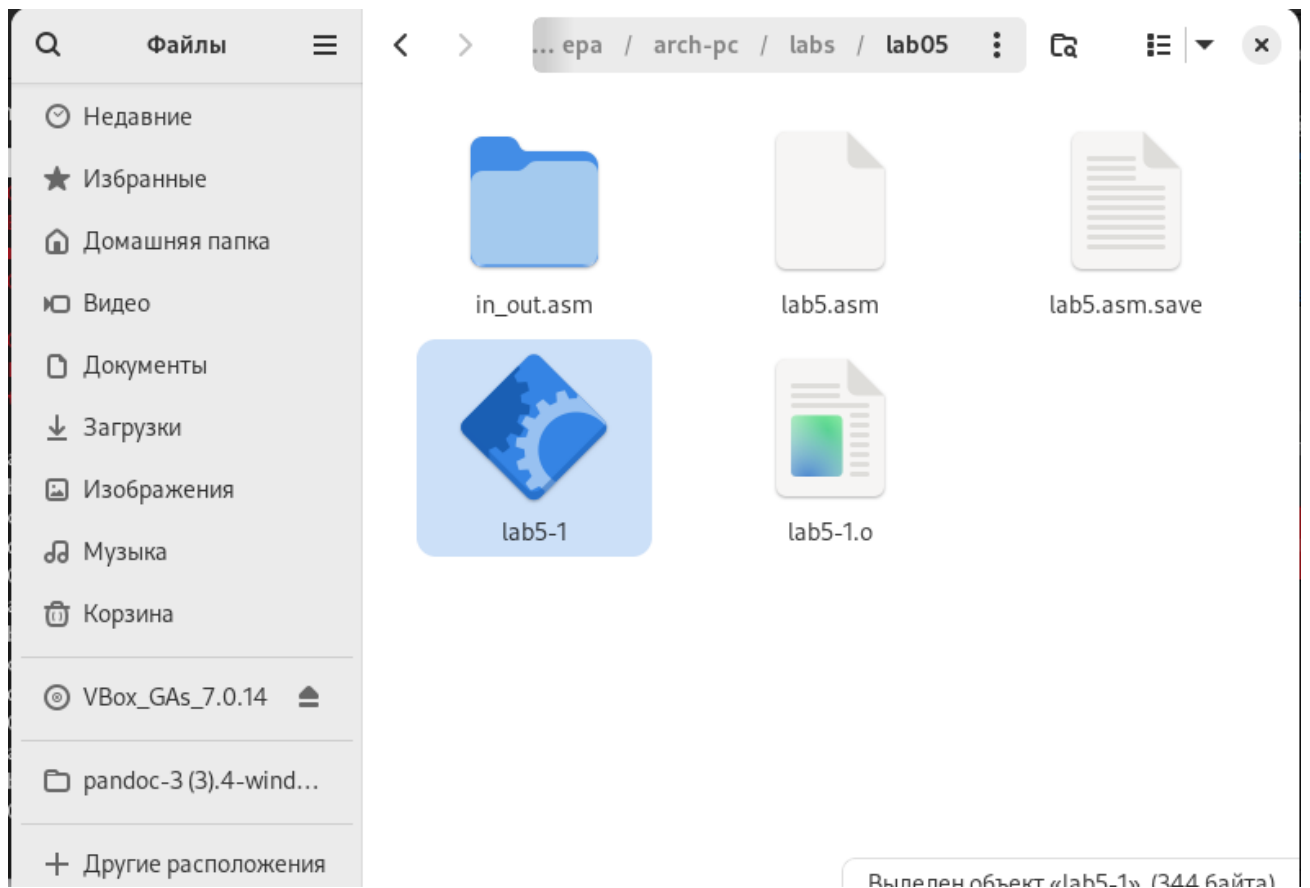


Figure 8: Скачиваем файл

Копируем файл в нужную директорию (рис. 9).

