

Отчёт по лабораторной работе №5

Дисциплина: архитектура компьютера

Репкина Елизавета Андреевна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	21
	Список литературы	22

Список иллюстраций

4.1 команда ms	8
4.2 Открытие каталога	9
4.3 создание папки	9
4.4 Переход в каталог	10
4.5 создание файла	10
4.6 Открытие файла для редактирования	11
4.7 ввод команды	11
4.8 сохранение файла	12
4.9 Открытие файла для просмотра	12
4.10 Исполнение файла	13
4.11 Скачанный файл	14
4.12 Копирование файла	14
4.13 Копирование файла	15
4.14 Редактирование файла	15
4.15 Исполнение файла	16
4.16 Изменение файла	16
4.17 Исполнение файла	17
4.18 Копирование файла	17
4.19 Редактирование файла	18
4.20 Выполнение программы	18
4.21 копирование файла	19
4.22 Редактирование файла	19
4.23 Выполнение программы	20

Список таблиц

1 Цель работы

Целью данной лабораторной работы является приобретение практических навыков работы в Midnight Commander, освоение инструкций языка ассемблера `mov` и `int`.

2 Задание

1. Основы работы с mc
2. Структура программы на языке ассемблера NASM
3. Подключение внешнего файла
4. Выполнение заданий для самостоятельной работы

3 Теоретическое введение

Midnight Commander (или просто mc) — это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управлению файловой системой, т.е. mc является файловым менеджером. Midnight Commander позволяет сделать работу с файлами более удобной и наглядной. Программа на языке ассемблера NASM, как правило, состоит из трёх секций: секция кода программы (SECTION .text), секция инициированных (известных во время компиляции) данных (SECTION .data) и секция неинициализированных данных (тех, под которые во время компиляции только отводится память, а значение присваивается в ходе выполнения программы) (SECTION .bss). Директивы используются для объявления простых переменных и для объявления массивов. Для определения строк принято использовать директиву DB в связи с особенностями хранения данных в оперативной памяти

4 Выполнение лабораторной работы

Основы работы с mc

Открываю Midnight Commander(рис. 4.1)

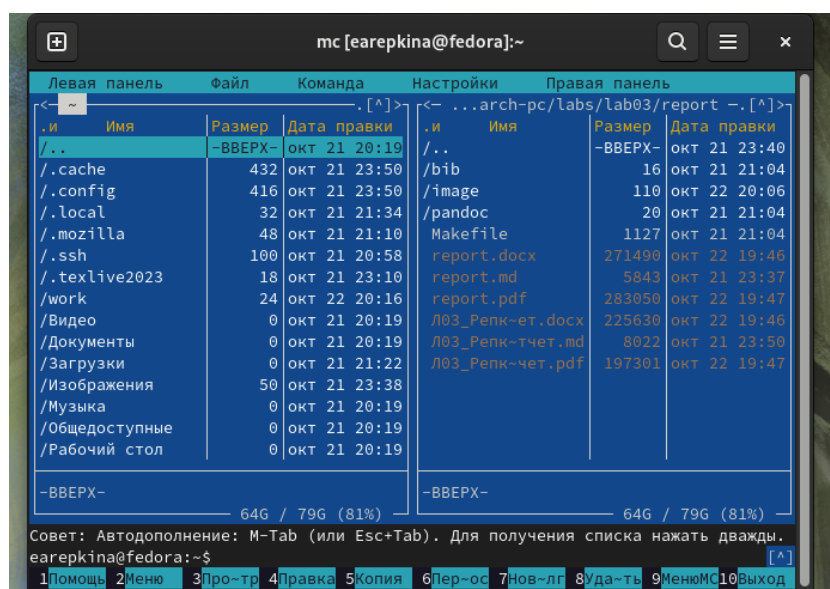


Рис. 4.1: команда mc

Пользуясь клавишами \uparrow , \downarrow и Enter перехожу в каталог `~/work/arch-rc` созданный при выполнении лабораторной работы №4 (рис. 4.2)

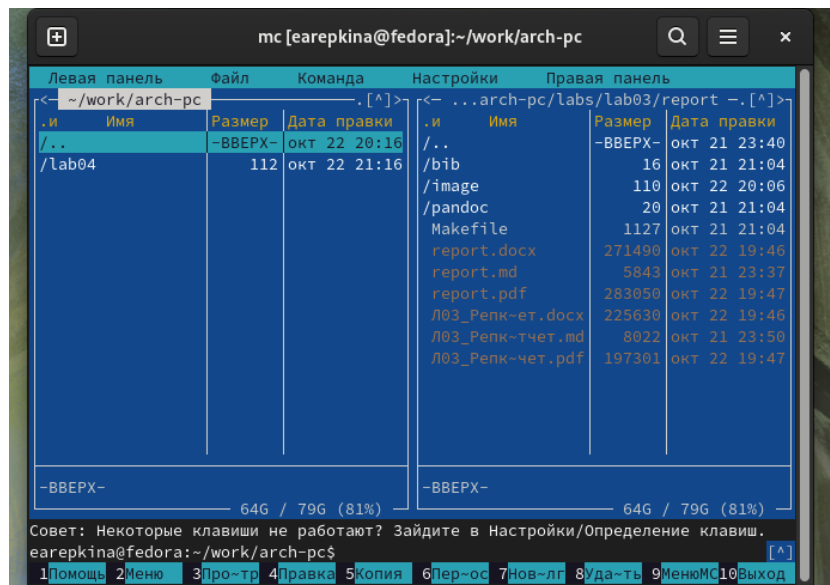


Рис. 4.2: Открытие каталога

Структура программы на языке ассемблера NASM

С помощью функциональной клавиши F7 создаю папку lab05(рис. 4.3)

