

Отчёта по лабораторной работе №5

Основы работы с Midnight Commander

Спелов Андрей Николаевич

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
3.1	Порядок выполнения лабораторной работы	6
3.2	Задание для самостоятельной работы	15
4	Выводы	19

Список иллюстраций

3.1	Вводим в консоль команду <code>mc</code>	6
3.2	Переходим в каталог	7
3.3	Создаем каталог функциональной клавишей F7	8
3.4	Воспользуемся командой <code>touch</code>	9
3.5	Открываем файл функциональной клавишей, заполняем и сохраняем	10
3.6	Открываем файл и убеждаемся, что файл содержит текст программы	11
3.7	Проверяем, как работает данная программа	11
3.8	Скачиваем файл	12
3.9	Копируем скаченный файл	12
3.10	Создаем копию файла клавишей F6	13
3.11	Проверяем скопировался ли файл	13
3.12	Открываем и заполняем файл	14
3.13	Смотрим, как сработала программа	14
3.14	Редактируем файл	14
3.15	Смотрим, как сработала программа и сравниваем с прошлой . . .	15
3.16	Создаем копию файла <code>lab5-1.asm</code>	15
3.17	Редактируем файл	16
3.18	Проверяем правильность написания программы	16
3.19	Создаем копию файла <code>lab5-2.asm</code>	17
3.20	Редактируем файл	17
3.21	Проверяем правильность написания программы	18

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. 3.1).

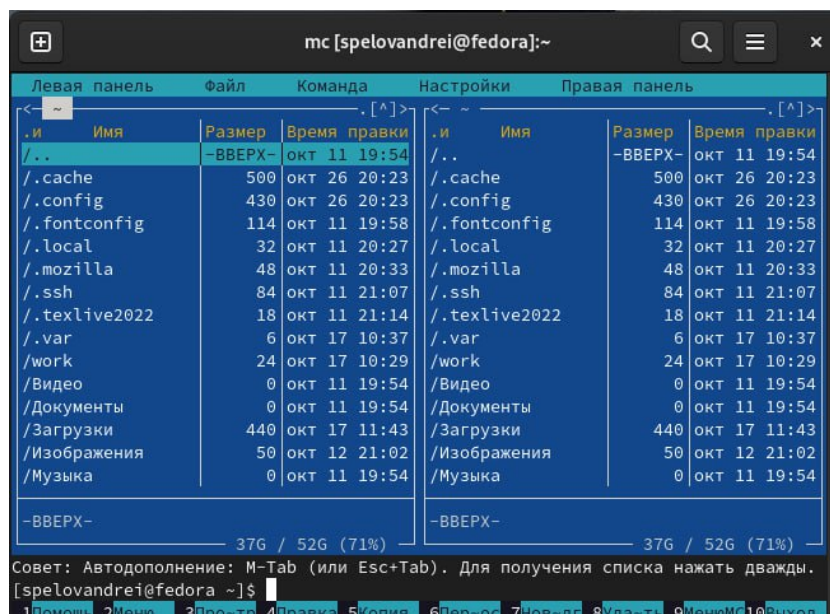


Рис. 3.1: Вводим в консоль команду mc

Переходим в каталог, созданный при выполнении 4 ЛБ (рис. 3.2).