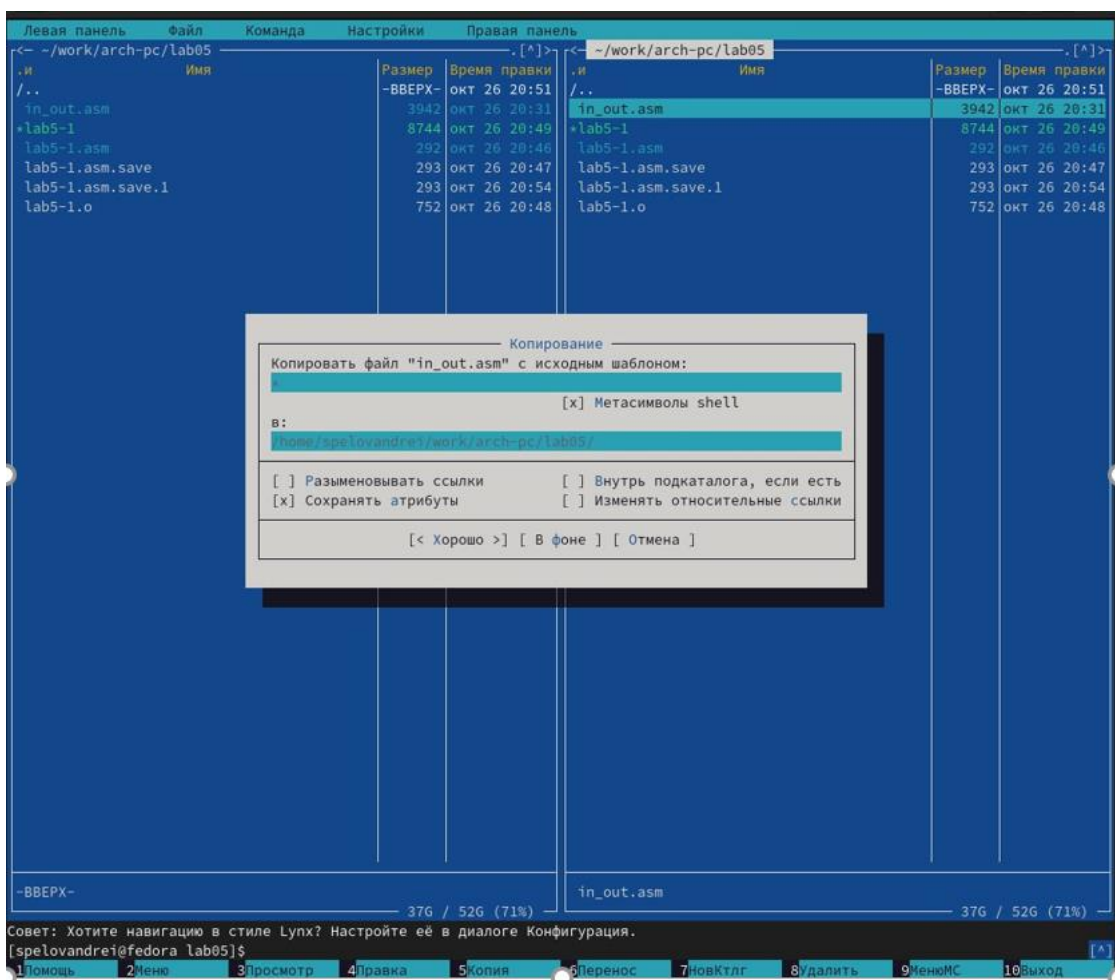


Figure 9: Копируем скаченный файл

Создаем копию файла lab5-1.asm (рис. 10).

