

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5

дисциплина: Архитектура компьютера

Студент: Володина Алиса Алексеевна

Группа: НКАбд-01-25

МОСКВА

2025 г.

Содержание

Оглавление

1)Цель работы	5
2)Задания	6
3)Теоретическое введение	6
4)Выполнение лабораторной работы	8
5)Вывод	22
Список литературы	23

Содержание иллюстраций

Рисунок 0.1	8
Рисунок 0.2	8
Рисунок 0.3	9
Рисунок 0.4	9
Рисунок 0.5	9
Рисунок 0.6	10
Рисунок 0.7	10
Рисунок 0.8	10
Рисунок 0.9	11
Рисунок 0.10	11
Рисунок 0.11	12
Рисунок 0.12	12
Рисунок 0.13	12
Рисунок 0.14	12
Рисунок 0.15	13
Рисунок 0.16	13
Рисунок 0.17	14
Рисунок 0.18	14
Рисунок 0.19	15
Рисунок 0.20	15
Рисунок 0.21	15
Рисунок 0.22	15
Рисунок 0.23	16
Рисунок 0.24	16
Рисунок 0.25	16
Рисунок 0.26	17
Рисунок 0.27	17
Рисунок 0.28	17
Рисунок 0.29	18
Рисунок 0.30	18
Рисунок 0.31	18

Рисунок 0.32	18
Рисунок 0.33	19
Рисунок 0.34	19
Рисунок 0.35	20
Рисунок 0.36	20
Рисунок 0.37	20
Рисунок 0.38	20
Рисунок 0.39	21
Рисунок 0.40	21

1)Цель работы

Приобретение практических навыков работы в Midnight Commander.
Освоение инструкций языка ассемблера mov и int

2)Задания

Основы работы с тс, структура программы на языке ассемблера NASM, подключение внешнего файла, 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). Для объявления инициированных данных в секции .data используются директивы DB, DW, DD, DQ и DT, которые резервируют память и указывают, какие значения должны храниться в этой памяти: - DB (define byte) — определяет переменную размером в 1 байт; - DW (define word) — определяет переменную размером в 2 байта (слово); - DD (define double word) — определяет переменную размером в 4 байта (двойное слово); - DQ (define quad word) — определяет переменную размером в 8 байт (учетверённое слово); - DT (define ten bytes) — определяет переменную размером в 10 байт. 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. Инструкция языка ассемблера mov предназначена для дублирования данных источника в приёмнике.

```
mov dst,src
```

4)Выполнение лабораторной работы

1. Откроем Midnight Commander (рисунок 0.1)

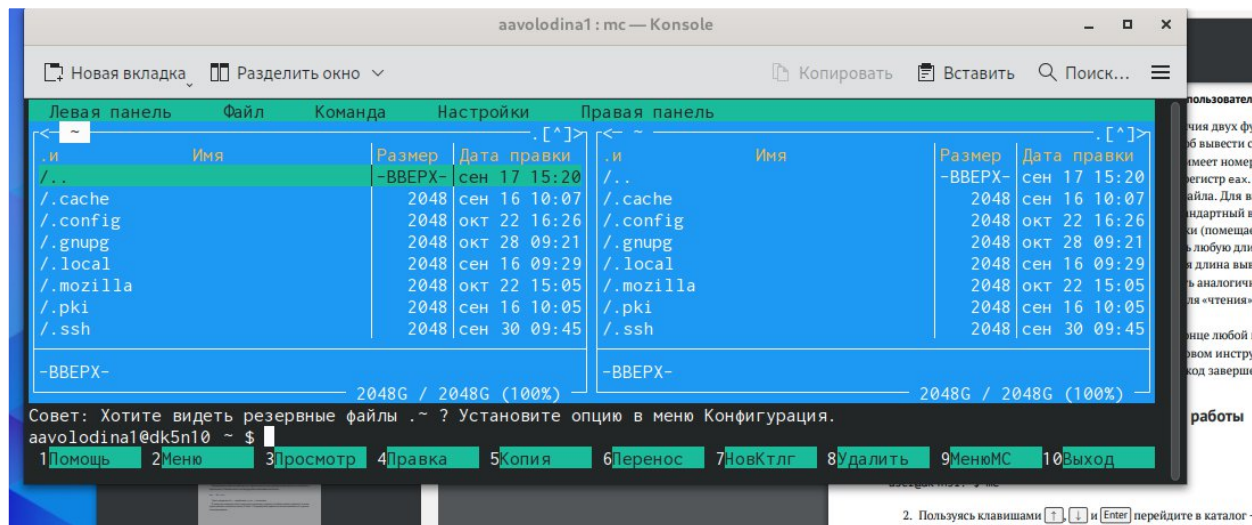


Рисунок 0.1

2. Пользуясь клавишами ↑, ↓ и Enter перейдем в каталог ~/work/arch-pc созданный при выполнении лабораторной работы №4. (рисунок 0.2-0.3)