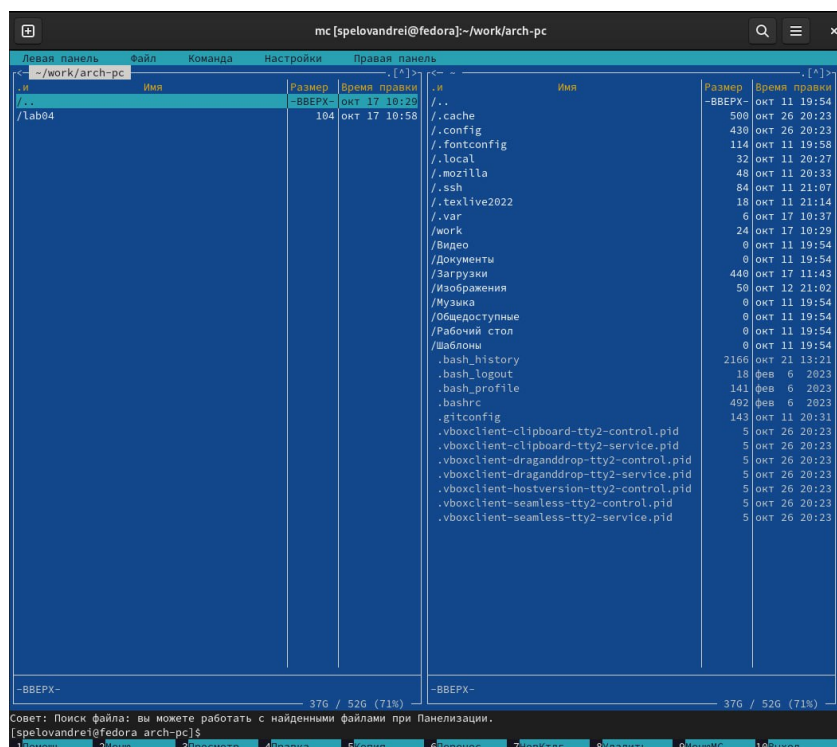


Рис. 3.2: Переходим в каталог

Создаем каталог lab05 (рис. 3.3).

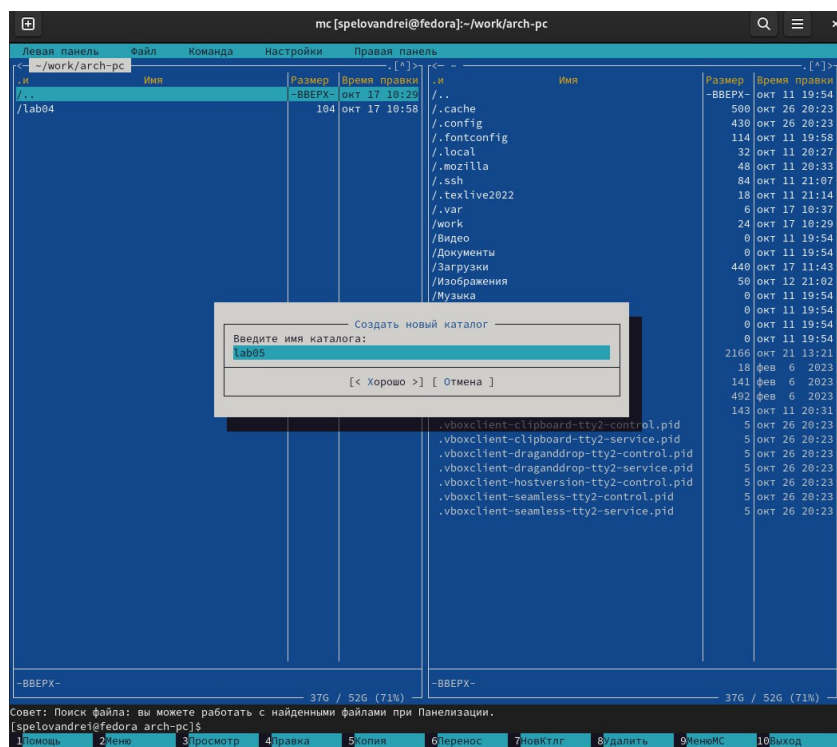


Рис. 3.3: Создаем каталог функциональной клавишей F7

Создаем файл lab5-1.asm (рис. 3.4).

