

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

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

Содержание иллюстраций

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

<u>Рисунок 0.32</u>	18
<u>Рисунок 0.33</u>	19
<u>Рисунок 0.34</u>	19
<u>Рисунок 0.35</u>	20
<u>Рисунок 0.36</u>	20
<u>Рисунок 0.37</u>	20
<u>Рисунок 0.38</u>	20
<u>Рисунок 0.39</u>	21
<u>Рисунок 0.40</u>	21

1) Цель работы

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

2)Задания

Основы работы с mc, 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 инициализированных (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 инициализированных 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 (четверёхбайтное 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)

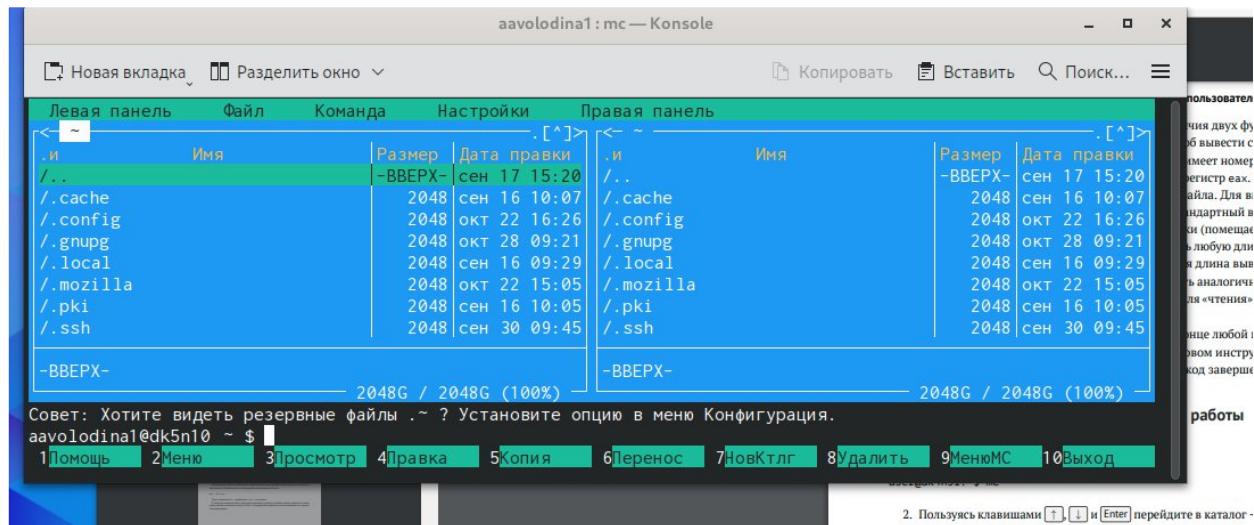


Рисунок 0.1