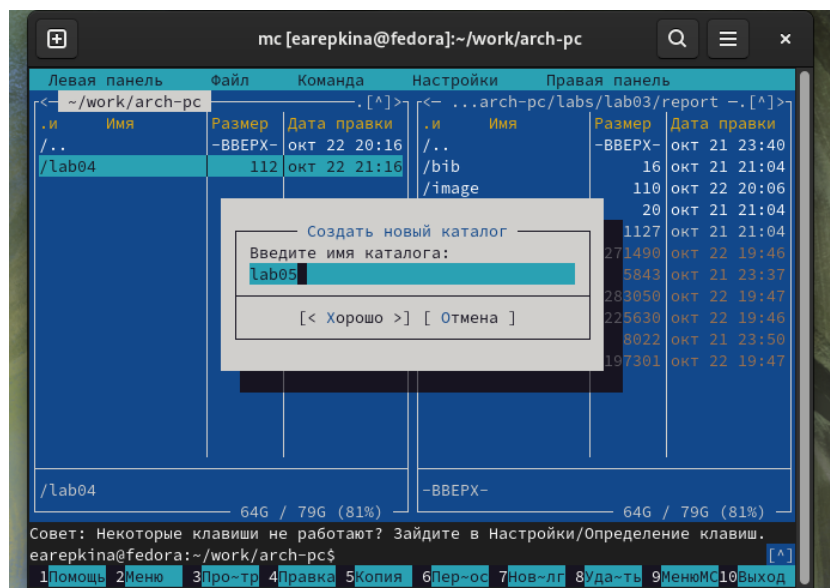


Рис. 4.3: создание папки

перехожу в созданный каталог(рис. 4.4)

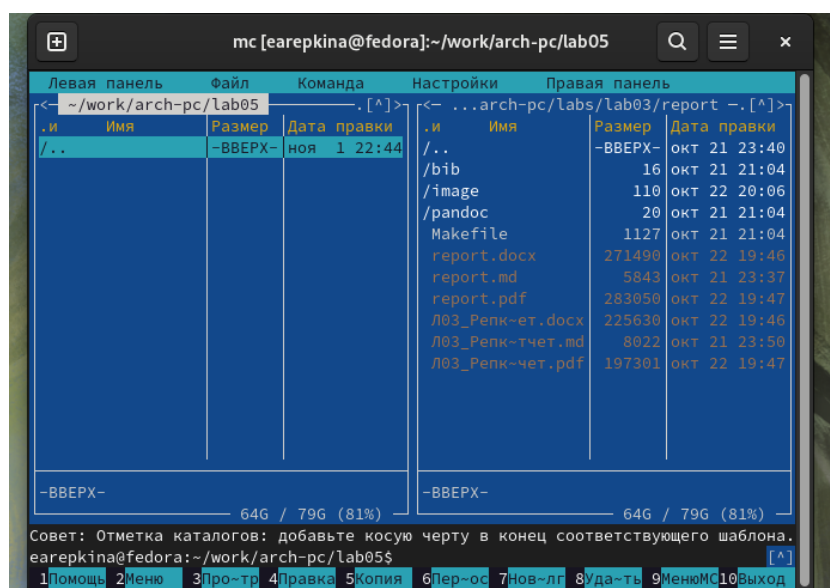


Рис. 4.4: Переход в каталог

Пользуясь строкой ввода и командой touch создаю файл lab5-1.asm (рис. 4.5)

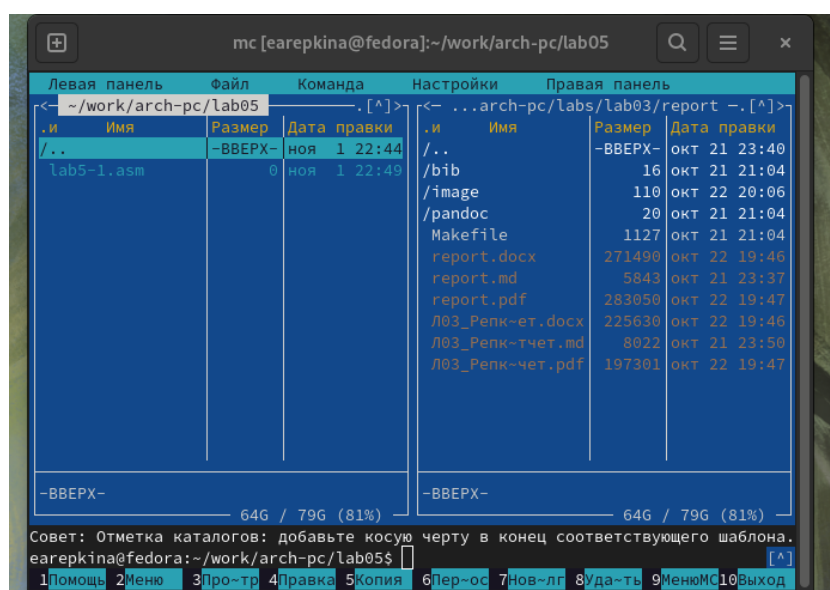


Рис. 4.5: создание файла

С помощью функциональной клавиши F4 открываю файл lab5-1.asm для редактирования во встроенном редакторе. Как правило в качестве встроенного редактора Midnight Commander используется редакторы nano или mcedit (рис. 4.6)