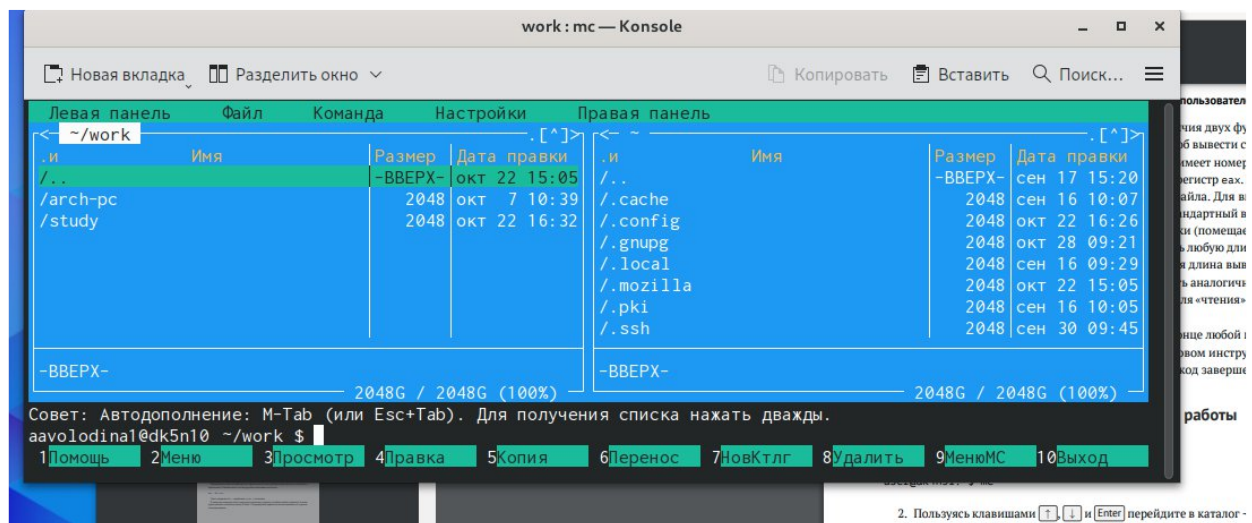


Рисунок 0.2

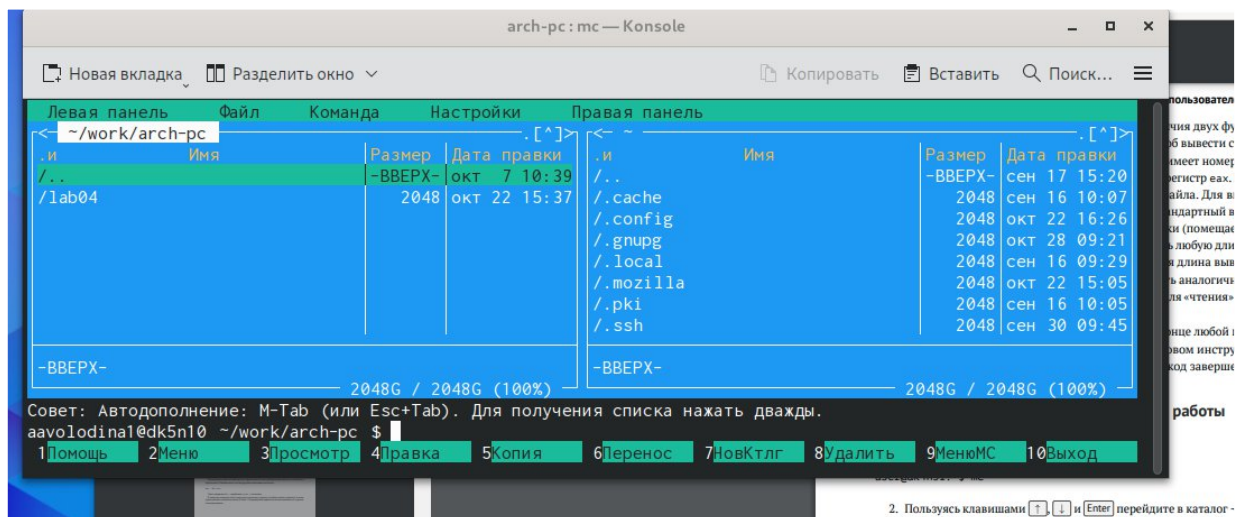


Рисунок 0.3

3. С помощью функциональной клавиши f7 создадим папку lab05 и перейдем в созданный каталог.(рисунок 0.4-0.5)

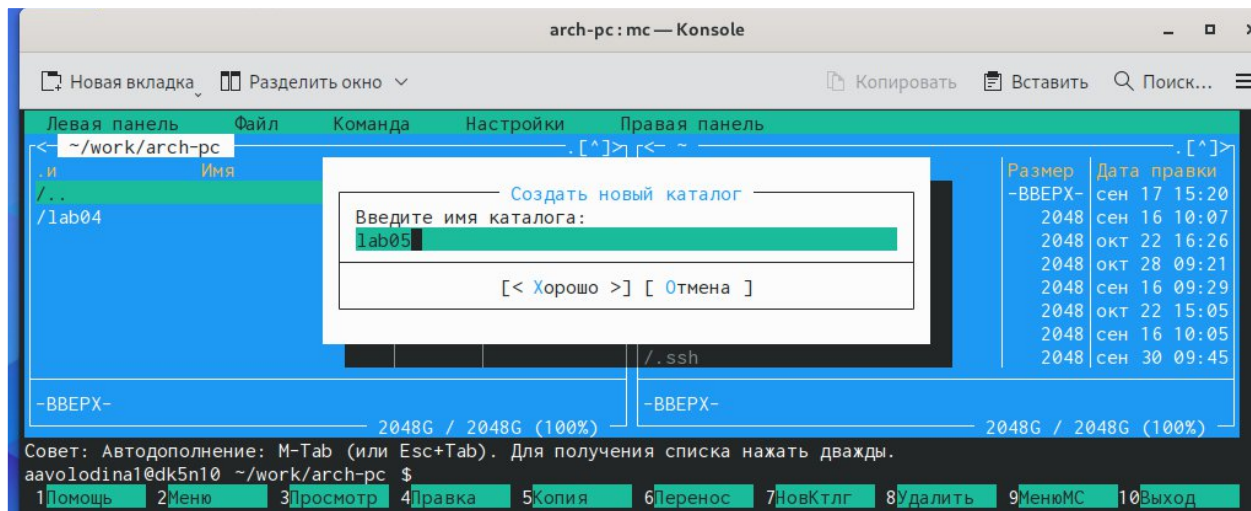


Рисунок 0.4

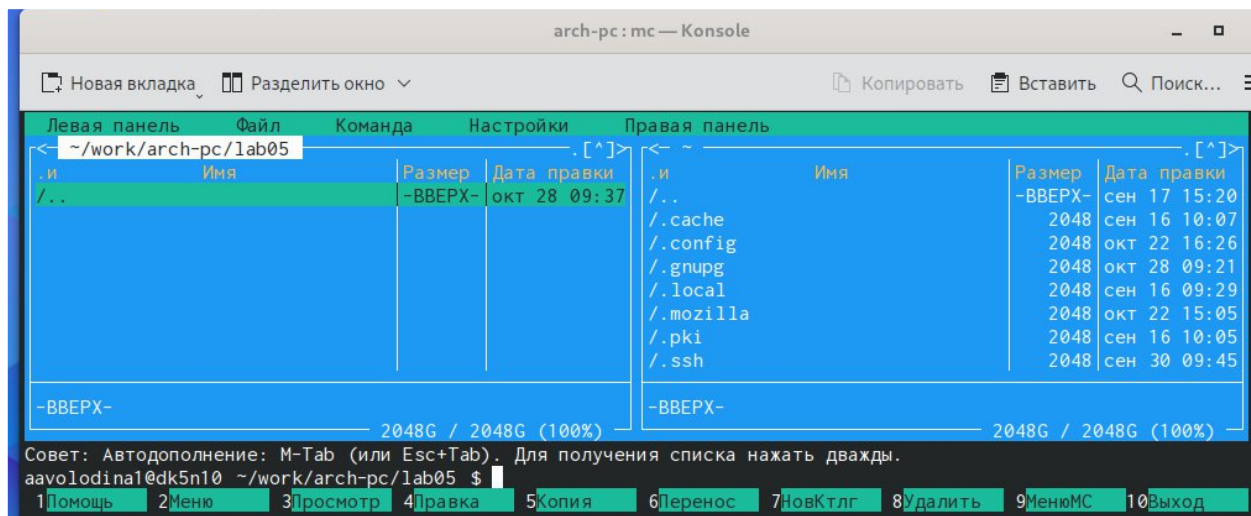


Рисунок 0.5

4. Пользуясь строкой ввода и командой touch создадим файл lab5-1.asm (рисунок 0.6)

```
aavolodina1@dk5n10 ~ $ touch lab5-1.asm
aavolodina1@dk5n10 ~ $
```

Рисунок 0.6

5. С помощью функциональной клавиши F4 откроем файл lab5-1.asm для редактирования во встроенном редакторе. (рисунок 0.7-0.9)