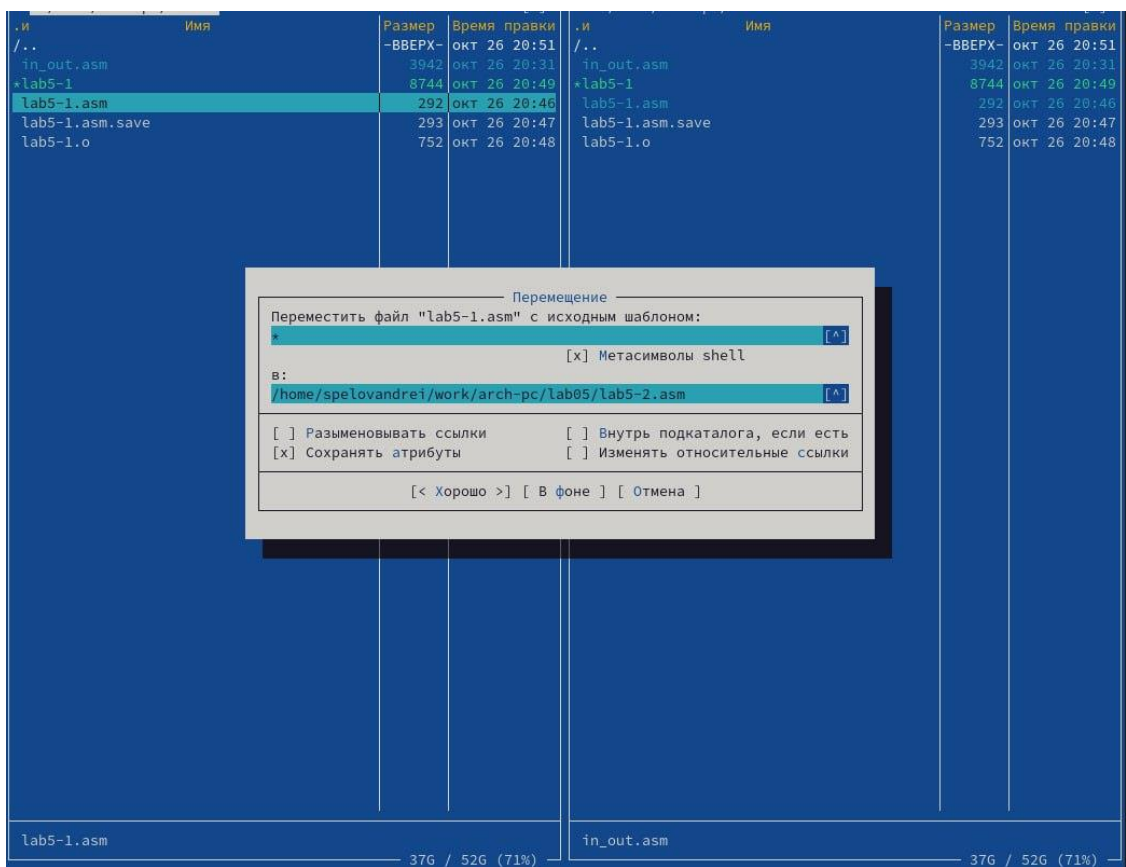


Figure 10: Создаем копию файла клавишей F6

Проверяем созданный файл (рис. 11).

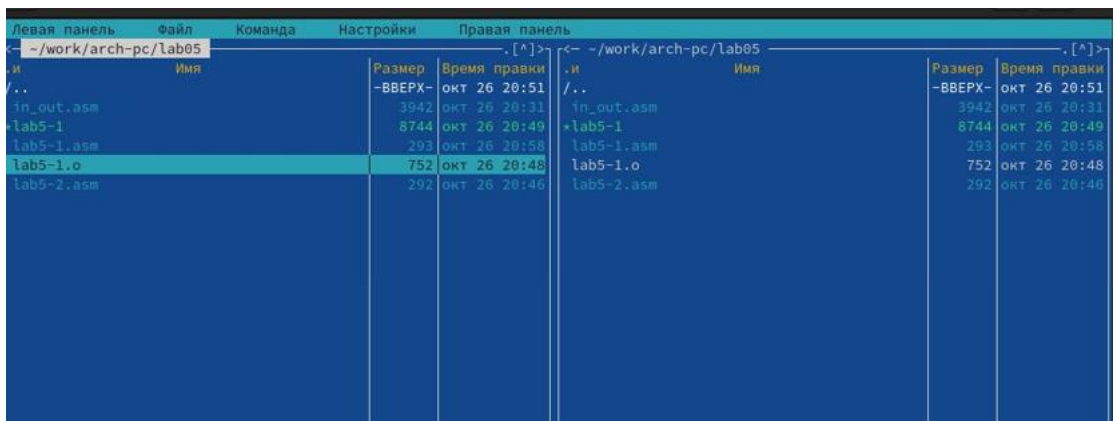


Figure 11: Проверяем скопировался ли файл

Открываем новый файл и заполняем его в соответствии с листингом (рис. 12).

```
lab5-2.asm      [-M--] 10 L:[ 1+13 14/ 14] *(224 / 224b) <EOF>
#include 'in_out.asm'
SECTION .data
msg: DB 'Введите строку:',0h
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintLF
mov ecx, buf1
mov edx, 80
call sread
call quit
```