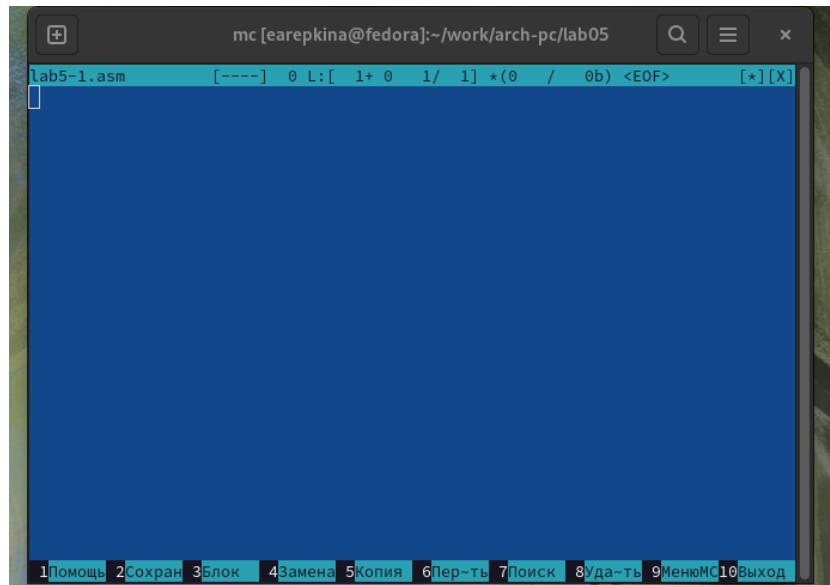


Рис. 4.6: Открытие файла для редактирования

Ввожу текст программы из листинга 5.1 (рис. 4.7)

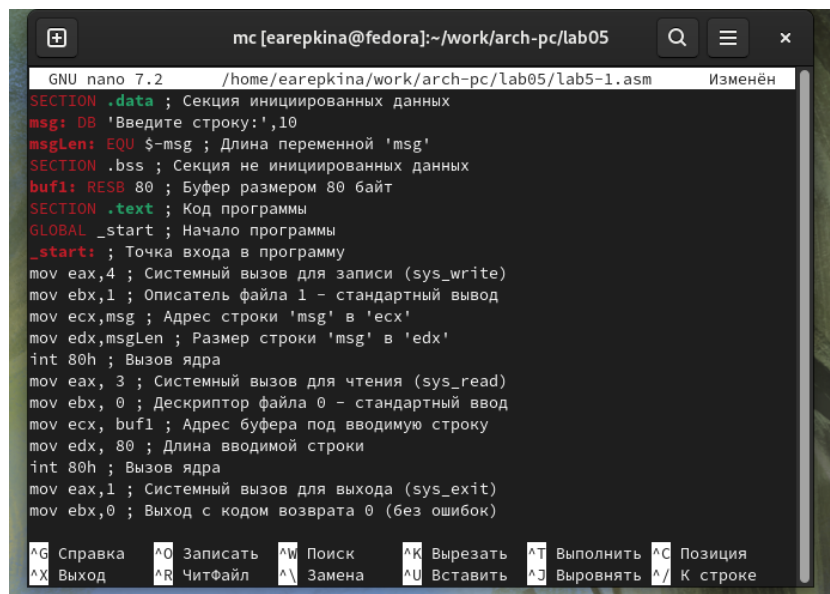


Рис. 4.7: ввод команды

сохраняю изменения и закрываю файл (рис. 4.8)

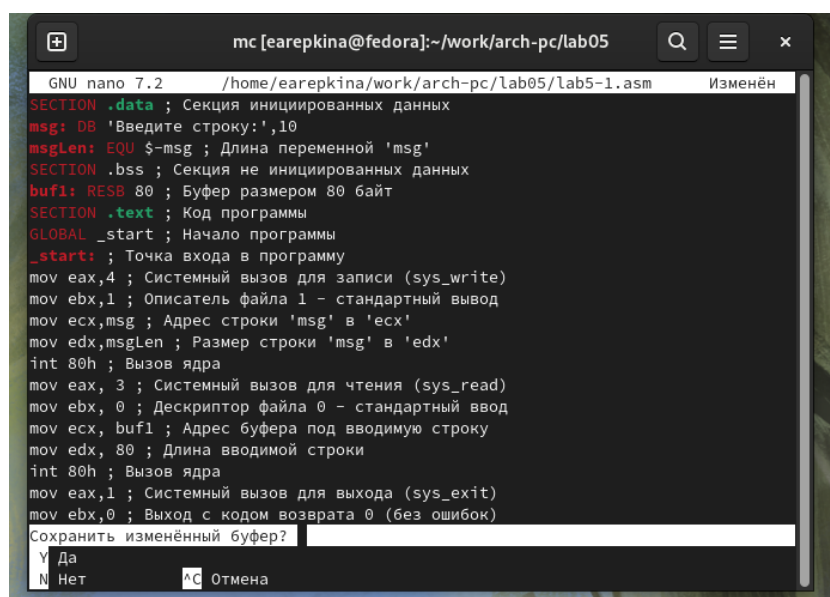


Рис. 4.8: сохранение файла

С помощью функциональной клавиши F3 открываю файл lab5-1.asm для просмотра. Убеждаюсь, что файл содержит текст программы. (рис. 4.9)