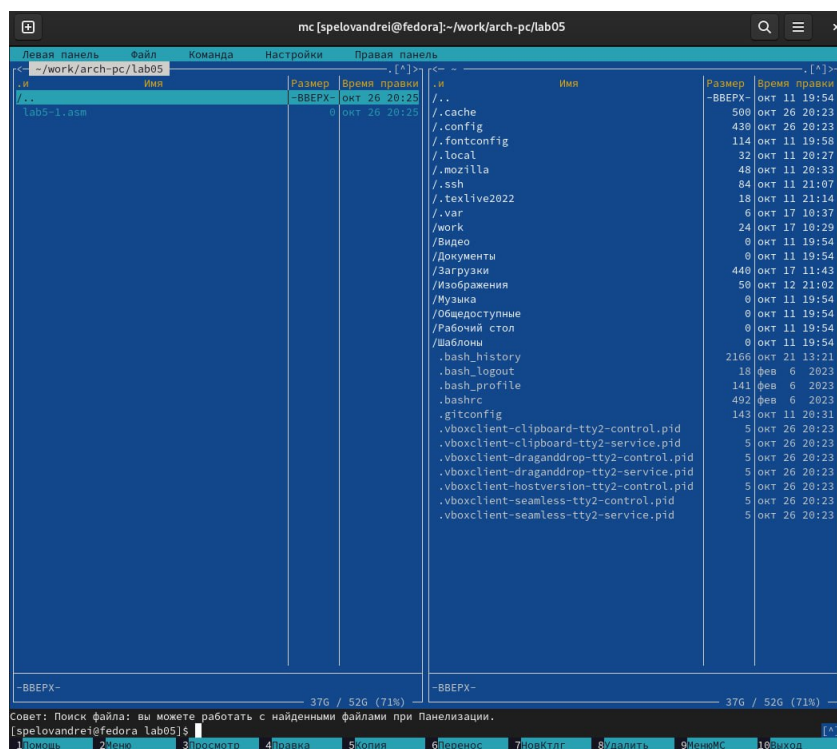
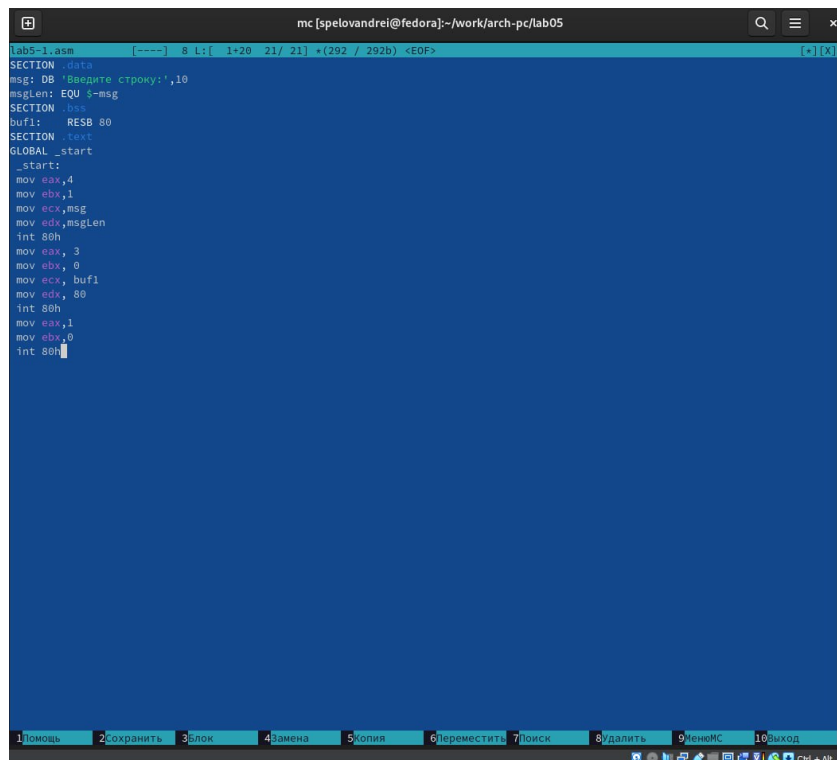


Рис. 3.4: Воспользуемся командой touch

Открываем файл для редактирования и заполняем его по листингу (рис. 3.5).

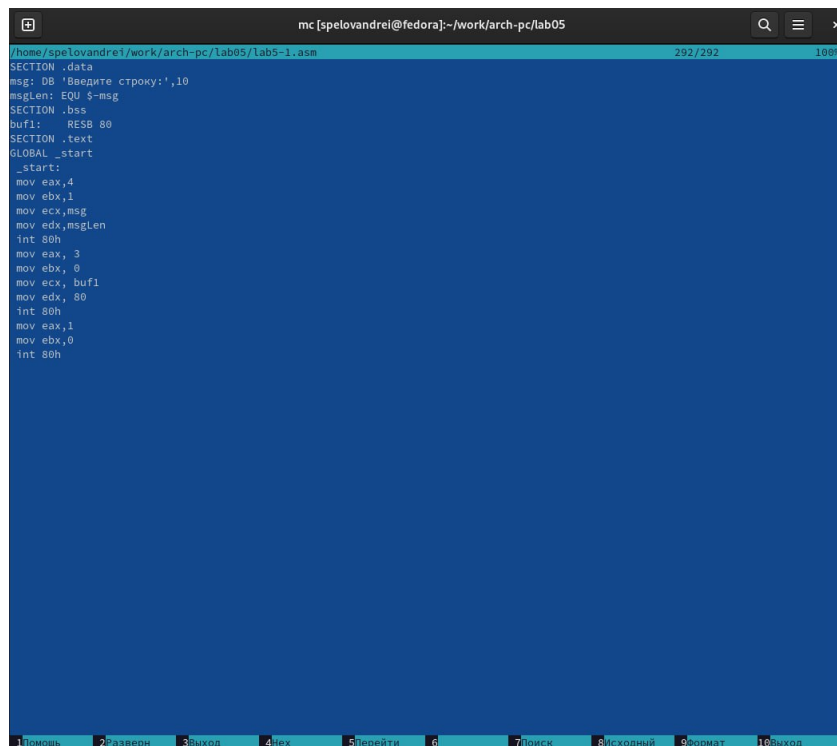


```
mc [spelovandrei@fedora]:~/work/arch-pc/lab05
lab5-1.asm [----] 8 L: [ 1+20 21/ 21] *(292 / 292b) <EOF> [*)(X]
SECTION .data
msg: DB 'Введите строку:',10
msglen: EQU $-msg
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msglen
int 80h
mov eax,3
mov ebx,0
mov ecx,buf1
mov edx,80
int 80h
mov eax,1
mov ebx,0
int 80h
```