Figure 12: Открываем и заполняем файл

Транслируем и запускаем новый файл (рис. 13).

```
domingosjoseramos@fedora:~/work/arch-pc/lab05$ nasm -f elf32 lab5-2.asm
domingosjoseramos@fedora:~/work/arch-pc/lab05$ nasm -f elf32 lab5-1.asm -o lab5-2.o
domingosjoseramos@fedora:~/work/arch-pc/lab05$ nasm -f elf32 lab5-2.asm -o lab5-2.o
domingosjoseramos@fedora:~/work/arch-pc/lab05$ ./lab5-2
bash: ./lab5-2: Нет такого файла или каталога
domingosjoseramos@fedora:~/work/arch-pc/lab05$
```

Figure 13: Смотрим, как сработала программа

Снова открываем файл для редактирования и меняем sprintLF на sprint(рис. 14).

```
lab5-2.asm      [-M--] 15 L:[ 1+11 12/ 16] *(186 / 249b) 0010 0x00A
#include 'in_out.asm'

SECTION .data
msg: DB 'Введите строку:',0h

SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprint
mov ecx, buf1
mov edx, 80
call sread
call quit
```

Figure 14: Редактируем файл

Транслируем и запускаем файл(рис. 15).

```
domingosjoseramos@fedora:~/work/arch-pc/lab05$ nasm -f elf32 lab5-2.asm
domingosjoseramos@fedora:~/work/arch-pc/lab05$ nasm -f elf32 lab5-1.asm -o lab5-2.o
domingosjoseramos@fedora:~/work/arch-pc/lab05$ nasm -f elf32 lab5-2.asm -o lab5-2.o
domingosjoseramos@fedora:~/work/arch-pc/lab05$ ./lab5-2
bash: ./lab5-2: Нет такого файла или каталога
domingosjoseramos@fedora:~/work/arch-pc/lab05$
```

Figure 15: Смотрим, как сработала программа и сравниваем с прошлой

Таким образом можем понять, что команда `sprint` выводит текст в той же строке, а `sprintLF` переносит на новую строку.

3.2 Задание для самостоятельной работы

Создаем копию файла `lab5-1.asm` и называем его так же (рис. 16).