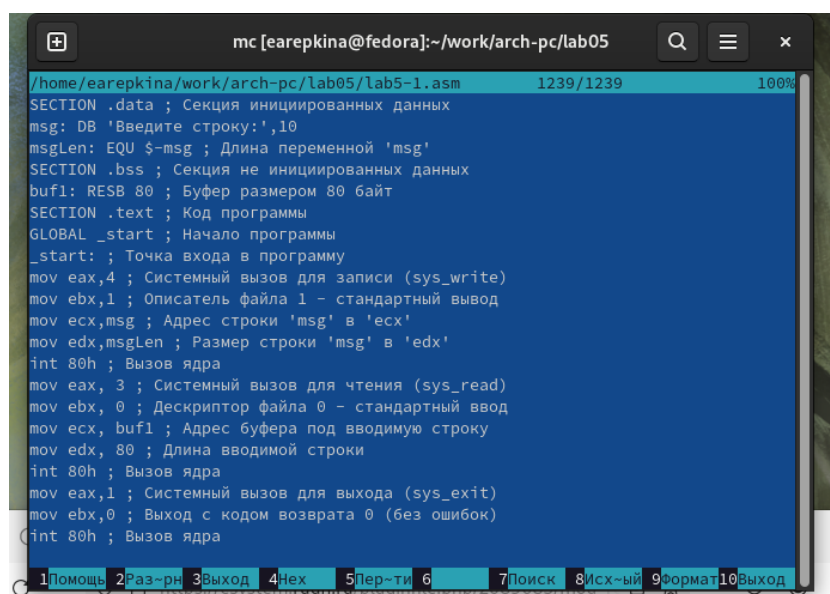
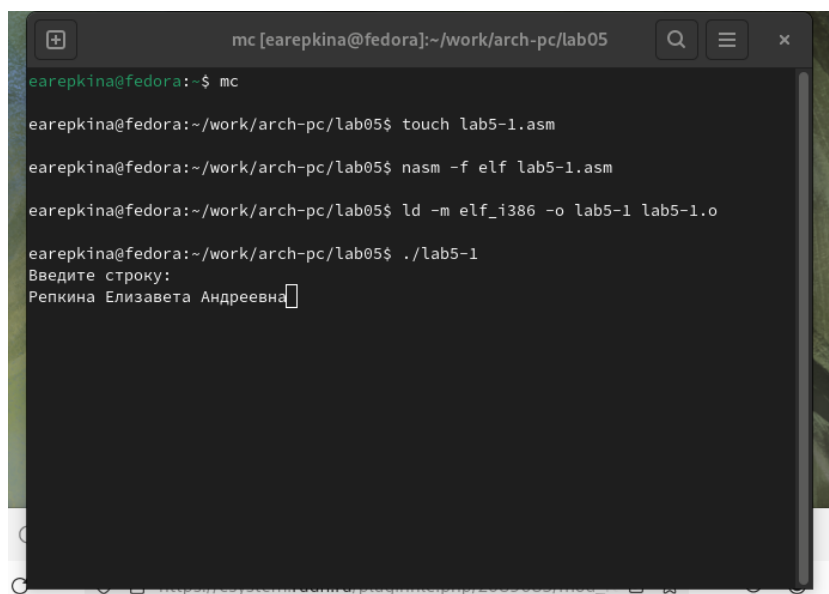


Рис. 4.9: Открытие файла для просмотра

Открываю файл и убеждаюсь, что файл содержит текст программы. Транслирую текст программы файла в объектный файл командой `nasm -f elf lab5-`

1.asm. Создался объектный файл lab5-1.o. Выполняю компоновку объектного файла с помощью команды `ld -m elf_i386 -o lab5-1 lab5-1.o`. Создался исполняемый файл lab5-1.

Запускаю исполняемый файл. Программа выводит строку “Введите строку:” и ждет ввода с клавиатуры, я ввожу свои ФИО, на этом программа заканчивает свою работу (рис. 4.10)



```
mc [earepkina@fedora]:~/work/arch-pc/lab05
earepkina@fedora:~$ mc
earepkina@fedora:~/work/arch-pc/lab05$ touch lab5-1.asm
earepkina@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
earepkina@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
earepkina@fedora:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Репкина Елизавета Андреевна
```

Рис. 4.10: Исполнение файла

Подключение внешнего файла

Скачиваю файл `in_out.asm` со страницы курса в ТУИС. (рис. 4.11)