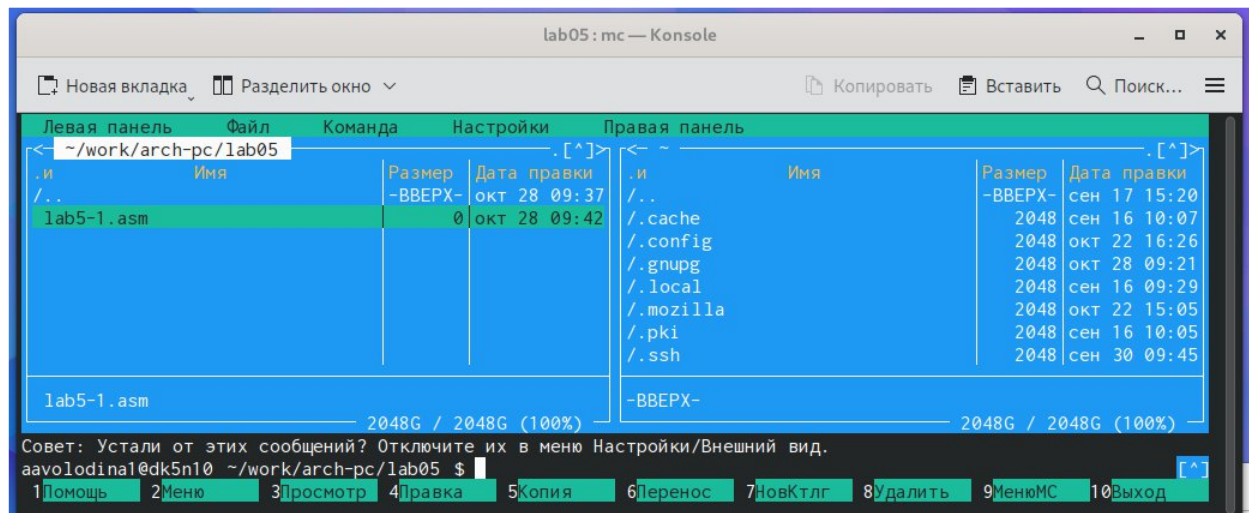


Рисунок 0.7

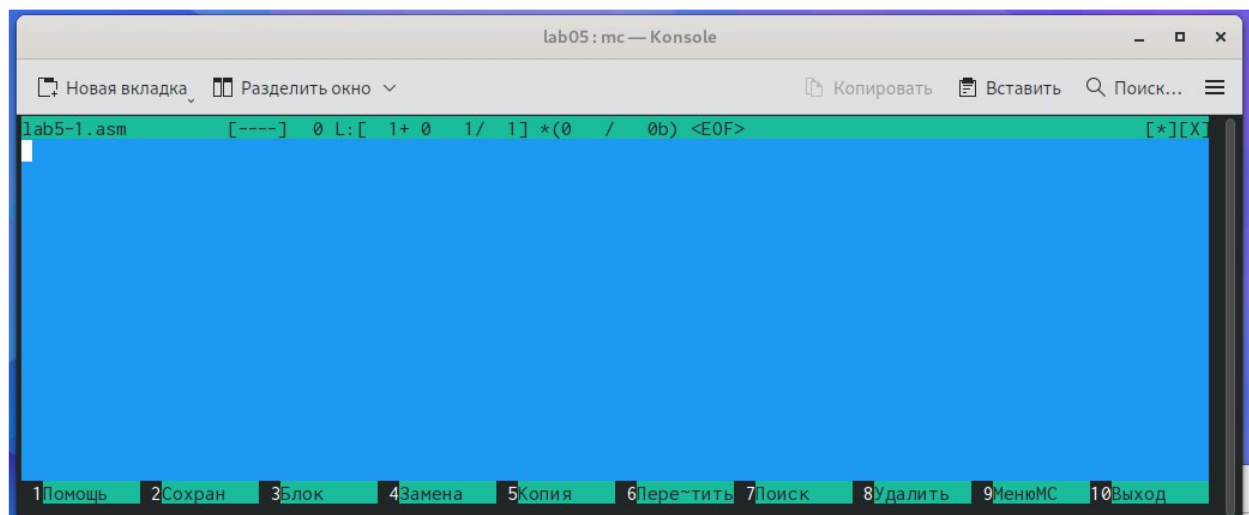
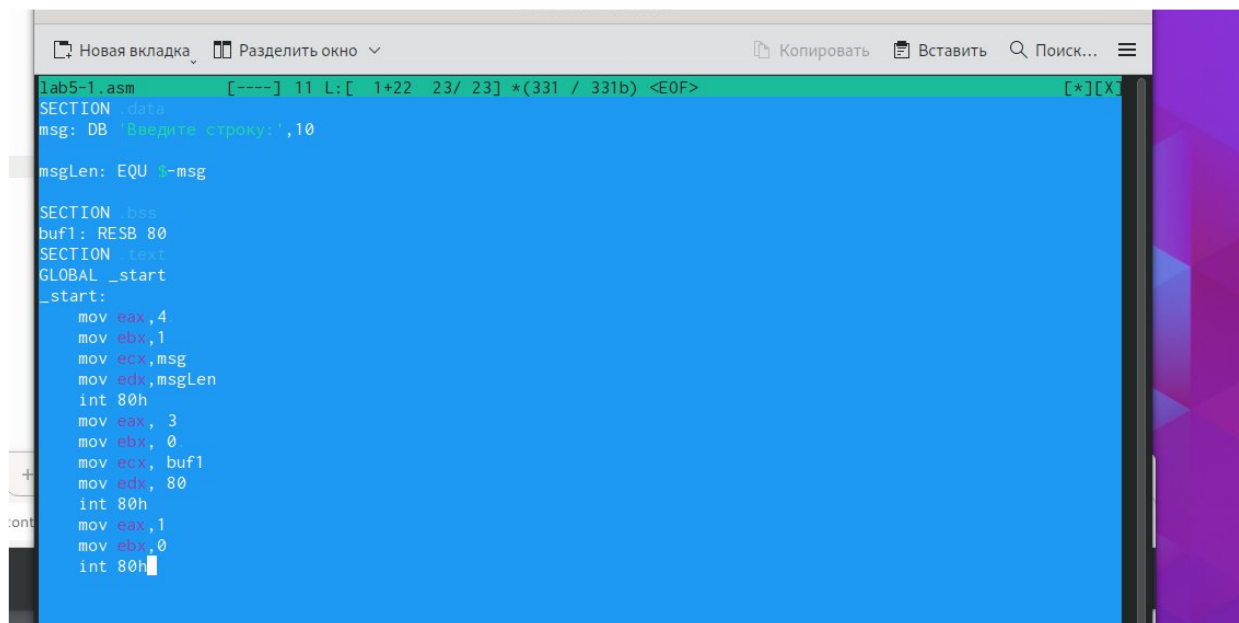


Рисунок 0.8

6. Введем текст программы из листинга и сохраним изменения и закроем файл.