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

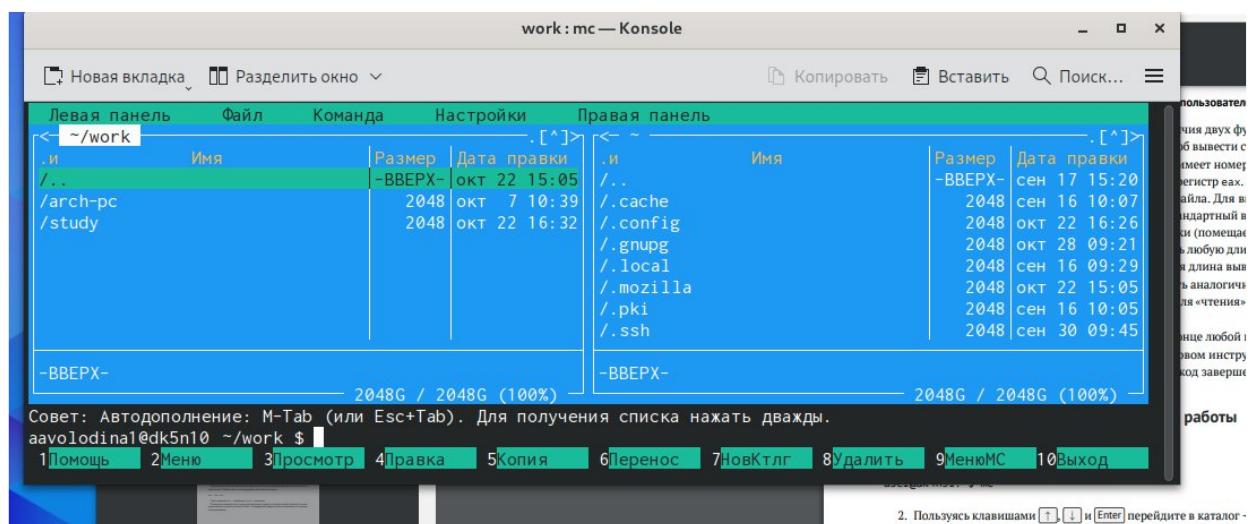


Рисунок 0.2

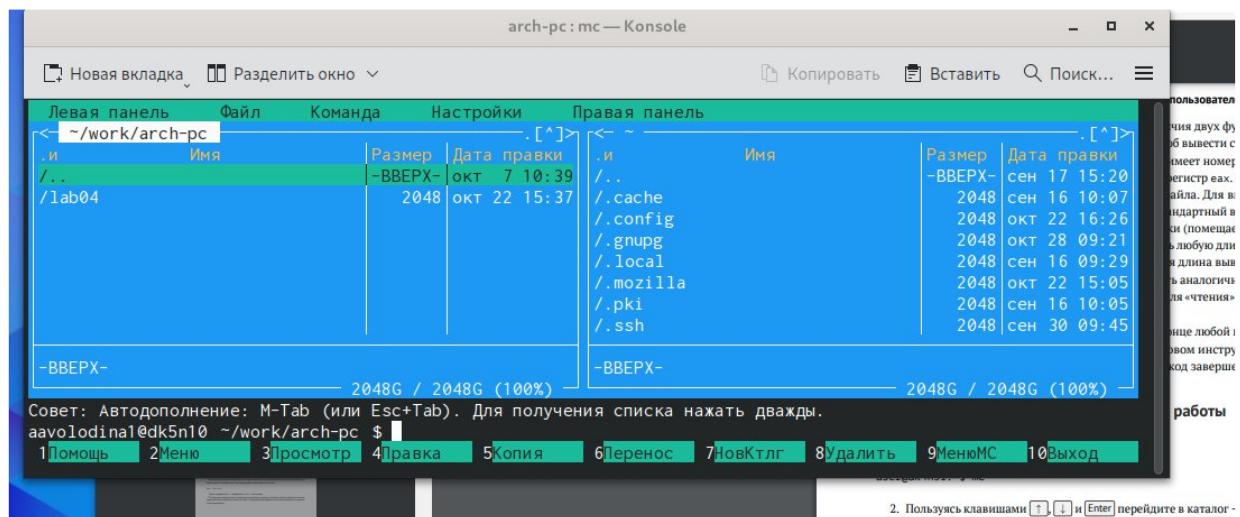


Рисунок 0.3

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

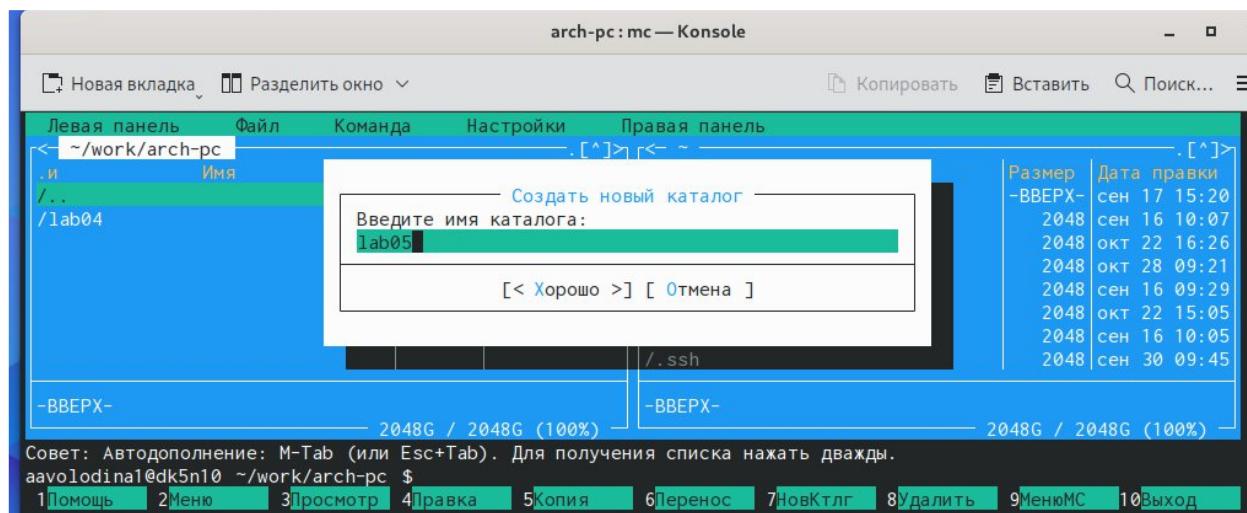


Рисунок 0.4

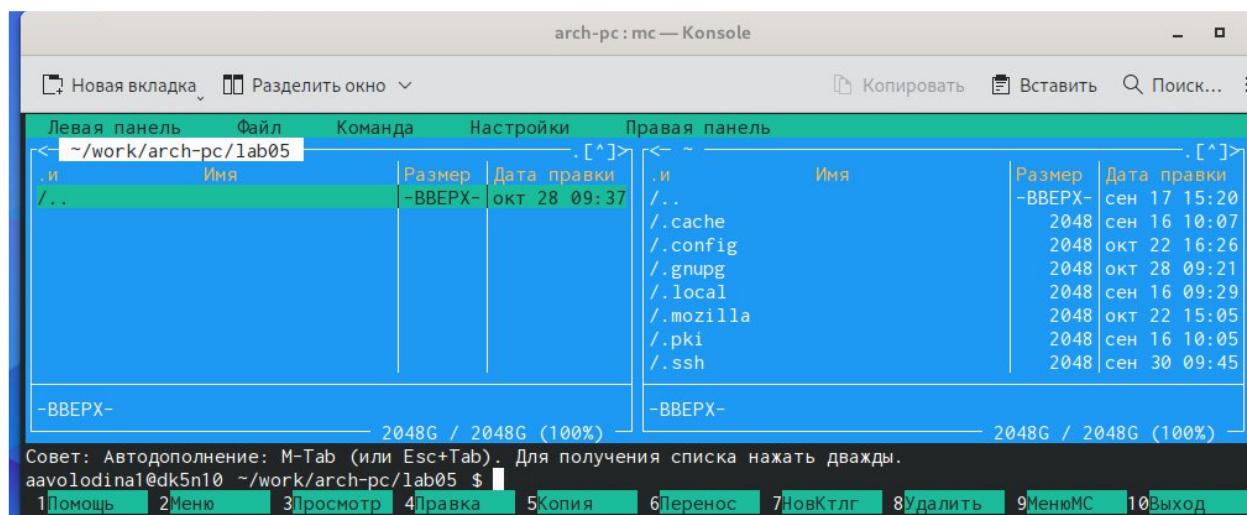


Рисунок 0.5

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 встроенным редакторе. (рисунок 0.7-0.9)