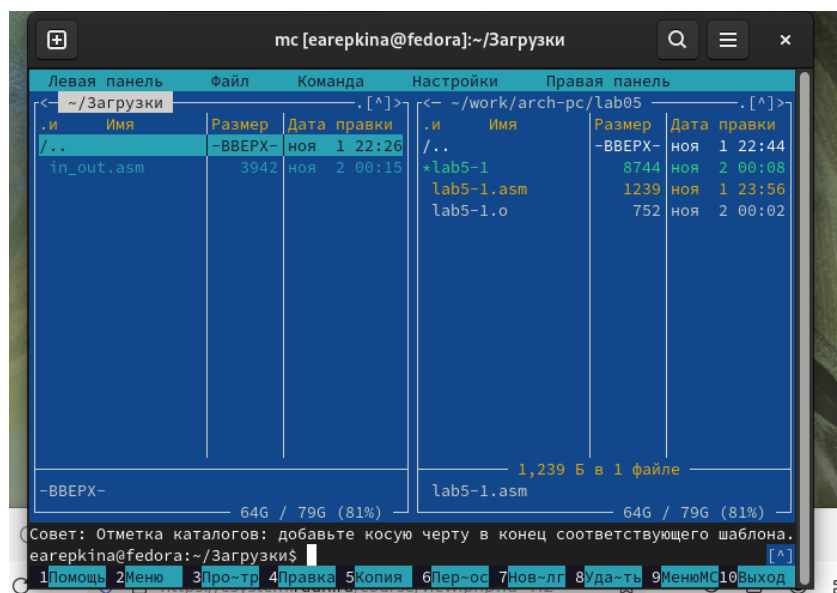


Рис. 4.11: Скачанный файл

Копирую файл in_out.asm из каталога Загрузки в созданный каталог lab05 (рис. 4.12)

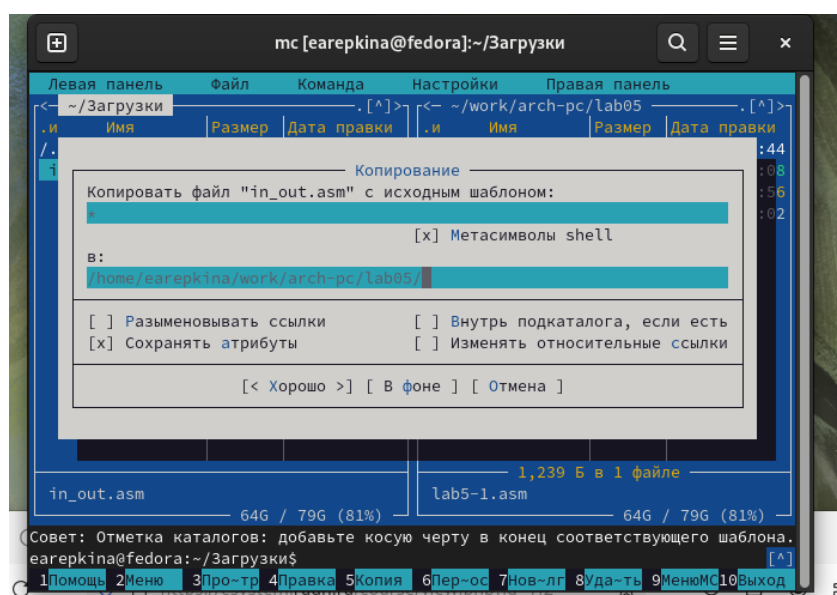


Рис. 4.12: Копирование файла

С помощью функциональной клавиши F5 копирую файл lab5-1 в тот же каталог, но с другим именем, для этого в появившемся окне mc прописываю имя для

копии файла (рис. 4.13)

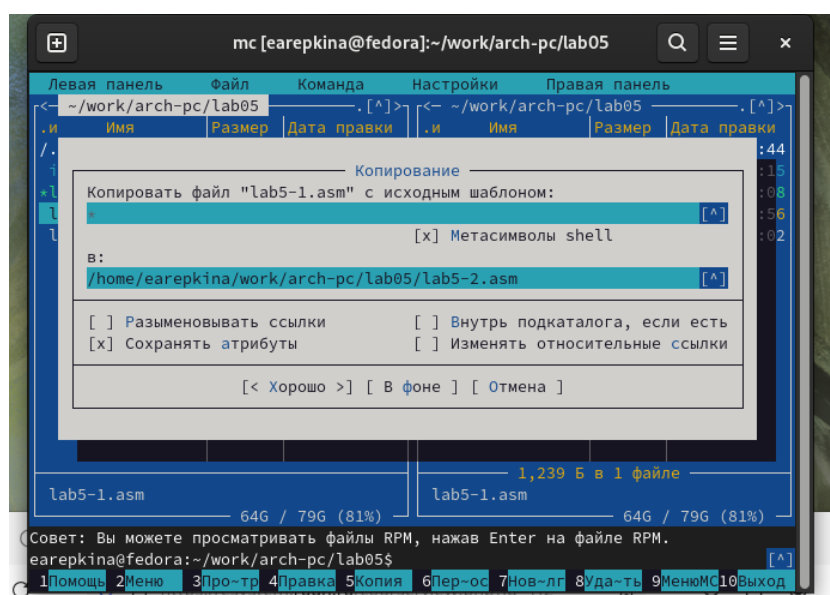


Рис. 4.13: Копирование файла

Изменяю содержимое файла lab5-2.asm во встроенном редакторе nano, чтобы в программе использовались подпрограммы из внешнего файла in_out.asm. (рис. 4.14)