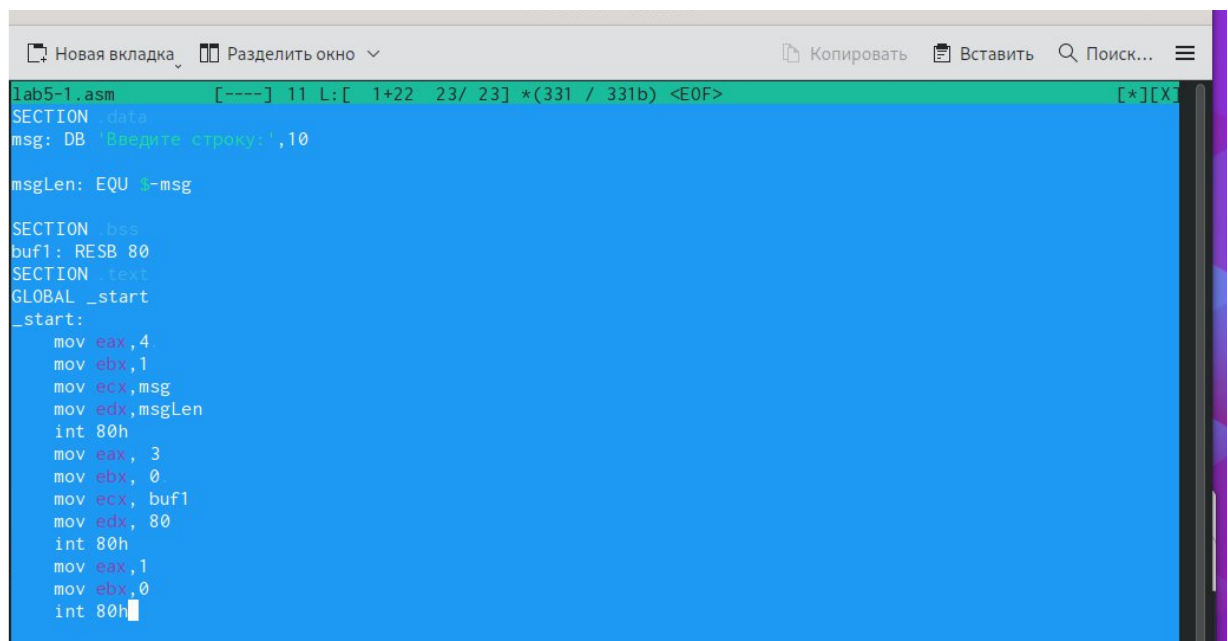
```
lab5-1.asm [----] 11 L: [ 1+22 23/ 23] *(331 / 331b) <E0F> [*][X]
SECTION .data
msg: DB 'Введите строку:',10

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
```

Рисунок 0.9

7. С помощью функциональной клавиши f3 откроем файл lab5-1.asm для просмотра. Убедимся, что файл содержит текст программы (рисунок 0.10)



```
lab5-1.asm [----] 11 L: [ 1+22 23/ 23] *(331 / 331b) <E0F> [*][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
```

Рисунок 0.10

8. Оттранслируем текст программы lab5-1.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл. Программа выводит строку 'Введите строку:' и ожидает ввода с клавиатуры. На запрос введем ФИО (рисунок 0.11-0.15)