1.Помощь 2.Сохранить 3.Блок 4.Замена 5.Копия 6.Переместить 7.Поиск 8.Удалить 9.МенюМС 10.Выход

Рис. 3.5: Открываем файл функциональной клавишей, заполняем и сохраняем


Открывем файл для просмотра (рис. 3.6).



```
mc [spelovandreif@fedora]:~/work/arch-pc/lab05
/home/spelovandreif/work/arch-pc/lab05/lab5-1.asm 292/292 108%
SECTION .data
msg: 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
```

Рис. 3.6: Открываем файл и убеждаемся, что файл содержит текст программы

Транслируем текст программы и запускаем исполняемый файл (рис. 3.7).



```
spelovandreif@fedora:~/work/arch-pc/lab05
[spelovandreif@fedora ~]$ mc
[spelovandreif@fedora lab05]$ nasm -f elf lab5-1.asm
[spelovandreif@fedora lab05]$ ld -m elf_i386 -o lab5-1 lab5-1.o
[spelovandreif@fedora lab05]$ ./lab5-1
Введите строку:
Спелов Андрей Николаевич
[spelovandreif@fedora lab05]$
```

Рис. 3.7: Проверяем, как работает данная программа

Скачиваем файл со страницы курса (рис. 3.8).

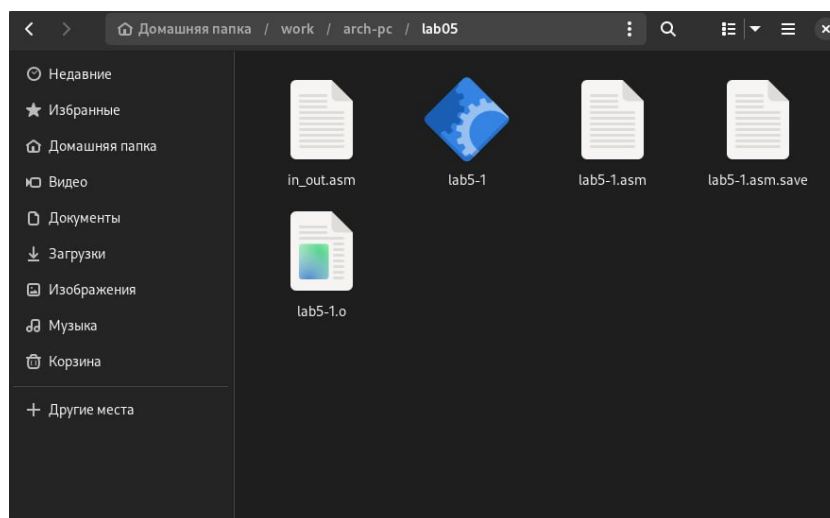


Рис. 3.8: Скачиваем файл

Копируем файл в нужную директорию (рис. 3.9).