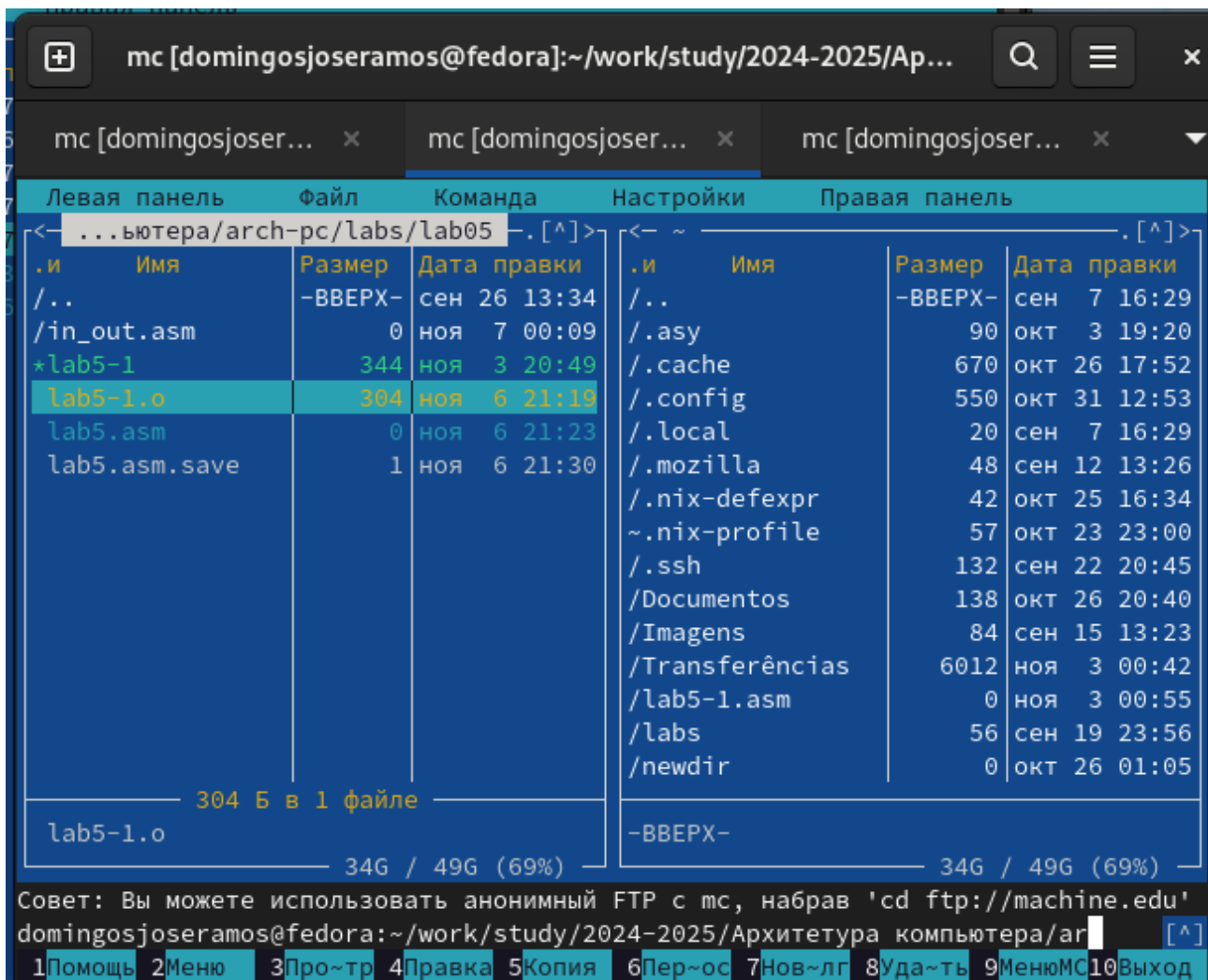
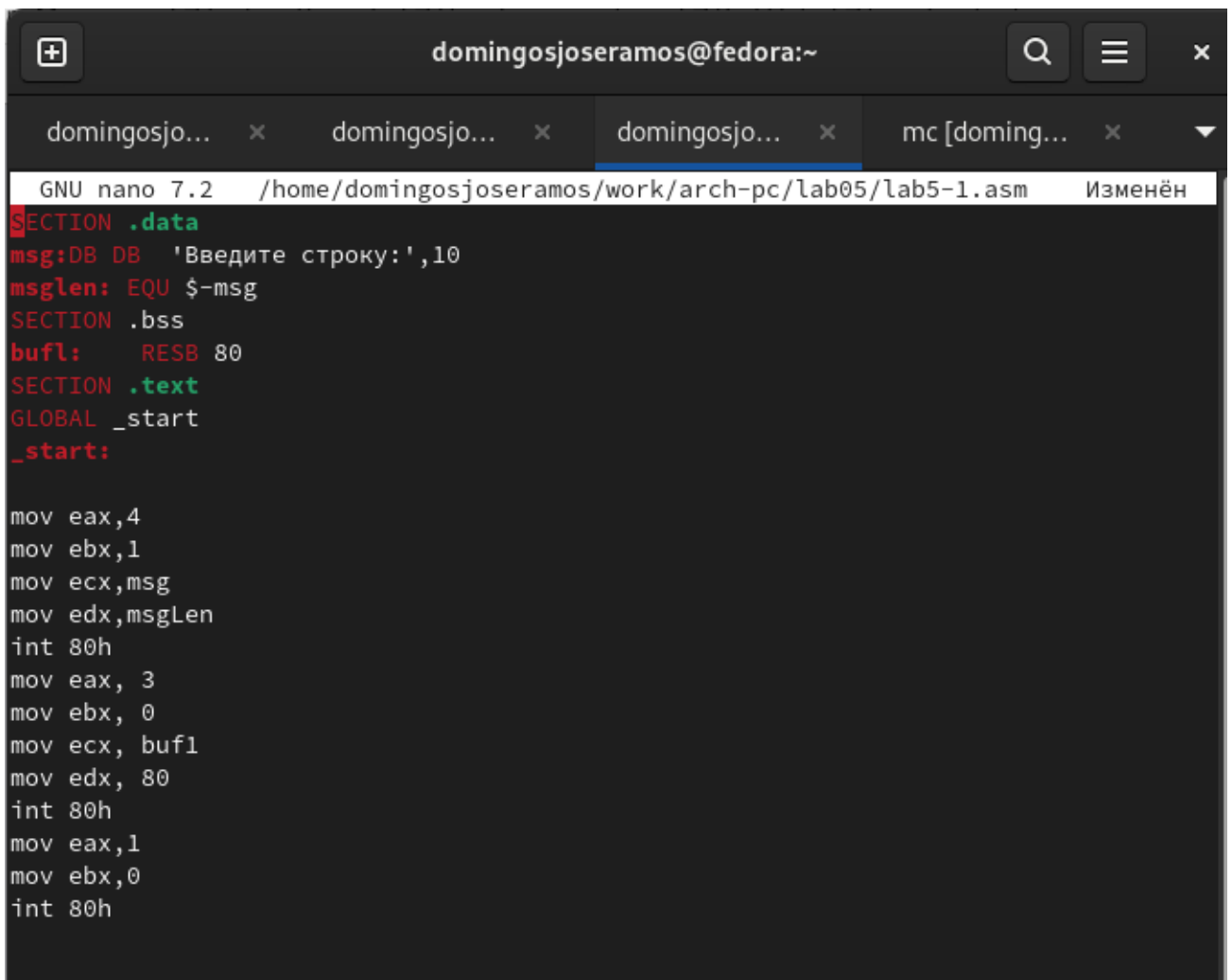