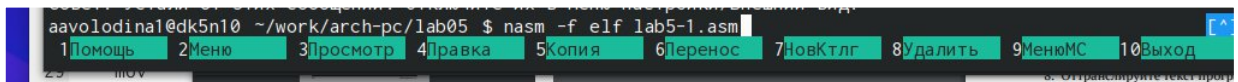


Рисунок 0.11

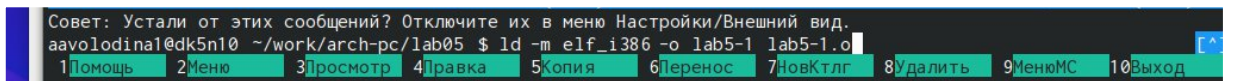


Рисунок 0.12

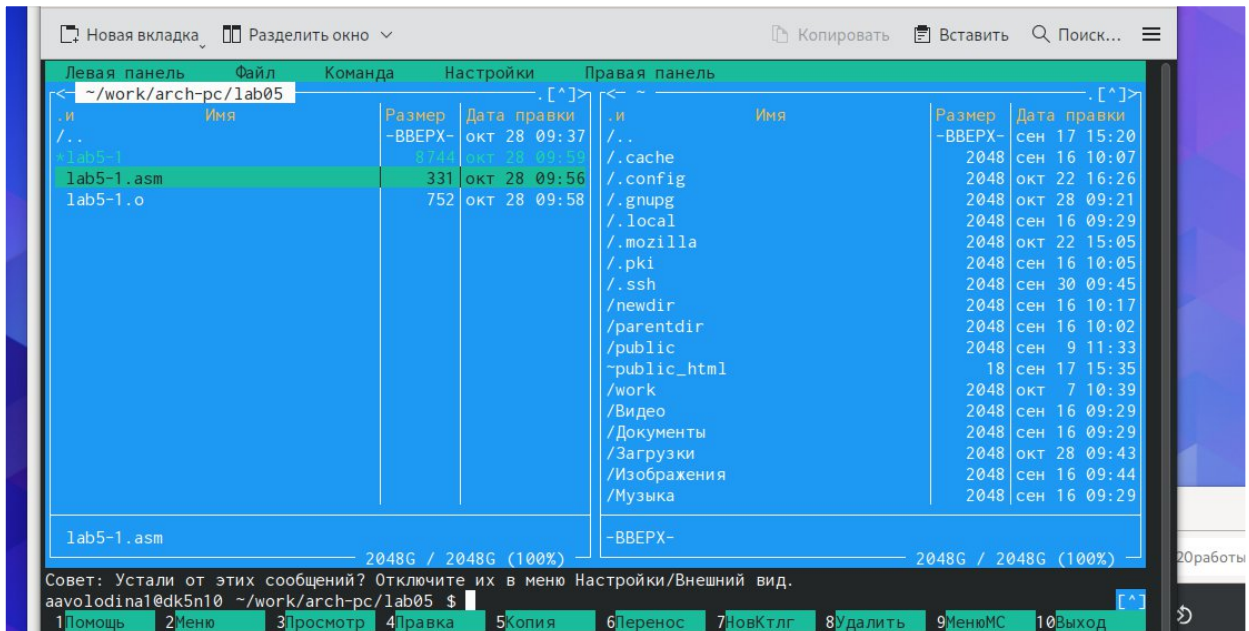


Рисунок 0.13

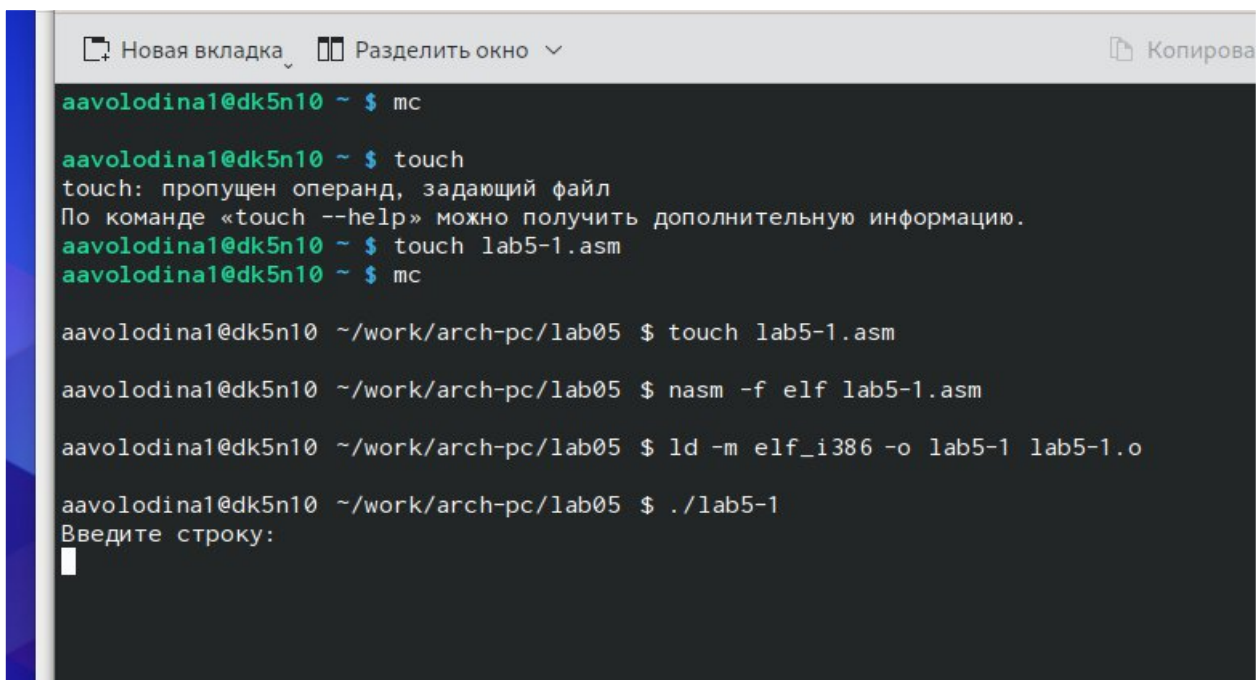
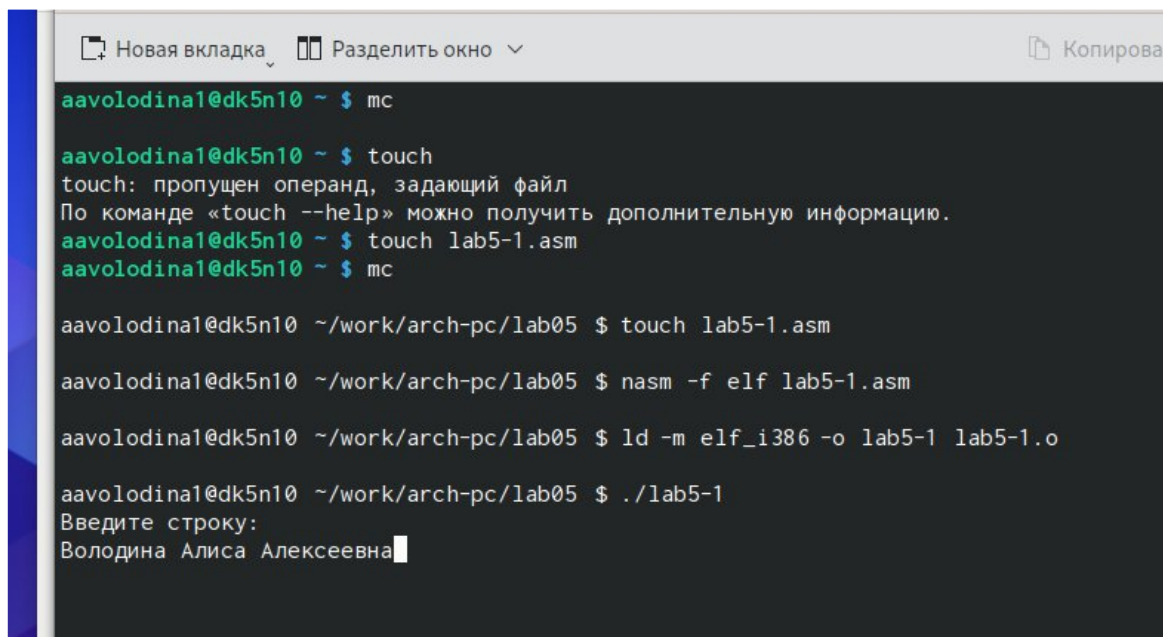


Рисунок 0.14



```
Новая вкладка Разделить окно Копировать
aavolodina1@dk5n10 ~ $ mc
aavolodina1@dk5n10 ~ $ touch
touch: пропущен операнд, задающий файл
По команде «touch --help» можно получить дополнительную информацию.
aavolodina1@dk5n10 ~ $ touch lab5-1.asm
aavolodina1@dk5n10 ~ $ mc
aavolodina1@dk5n10 ~/work/arch-pc/lab05 $ touch lab5-1.asm
aavolodina1@dk5n10 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.asm
aavolodina1@dk5n10 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o
aavolodina1@dk5n10 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку:
Володина Алиса Алексеевна
```

Рисунок 0.15

9. Скачаем файл in_out.asm со страницы курса в ТУИС

10. Подключаемый файл in_out.asm должен лежать в том же каталоге, что и файл с программой, в которой он используется.

11. С помощью функциональной клавиши f6 создадим копию файла lab5-1.asm с именем lab5-2.asm. Выделим файл lab5-1.asm, нажмем клавишу f6, введем имя файла lab5-2.asm и нажмем клавишу enter (рисунок 0.16-рисунок 0.17)