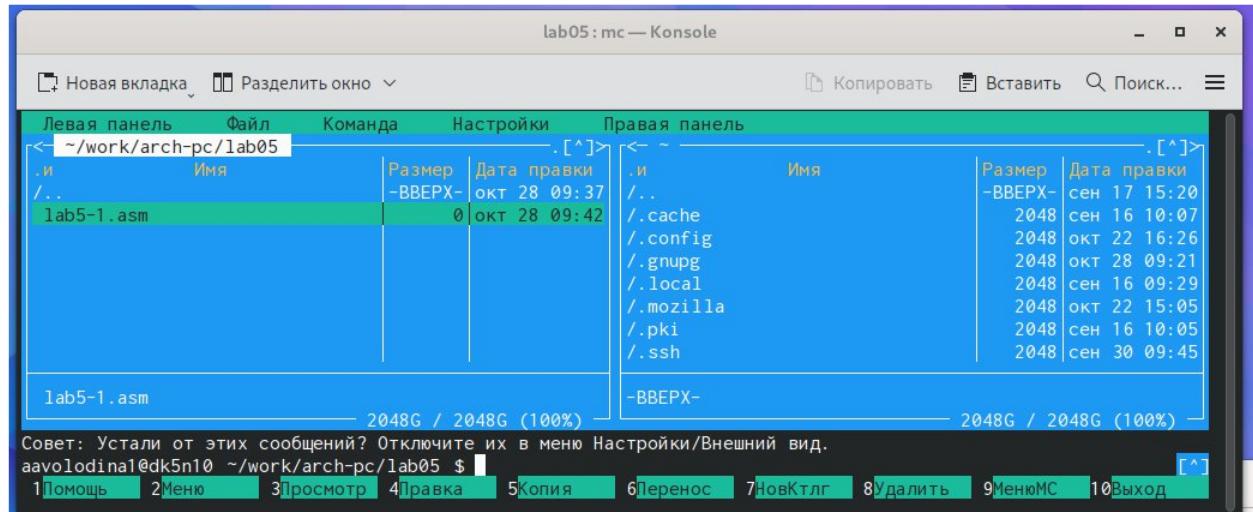


Рисунок 0.7



Рисунок 0.8

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

The screenshot shows a Windows Notepad window with the title bar 'Новая вкладка' and 'Разделить окно'. The menu bar includes 'Копировать', 'Вставить', 'Поиск...', and a file icon. The file path is 'lab5-1.asm' and the file size is 11 L: [1+22 23/ 23] *(331 / 331b) <EOF>. The content of the file is:

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

The screenshot shows a Windows Notepad window with the title bar 'Новая вкладка' and 'Разделить окно'. The menu bar includes 'Копировать', 'Вставить', 'Поиск...', and a file icon. The file path is 'lab5-1.asm' and the file size is 11 L: [1+22 23/ 23] *(331 / 331b) <EOF>. The content of the file is identical to the one in Figure 0.9.

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

```
Совет: Устали от этих сообщений? Отключите их в меню Настройки/Внешний вид.
aavolodina1@dk5n10 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.asm
1 Помощь 2 Меню 3 Просмотр 4 Правка 5 Копия 6 Перенос 7 НовКтлг 8 Удалить 9 МенюМС 10 Выход
25 mov
```

Рисунок 0.11

```
Совет: Устали от этих сообщений? Отключите их в меню Настройки/Внешний вид.
aavolodina1@dk5n10 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o
1 Помощь 2 Меню 3 Просмотр 4 Правка 5 Копия 6 Перенос 7 НовКтлг 8 Удалить 9 МенюМС 10 Выход
```

Рисунок 0.12

Левая панель	Файл	Команда	Настройки	Правая панель
~	Имя	Размер	Дата правки	~
.	-BBERX-	8744	окт 28 09:59	.
..				/.cache
lab5-1		331	окт 28 09:56	/.config
lab5-1.asm		752	окт 28 09:58	/.gnupg
lab5-1.o				/.local
				/.mozilla
				/.pki
				/.ssh
				/newdir
				/parentdir
				/public
				~public_html
				/work
				/Видео
				/Документы
				/Загрузки
				/Изображения
				/Музыка
				-BBERX-
				2048G / 2048G (100%)

Рисунок 0.13

```
aavolodina1@dk5n10 ~ $ mc
aavolodina1@dk5n10 ~ $ touch
touch: пропущен operand, 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:
```

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)