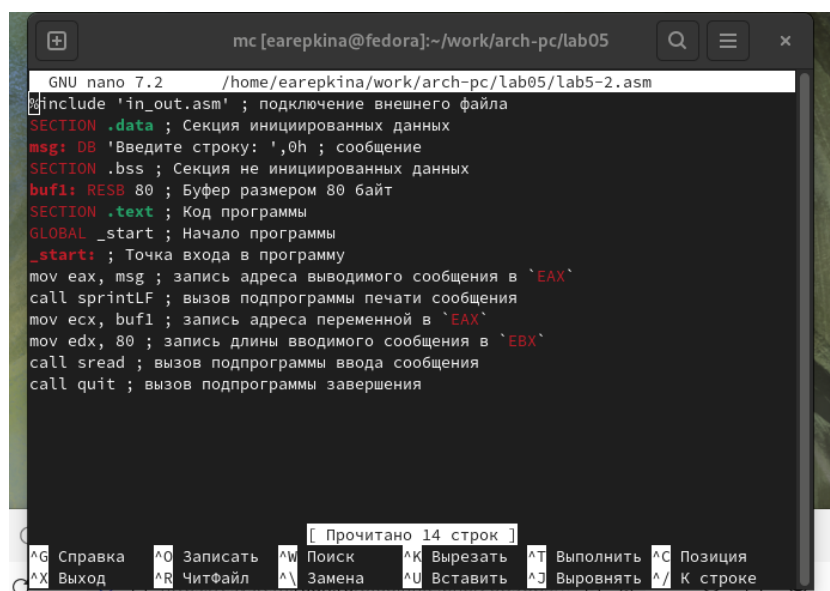
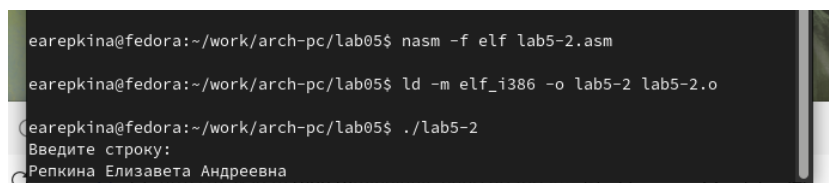


Рис. 4.14: Редактирование файла

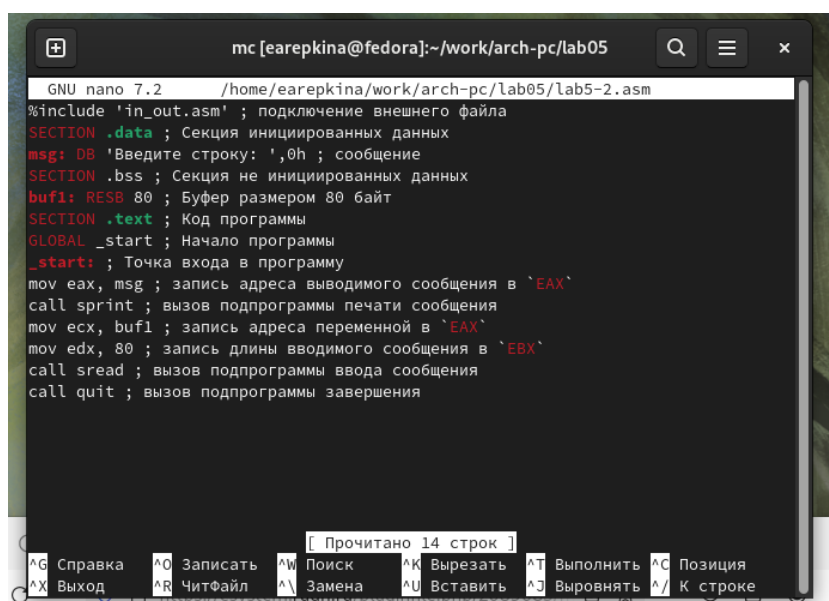
Транслирую текст программы файла в объектный файл командой `nasm -f elf lab5-2.asm`. Создался объектный файл `lab5-2.o`. Выполняю компоновку объектного файла с помощью команды `ld -m elf_i386 -o lab5-2 lab5-2.o`. Создался исполняемый файл `lab5-2`. Запускаю исполняемый файл. (рис. 4.15)



```
earepkina@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
earepkina@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
earepkina@fedora:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Репкина Елизавета Андреевна
```

Рис. 4.15: Исполнение файла

Открываю файл `lab5-2.asm` для редактирования в `nano`. Изменяю в нем подпрограмму `sprintLF` на `sprint`. Сохраняю изменения и открываю файл для просмотра, чтобы проверить сохранение действий (рис. 4.16)



```
GNU nano 7.2 /home/earepkina/work/arch-pc/lab05/lab5-2.asm
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция инициированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не инициированных данных
buf1: RESB 80 ; Буфер размером 80 байт
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
mov eax, msg ; запись адреса выводимого сообщения в `EAX`
call sprint ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса переменной в `EAX`
mov edx, 80 ; запись длины вводимого сообщения в `EBX`
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения
```

Рис. 4.16: Изменение файла

Снова транслирую файл, выполняю компоновку созданного объектного файла, запускаю новый исполняемый файл (рис. 4.17)


```
earepkina@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
earepkina@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
earepkina@fedora:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Репкина Елизавета Андреевна
```

Рис. 4.17: Исполнение файла

Разница между первым исполняемым файлом lab6-2 и вторым lab6-2-2 в том, что запуск первого запрашивает ввод с новой строки, а программа, которая исполняется при запуске второго, запрашивает ввод без переноса на новую строку, потому что в этом заключается различие между подпрограммами `sprintLF` и `sprint`.