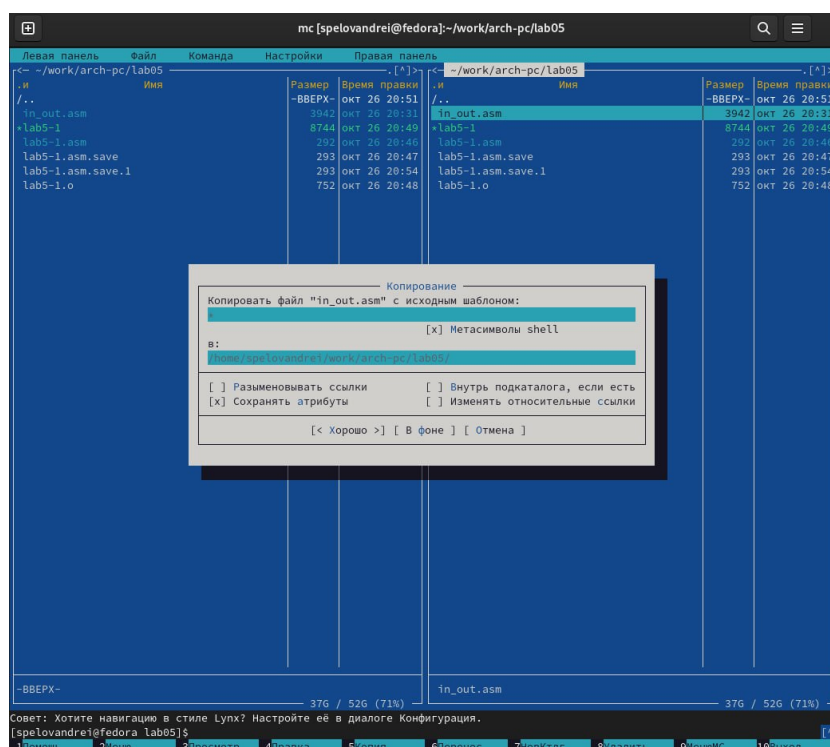


Рис. 3.9: Копируем скаченный файл

Создаем копию файла lab5-1.asm (рис. 3.10).

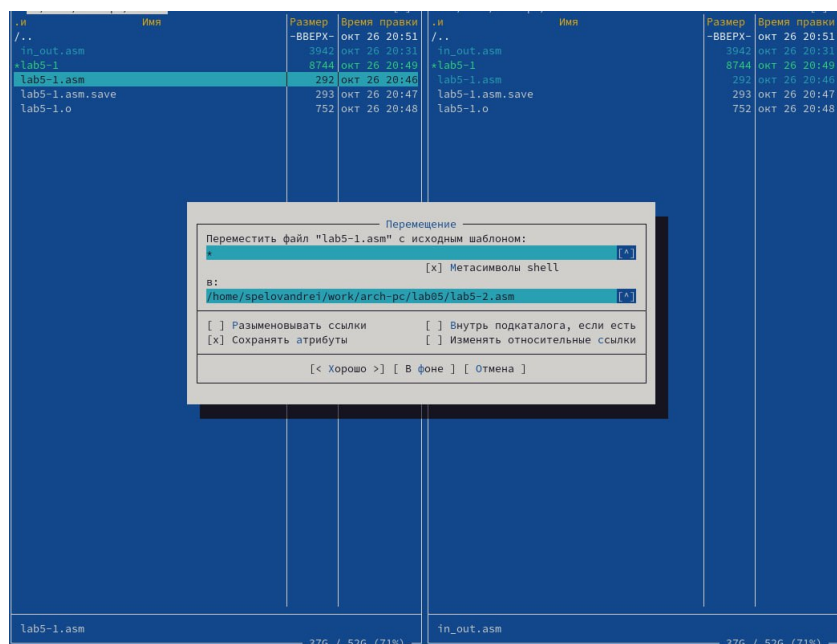


Рис. 3.10: Создаем копию файла клавишей F6

Проверяем созданный файл (рис. 3.11).

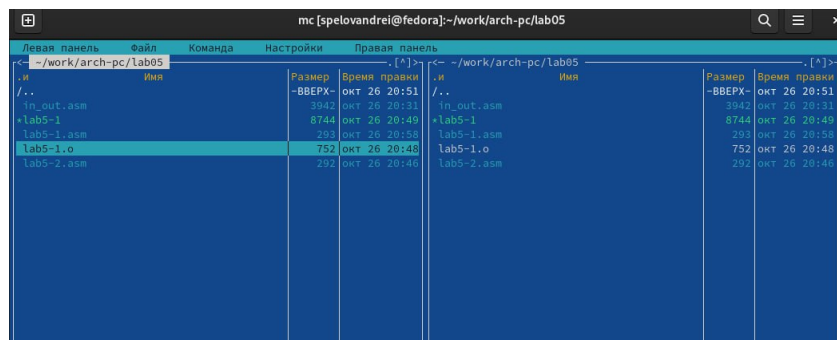
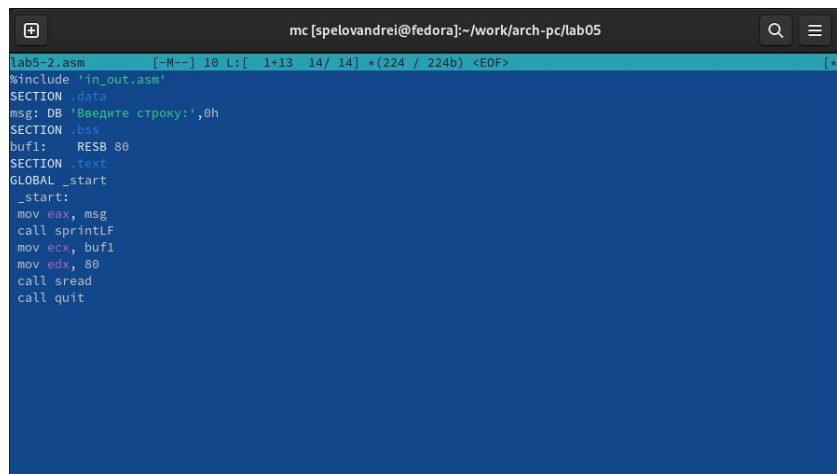