Figure 16: Создаем копию файла `lab5-1.asm`

Редактируем файл, чтобы введенный текст с клавиатуры выводился в консоль (рис. 17).

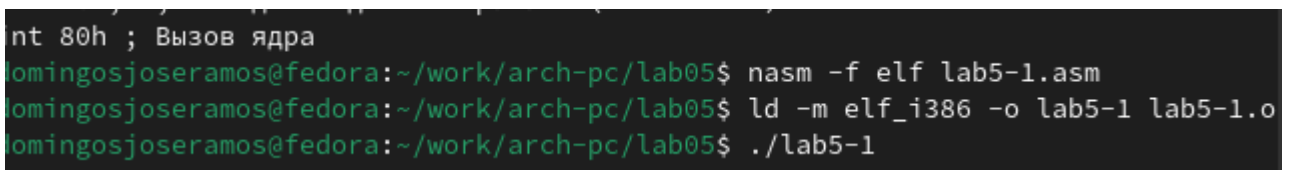


```
domingosjoseramos@fedora:~
GNU nano 7.2 /home/domingosjoseramos/work/arch-pc/lab05/lab5-1.asm  Изменён
SECTION .data
msg:DB DB 'Введите строку:',10
msglen: EQU $-msg
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:

mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msglen
int 80h
mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h
mov eax,1
mov ebx,0
int 80h
```

Figure 17: Редактируем файл

Транслируем файл и запускаем программу (рис. 18).



```
domingosjoseramos@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
domingosjoseramos@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
domingosjoseramos@fedora:~/work/arch-pc/lab05$ ./lab5-1
```

Figure 18: Проверяем правильность написания программы

Создаем копию файла lab5-2.asm и называем его так же (рис. 19).