Выполнение заданий для самостоятельной работы

1. Создайте копию файла lab5-1.asm. (рис. 4.18)

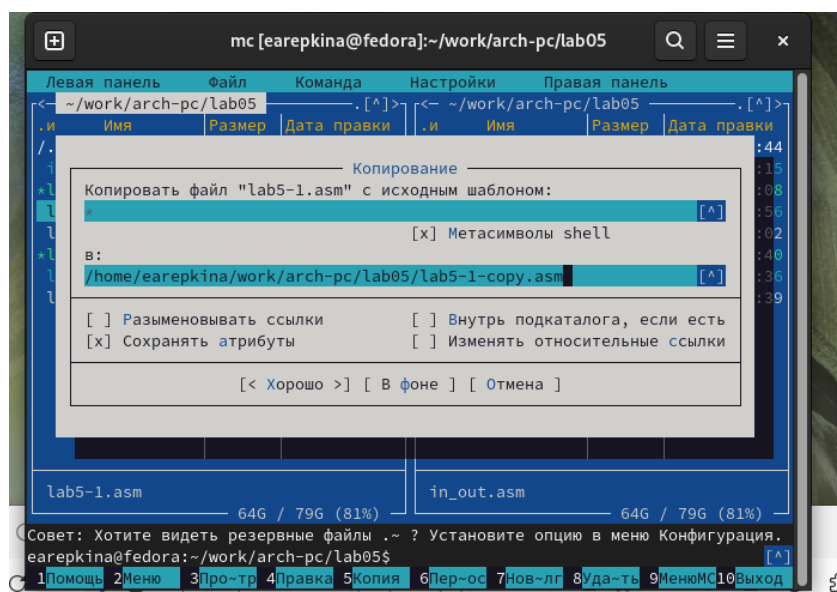
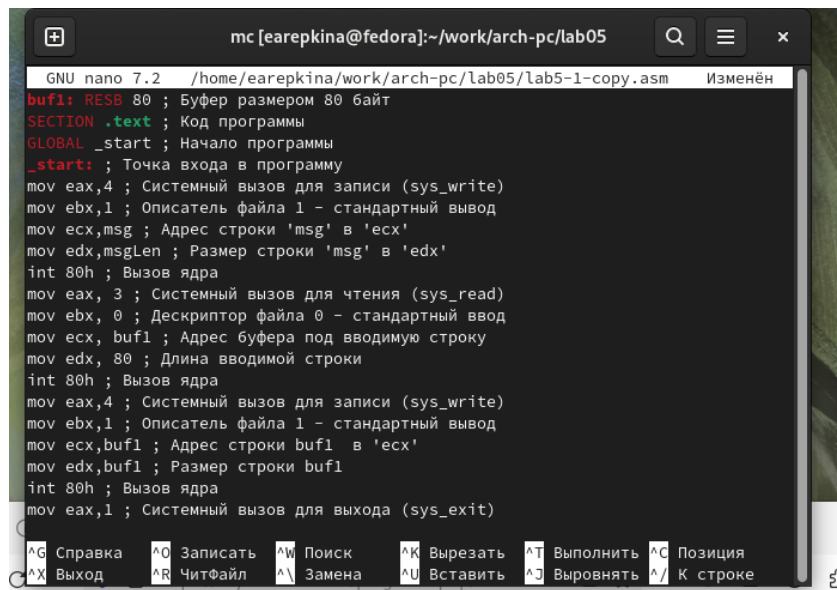


Рис. 4.18: Копирование файла

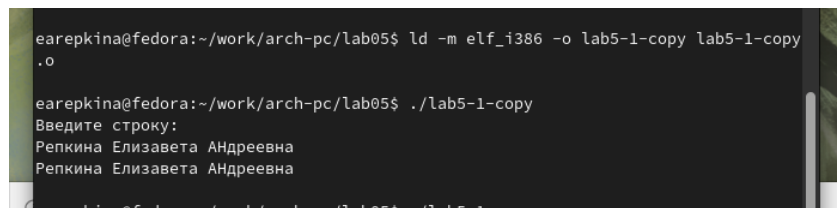
Внесите изменения в программу (без использования внешнего файла `in_out.asm`), так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введенную строку на экран. (рис. 4.19)



```
mc [earepkina@fedora]:~/work/arch-pc/lab05
GNU nano 7.2 /home/earepkina/work/arch-pc/lab05/lab5-1-copy.asm
buf1: RESB 80 ; Буфер размером 80 байт
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла 1 - стандартный вывод
mov ecx,msg ; Адрес строки 'msg' в 'ecx'
mov edx,msglen ; Размер строки 'msg' в 'edx'
int 80h ; Вызов ядра
mov eax,3 ; Системный вызов для чтения (sys_read)
mov ebx,0 ; Дескриптор файла 0 - стандартный ввод
mov ecx,buf1 ; Адрес буфера под вводимую строку
mov edx,80 ; Длина вводимой строки
int 80h ; Вызов ядра
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла 1 - стандартный вывод
mov ecx,buf1 ; Адрес строки buf1 в 'ecx'
mov edx,buf1 ; Размер строки buf1
int 80h ; Вызов ядра
mov eax,1 ; Системный вызов для выхода (sys_exit)
```