Рис. 3.11: Проверяем скопировался ли файл

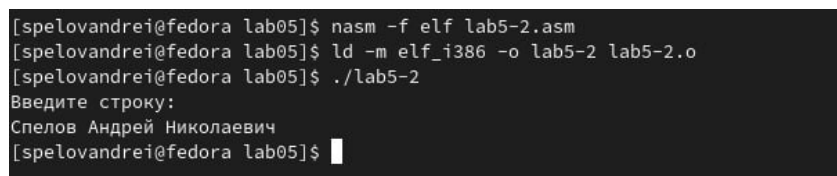
Открываем новый файл и заполняем его в соответствии с листингом (рис. 3.12).



```
mc [spelovandrei@fedora]:~/work/arch-pc/lab05
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 sprintf
mov ecx, buf1
mov edx, 80
call sread
call quit
```

Рис. 3.12: Открываем и заполняем файл

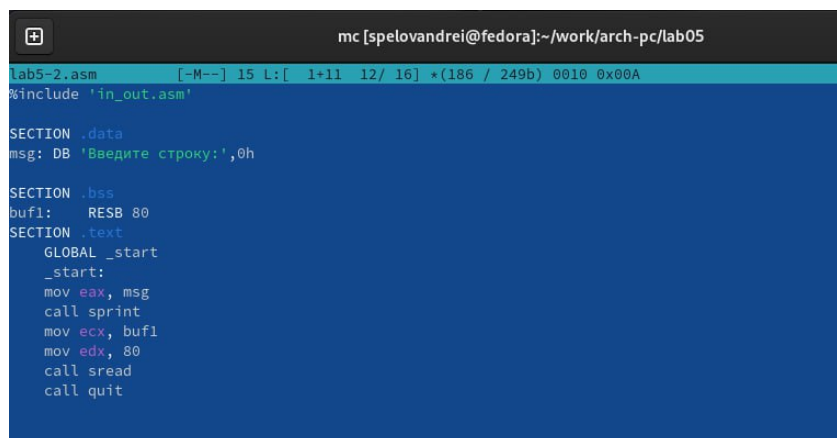
Транслируем и запускаем новый файл (рис. 3.13).



```
[spelovandrei@fedora lab05]$ nasm -f elf lab5-2.asm
[spelovandrei@fedora lab05]$ ld -m elf_i386 -o lab5-2 lab5-2.o
[spelovandrei@fedora lab05]$ ./lab5-2
Введите строку:
Спелов Андрей Николаевич
[spelovandrei@fedora lab05]$
```

Рис. 3.13: Смотрим, как сработала программа

Снова открываем файл для редактирования и меняем sprintf на sprintf (рис. 3.14).



```
mc [spelovandrei@fedora]:~/work/arch-pc/lab05
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 sprintf
mov ecx, buf1
mov edx, 80
call sread
call quit
```

Рис. 3.14: Редактируем файл

Транслируем и запускаем файл(рис. 3.15).

```
[spelovandrei@fedora lab05]$ nasm -f elf lab5-2.asm
[spelovandrei@fedora lab05]$ ld -m elf_i386 -o lab5-2 lab5-2.o
[spelovandrei@fedora lab05]$ ./lab5-2
Введите строку:Спелов Андрей Николаевич
[spelovandrei@fedora lab05]$
```

Рис. 3.15: 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. 3.16).