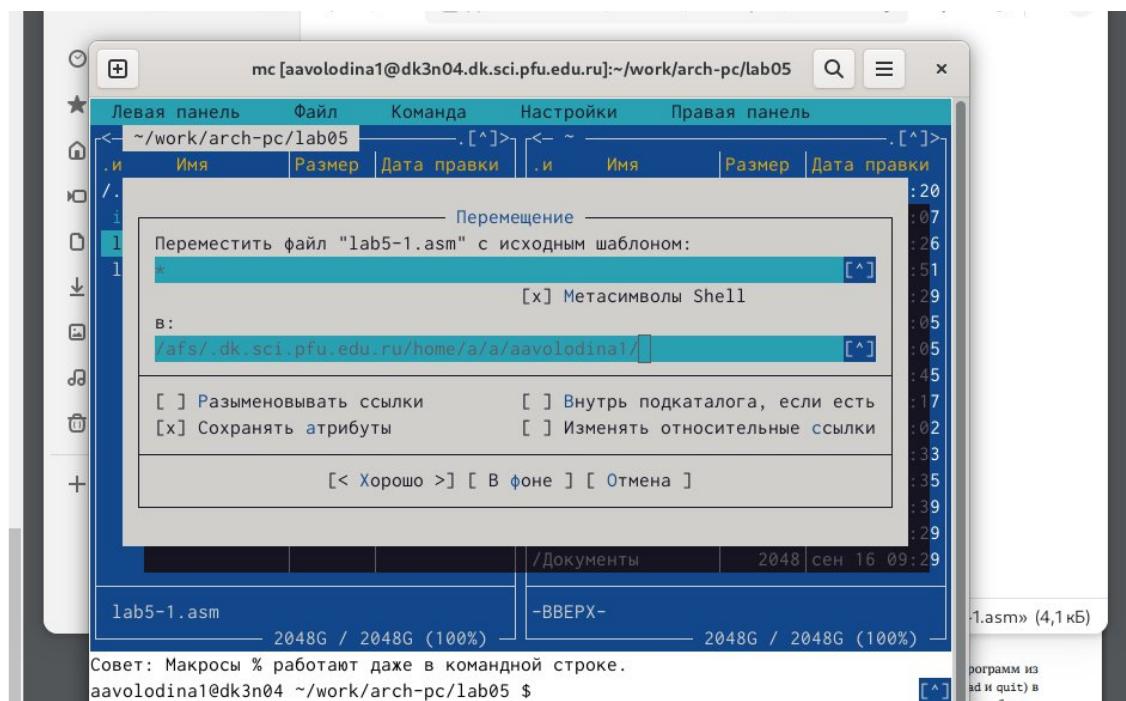


Рисунок 0.16

Левая панель	Файл	Команды	
.	Имя	Размер	Дата
<--	~/work/arch-pc/lab05	-ВВЕРХ-	окт 2
.	..		
	in_out.asm	3942	окт 2
	lab5-1.o	752	окт 2
	lab5-2.asm	4107	окт 2

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

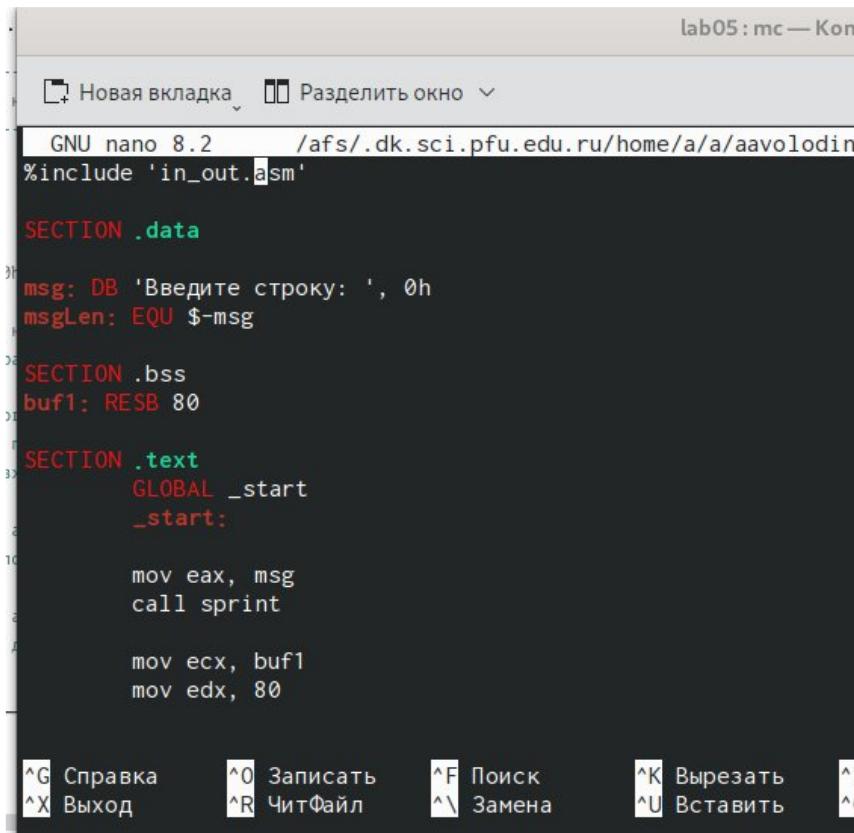
```

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/aavolodin
%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

^G Справка      ^O Записать      ^F Поиск      ^K Вырезать      ^
^X Выход        ^R ЧитФайл      ^\ Замена      ^U Вставить      ^C
```