Рис. 4.19: Редактирование файла

2. Получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите свою фамилию.(рис. 4.20)



```
earepkina@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1-copy lab5-1-copy.o
earepkina@fedora:~/work/arch-pc/lab05$ ./lab5-1-copy
Введите строку:
Репкина Елизавета Андреевна
Репкина Елизавета Андреевна
```

Рис. 4.20: Выполнение программы

3. Создайте копию файла lab5-2.asm. (рис. 4.21)

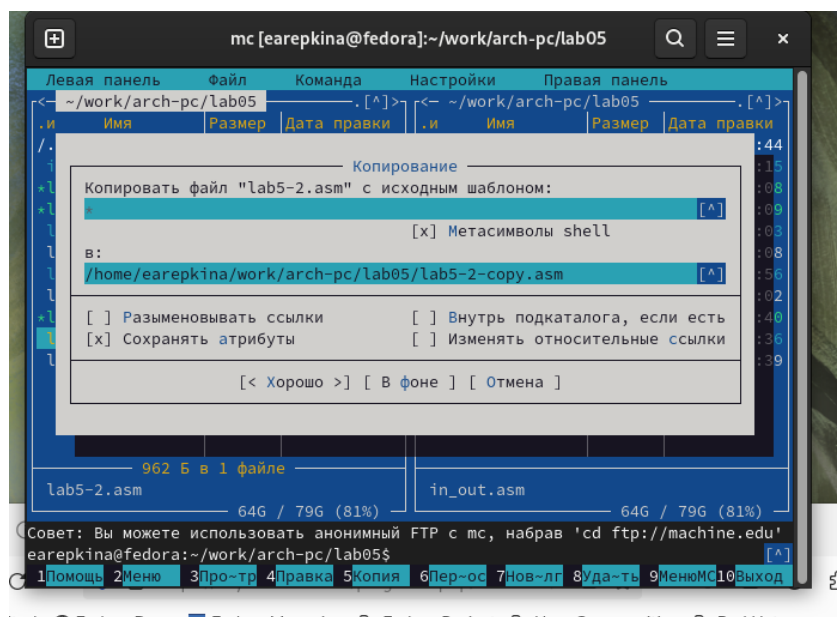


Рис. 4.21: копирование файла

Исправьте текст программы с использованием подпрограмм из внешнего файла in_out.asm, (рис. 4.22) так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введенную строку на экран. (рис. 4.23)

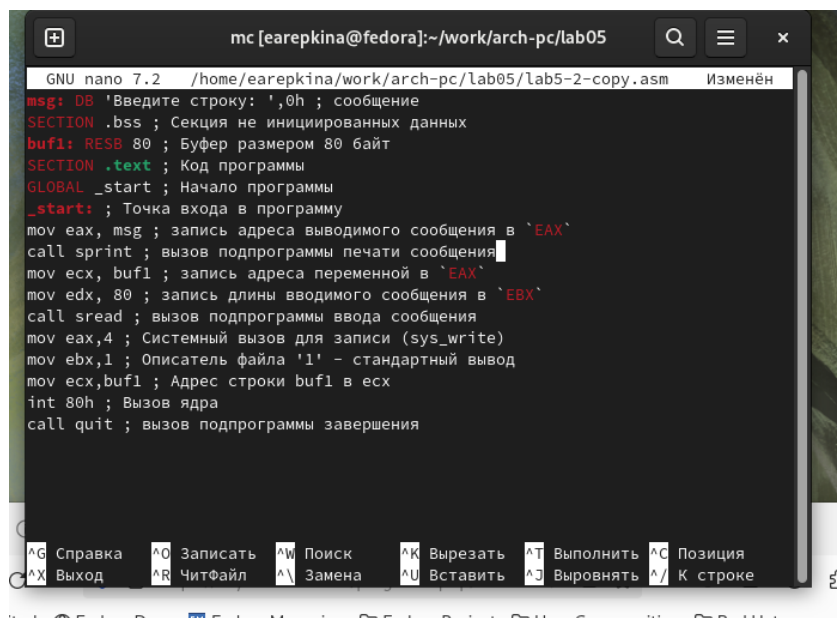


Рис. 4.22: Редактирование файла

```
earepkina@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-2-copy.asm
earepkina@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2-copy lab5-2-copy.o
earepkina@fedora:~/work/arch-pc/lab05$ ./lab5-2-copy
Введите строку: Репкина Елизавета Андреевна
Репкина Елизавета Андреевна
```

Рис. 4.23: Выполнение программы

5 Выводы

При выполнении данной лабораторной работы я приобрела практические навыки работы в Midnight Commander, а также освоила инструкции языка ассемблера `mov` и `int`.

Список литературы

1. GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>.
2. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>.
3. Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>.
4. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: <http://www.amazon.com/Learningbash-Shell-Programming-Nutshell/dp/0596009658>.
6. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс,
- 11.
12. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
13. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.
14. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВПетербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
15. Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-

- е изд. — М. : МАКС Пресс, 2011. — URL: http://www.stolyarov.info/books/asm_unix.
16. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
 17. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер,
 18. — 1120 с. — (Классика Computer Science).