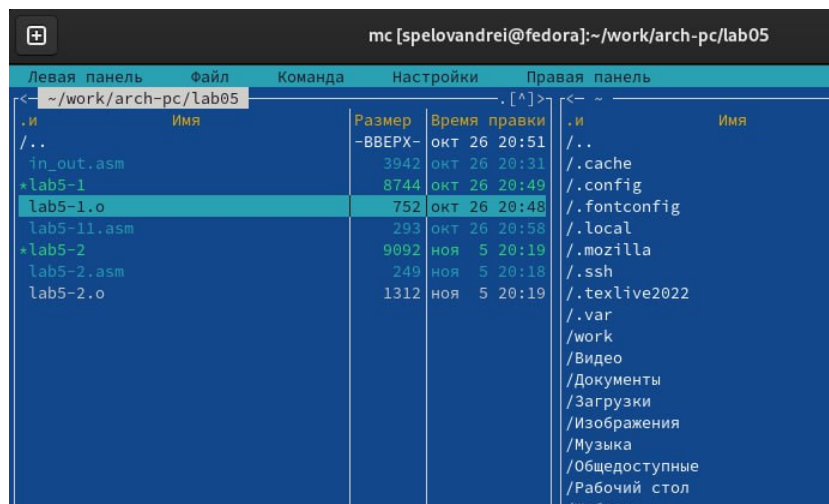
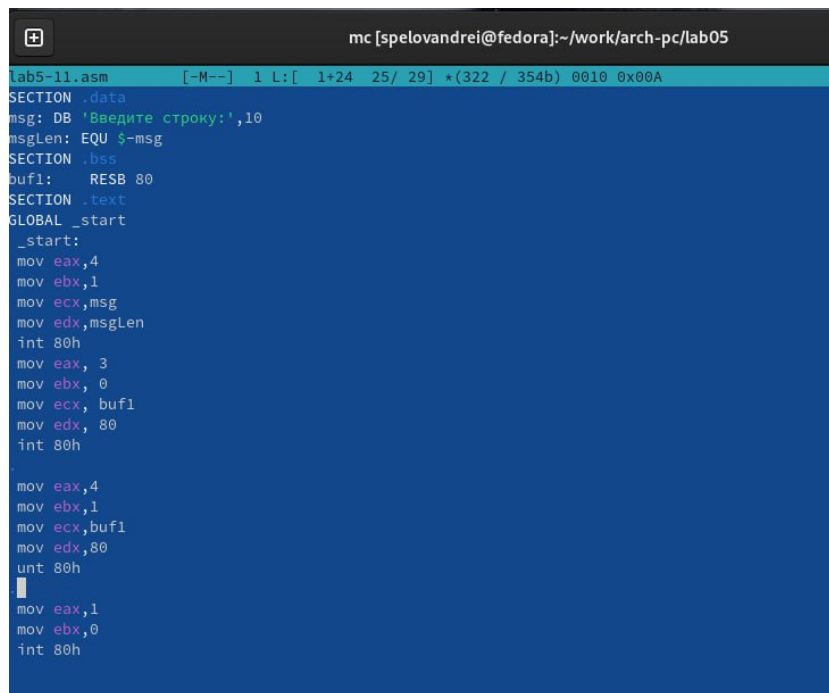


Рис. 3.16: Создаем копию файла `lab5-1.asm`

Редактируем файл, чтобы введенный текст с клавиатуры выводился в консоль (рис. 3.17).



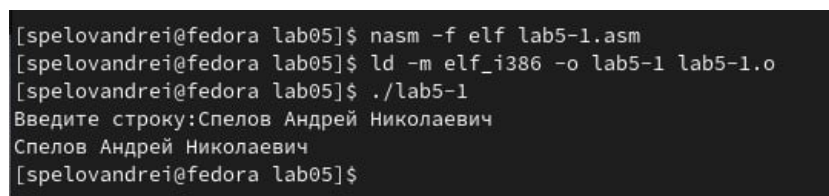
```
mc [spelovandrei@fedora]: ~/work/arch-pc/lab05
lab5-1.asm  [-M--]  1 L:[ 1+24  25/ 29] *(322 / 354b) 0010 0x00A
SECTION .data
msg: DB 'Введите строку:',10
msglen: EQU $-msg
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msglen
int 80h
mov eax,3
mov ebx,0
mov ecx,buf1
mov edx,80
int 80h

mov eax,4
mov ebx,1
mov ecx,buf1
mov edx,80
int 80h

mov eax,1
mov ebx,0
int 80h
```

Рис. 3.17: Редактируем файл

Транслируем файл и запускаем программу (рис. 3.18).



```
[spelovandrei@fedora lab05]$ nasm -f elf lab5-1.asm
[spelovandrei@fedora lab05]$ ld -m elf_i386 -o lab5-1 lab5-1.o
[spelovandrei@fedora lab05]$ ./lab5-1
Введите строку:Спелов Андрей Николаевич
Спелов Андрей Николаевич
[spelovandrei@fedora lab05]$
```

Рис. 3.18: Проверяем правильность написания программы

Создаем копию файла lab5-2.asm и называем его так же (рис. 3.19).