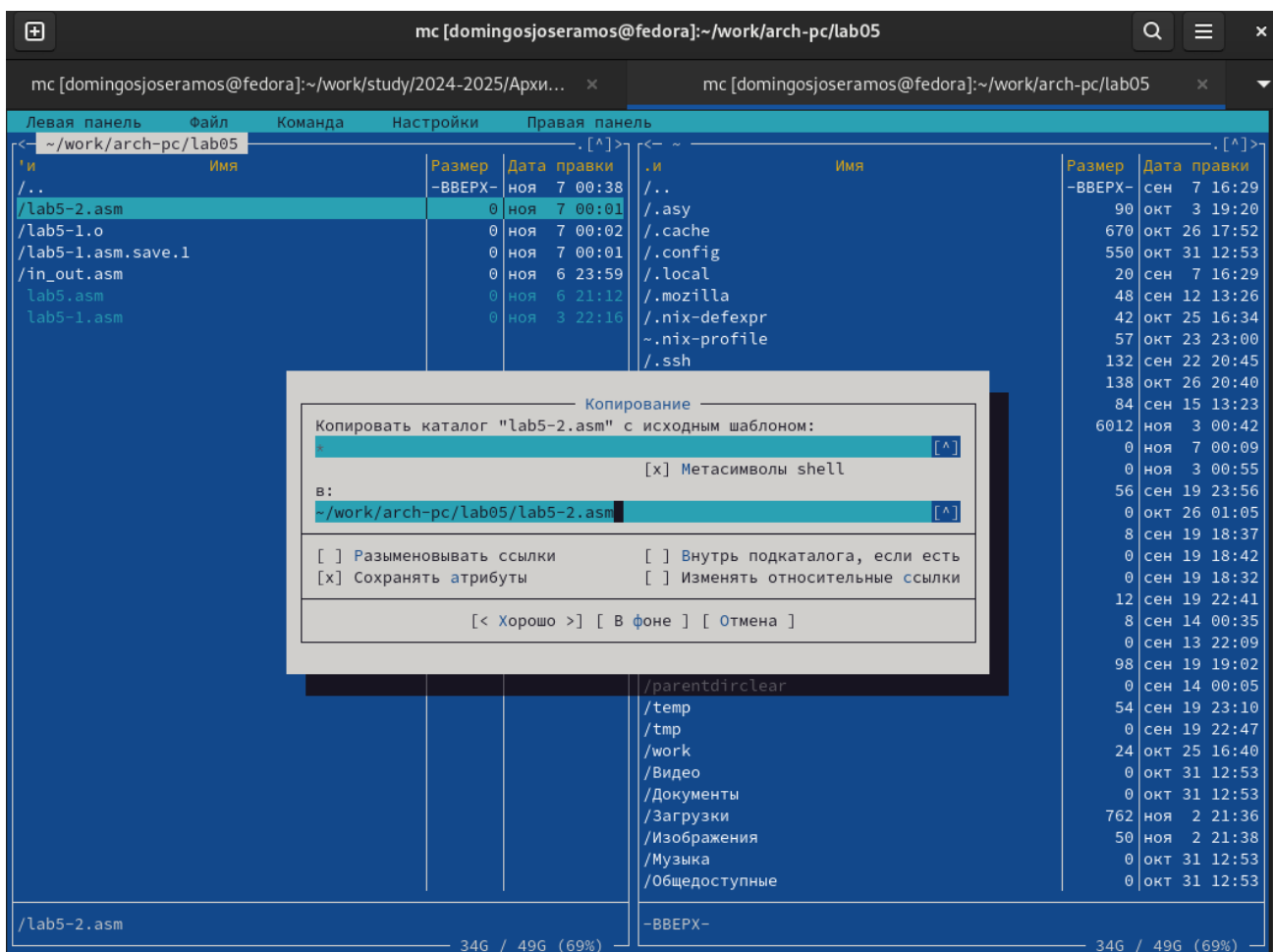


Figure 19: Создаем копию файла lab5-2.asm

Редактируем файл, чтобы введенный текст с клавиатуры выводился в консоль (рис. 20).

```

lab5-2.asm      [-M--] 13 L:[ 1+15 16/ 16] *(278 / 278b) <EOF>
#include 'in_out.asm'
SECTION .data
msg: DB 'Введите строку: ' ,0h
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,msg
call sprintf
mov ecx,buf1
mov edx,80
call sread
mov eax,buf1
call sprintf
call quit

```

Figure 20: Редактируем файл

Транслируем файл и запускаем программу (рис. 21).

```

domingosjoseramos@fedora:~/work/arch-pc/lab05$ nasm -f elf32 lab5-2.asm
domingosjoseramos@fedora:~/work/arch-pc/lab05$ nasm -f elf32 lab5-1.asm -o lab5-2.o
domingosjoseramos@fedora:~/work/arch-pc/lab05$ nasm -f elf32 lab5-2.asm -o lab5-2.o
domingosjoseramos@fedora:~/work/arch-pc/lab05$ ./lab5-2
bash: ./lab5-2: Нет такого файла или каталога
domingosjoseramos@fedora:~/work/arch-pc/lab05$

```

Figure 21: Проверяем правильность написания программы

4 Выводы

Мы приобрели навыки работы с Midnight Commander и освоили инструкцию mov.