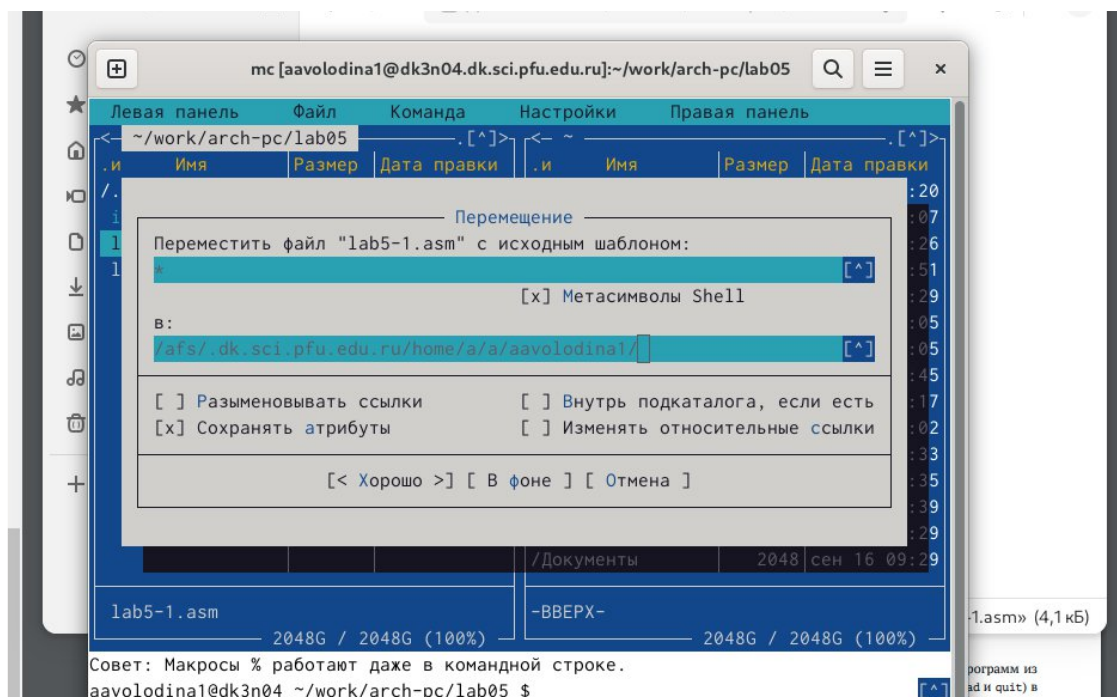


Рисунок 0.16

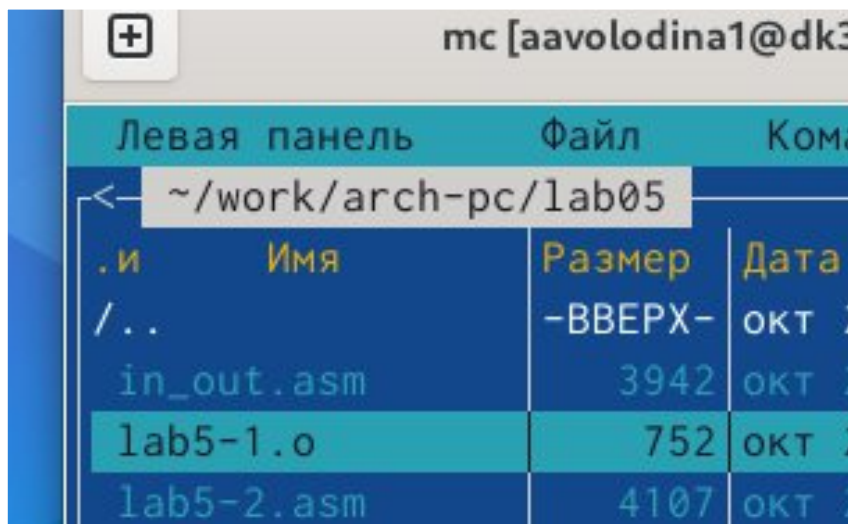


Рисунок 0.17

12. Исправим текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in_out.asm в соответствии с листингом. Создадим исполняемый файл и проверим его работу. (рисунок 0.18-0.19)

```

GNU nano 8.2 /afs/.dk.sci.pfu.edu.ru/home/a/
%include 'in_out.asm'

SECTION .data
msg: DB 'Введите строку: ', 0h
msgLen: EQU $-msg

SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:

    mov eax, msg
    call sprintLF

    mov ecx, buf1
    mov edx, 80

```

Рисунок 0.18

```

aavolodina1@dk5n06 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm

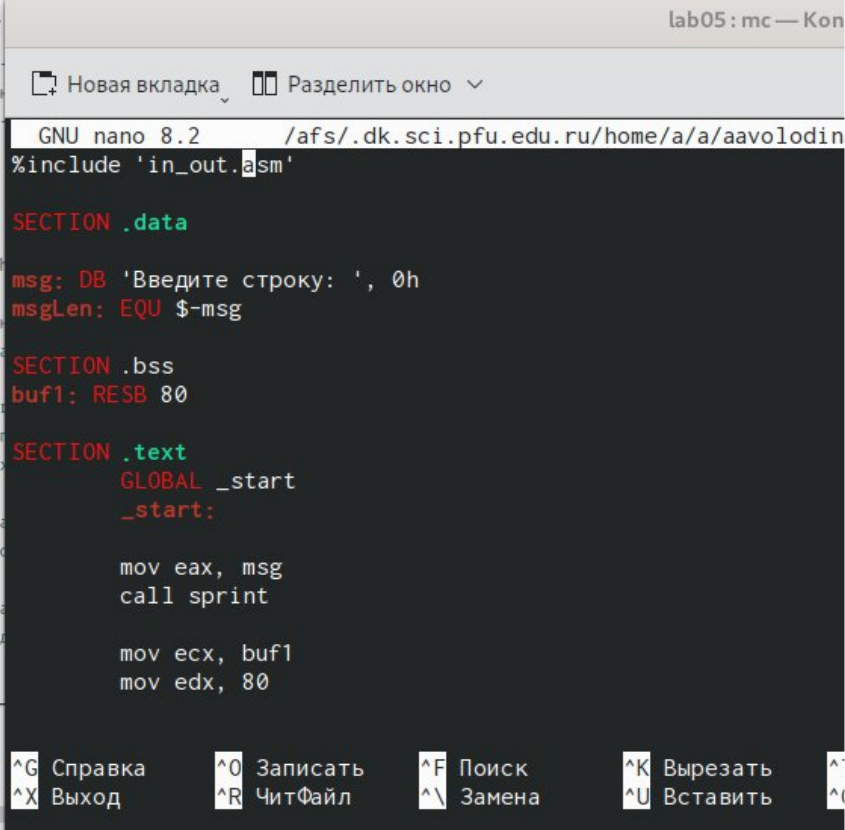
aavolodina1@dk5n06 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o

aavolodina1@dk5n06 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку:
Володина Алиса Алексеевна

```

Рисунок 0.19

13. В файле lab5-2.asm заменим подпрограмму sprintLF на sprint. Создадим исполняемый файл и проверим его работу. (рисунок 0.20-0.24)



```

GNU nano 8.2 /afs/.dk.sci.pfu.edu.ru/home/a/a/aavolodina1@dk5n06:~/work/arch-pc/lab05/lab5-2.asm
%include 'in_out.asm'

SECTION .data
msg: DB 'Введите строку: ', 0h
msgLen: EQU $-msg

SECTION .bss
buf1: RESB 80

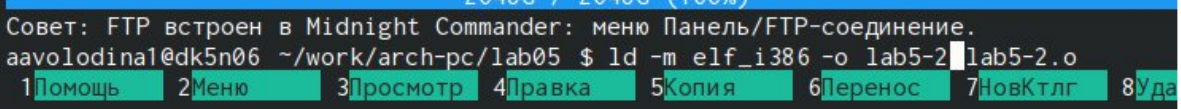
SECTION .text
GLOBAL _start
_start:

    mov eax, msg
    call sprint

    mov ecx, buf1
    mov edx, 80