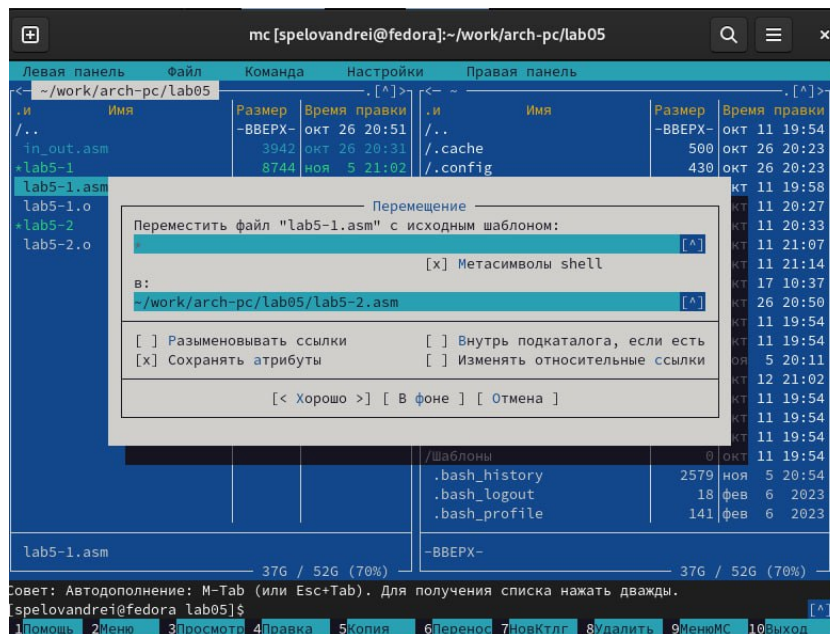


Рис. 3.19: Создаем копию файла lab5-2.asm

Редактируем файл, чтобы введенный текст с клавиатуры выводился в консоль (рис. 3.20).

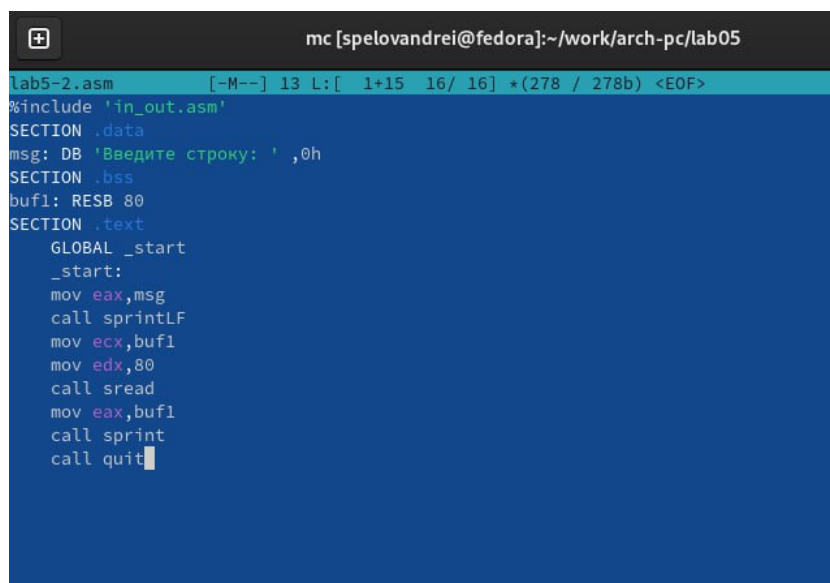


Рис. 3.20: Редактируем файл

Транслируем файл и запускаем программу (рис. 3.21).

```
[spelovandreifedora lab05]$ nasm -f elf lab5-2.asm
[spelovandreifedora lab05]$ ld -m elf_i386 -o lab5-2 lab5-2.o
[spelovandreifedora lab05]$ ./lab5-2
Введите строку:
Спелов Андрей Николаевич
Спелов Андрей Николаевич
[spelovandreifedora lab05]$
```

Рис. 3.21: Проверяем правильность написания программы

4 Выводы

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