```

Рисунок 0.20

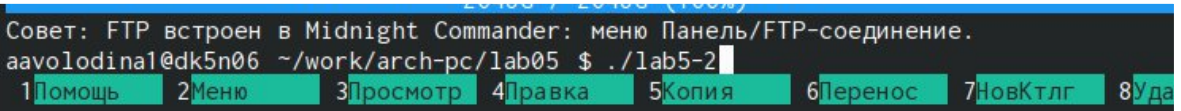


```

Совет: FTP встроен в Midnight Commander: меню Панель/FTP-соединение.
aavolodina1@dk5n06 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
1Помощь 2Меню 3Просмотр 4Правка 5Копия 6Перенос 7НовКтлг 8Уда

```

Рисунок 0.21



```

Совет: FTP встроен в Midnight Commander: меню Панель/FTP-соединение.
aavolodina1@dk5n06 ~/work/arch-pc/lab05 $ ./lab5-2
1Помощь 2Меню 3Просмотр 4Правка 5Копия 6Перенос 7НовКтлг 8Уда

```

Рисунок 0.22

```

aavolodina1@dk5n06 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
aavolodina1@dk5n06 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
aavolodina1@dk5n06 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку: 

```

Рисунок 0.23

```

aavolodina1@dk5n06 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
aavolodina1@dk5n06 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
aavolodina1@dk5n06 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку: Володина Алиса Алексеевна

```

Рисунок 0.24

Разница в `sprint` и `sprintLF` заключается в том, что `sprint` – обычный вывод, а `sprintLF` – вывод с переводом строки

Задание для самостоятельной работы

1 Создадим копию файла `lab5-1.asm`. Внесем изменения в программу, так чтобы она работала по следующему алгоритму:

- вывести приглашение типа “Введите строку:”;
- ввести строку с клавиатуры;
- вывести введенную строку на экран

(рисунок 0.25-0.30)

```

lab5-1.o
lab5-1copy.asm
*lab5-2
lab5-2.o
lab5-2.o

```

Рисунок 0.25


```

GNU nano 8.2 /afs/.dk.sci.pfu.edu.ru/home/a/a/aav
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

```

Рисунок 0.26

```

GNU nano 8.2 /afs/.dk.sci.pfu.edu.ru/home/a/a/aavolodina1/wor
_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, buf1
    int     80h
    mov     eax, 1
    mov     ebx, 0
    int     80h

```

Рисунок 0.27

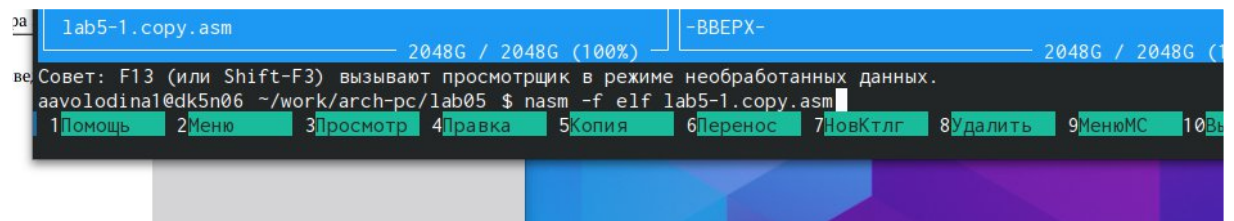


Рисунок 0.28

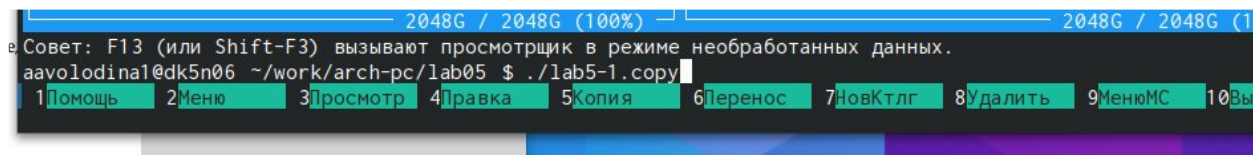


Рисунок 0.29

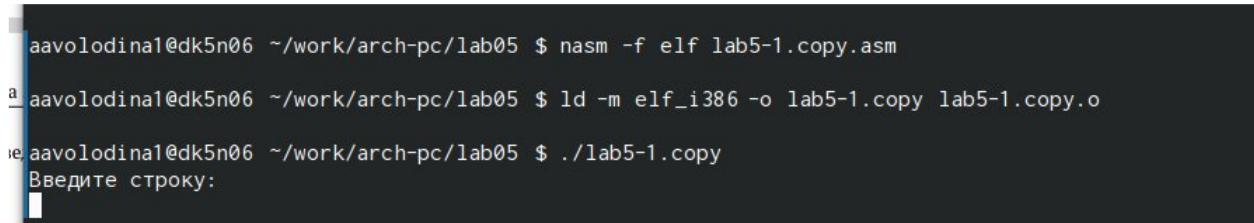


Рисунок 0.30

2. Получим исполняемый файл и проверим его работу. На приглашение ввести строку введем фамилию (рисунок 0.31-0.32)

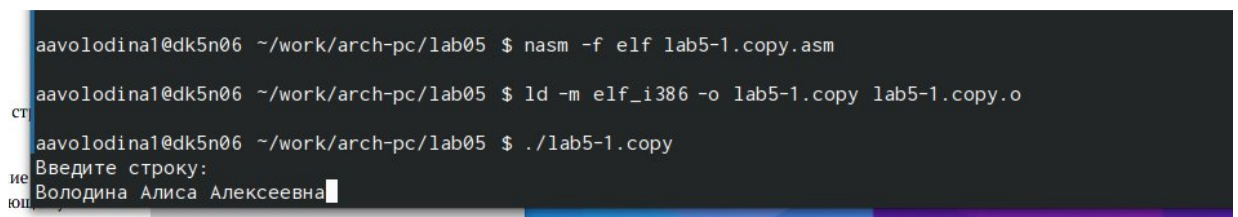


Рисунок 0.31

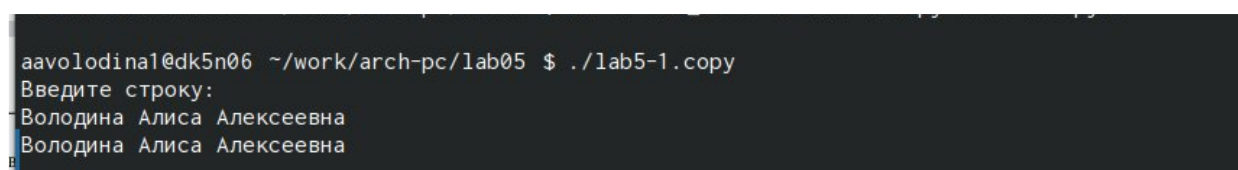


Рисунок 0.32

3. Создадим копию файла lab5-2.asm. Исправим текст программы с использованием подпрограмм из внешнего файла in_out.asm, так чтобы она работала по следующему алгоритму:

- вывести приглашение типа "Введите строку:";

- ввести строку с клавиатуры;

- вывести введённую строку на экран

(рисунок 0.33-0.36)

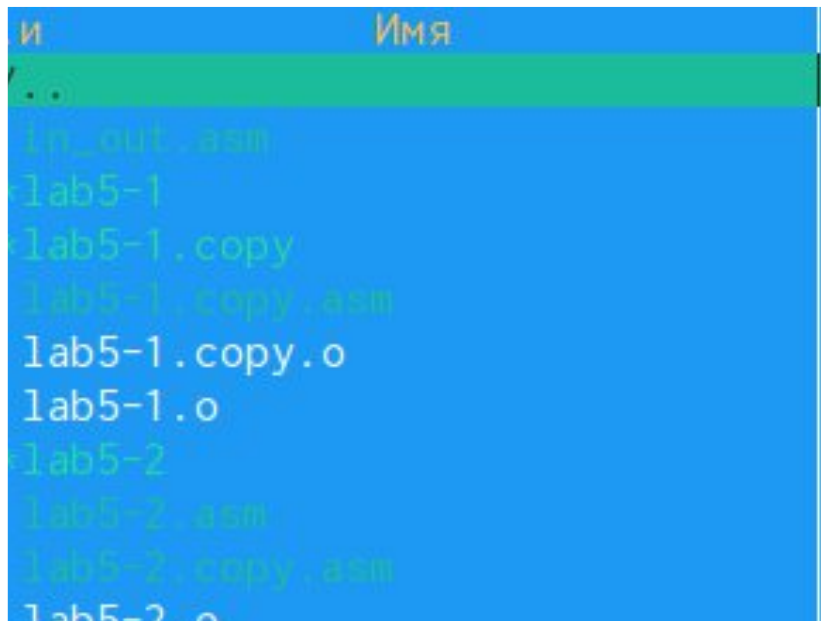


Рисунок 0.33

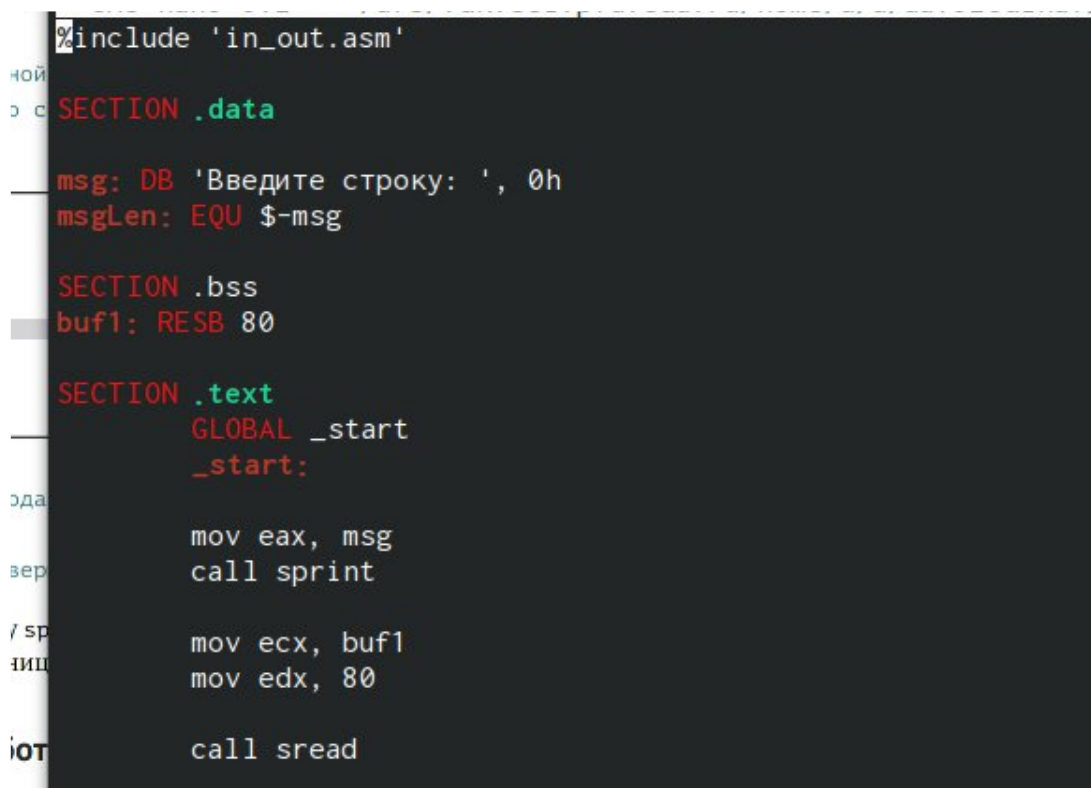


Рисунок 0.34

```

SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:

    mov eax, msg
    call sprint

    mov ecx, buf1
    mov edx, 80

    call sread

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

    call quit

```

Рисунок 0.35

```

lab5-1.o
*lab5-2
lab5-2.asm
lab5-2.copy.asm
lab5-2.copy.o

```

Рисунок 0.36

4. Создадим исполняемый файл и проверим его работу.(рисунок 0.37-0.40)

Совет: Устали от этих сообщений? Отключите их в меню Настройки/Внешний вид.

```

aavolodina1@dk5n06 ~/work/arch-pc/lab05 $ d -m elf_i386 -o lab5-2.copy lab5-2.copy.o

```

Рисунок 0.37

```

aavolodina1@dk5n06 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2.copy lab5-2.copy.o
aavolodina1@dk5n06 ~/work/arch-pc/lab05 $ ./lab5-2.copy
Введите строку:

```

Рисунок 0.38

```
aavolodina1@dk5n06 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2.copy lab5-2.copy.o
aavolodina1@dk5n06 ~/work/arch-pc/lab05 $ ./lab5-2.copy
Введите строку: Володина Алиса Алексеевна
```

Рисунок 0.39

```
aavolodina1@dk5n06 ~/work/arch-pc/lab05 $ ./lab5-2.copy
Введите строку: Володина Алиса Алексеевна
Володина Алиса Алексеевна
ид ааволодина1@dk5n06 ~ $
```

Рисунок 0.40

5)Вывод

В ходе выполнения данной лабораторной работы я приобрела практические навыки работы в Midnight Commander и 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/Learning bash-Shell-Programming-Nutshell/dp/0596009658](http://www.amazon.com/Learning-bash-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, 2017.
11. Новожилов О. П. Архитектура ЭВМ систем.—М.: Юрайт, 2016.
12. Расширенный ассемблер : NASM.—2021.—URL: <https://www.opennet.ru/docs/RUS/nasm/>.
13. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX.—2-е изд.—БХВ Петербург, 2010.—656 с.—ISBN 978-5-94157-538-1.
14. 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.
15. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
16. Таненбаум Э., Бос Х. Современные операционные системы.—4-е изд.—СПб.: Питер, 2015. —1120 с.—(Классика Computer Science)