Рисунок 0.20

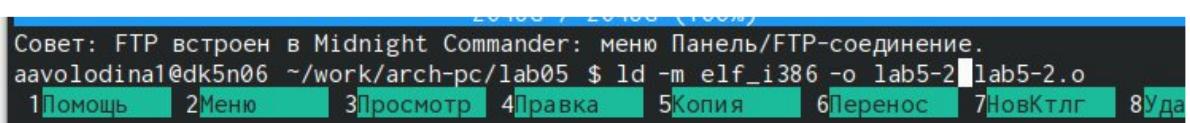


Рисунок 0.21

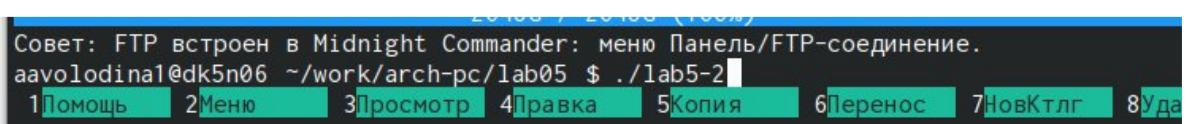


Рисунок 0.22

```
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 введённую строку на экран

(рисунок 0.25-0.30)

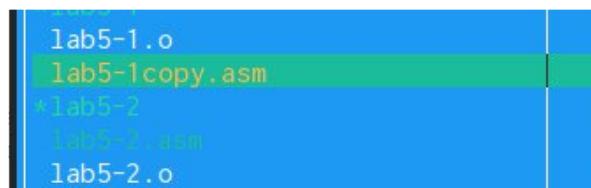


Рисунок 0.25

```
GNU nano 8.2      /afs/.dk.sci.pfu.edu.ru/home/a/a/aa\nSECTION .data\n\nmsg: DB 'Введите строку:',10\nmsgLen: EQU $-msg\n\nSECTION .bss\nbuf1: RESB 80\n\nSECTION .text\n\nGLOBAL _start\n\n_start:\n    mov     eax, 4\n    mov     ebx, 1\n    mov     ecx, msg\n    mov     edx, msgLen\n    int    80h\n    mov     eax, 3\n    mov     ebx, 0\n
```

Г. Правитов

Рисунок 0.26

```
GNU nano 8.2      /afs/.dk.sci.pfu.edu.ru/home/a/a/aavolodina1/wor\n_start:\n    mov     eax, 4\n    mov     ebx, 1\n    mov     ecx, msg\n    mov     edx, msgLen\n    int    80h\n    mov     eax, 3\n    mov     ebx, 0\n    mov     ecx, buf1\n    mov     edx, 80\n    int    80h\n    mov     eax, 4\n    mov     ebx, 1\n    mov     ecx, buf1\n    mov     edx, buf1\n    int    80h\n    mov     eax, 1\n    mov     ebx, 0\n    int    80h
```

Рисунок 0.27

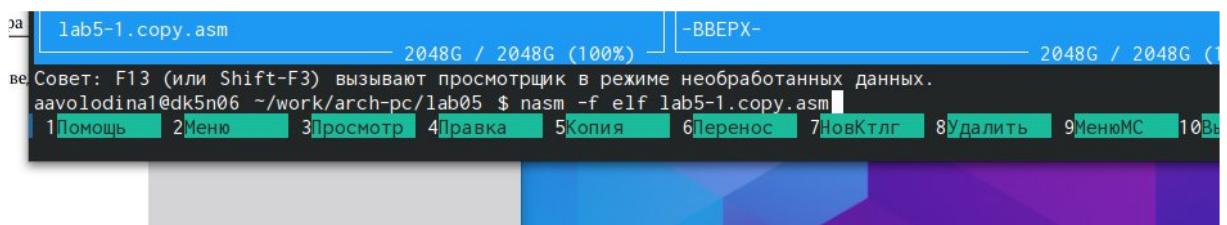


Рисунок 0.28

2048G / 2048G (100%) — 2048G / 2048G (1)
Совет: F13 (или Shift-F3) вызывают просмотрщик в режиме необработанных данных.
aavolodina1@dk5n06 ~/work/arch-pc/lab05 \$./lab5-1.copy
1Помощь 2Меню 3Просмотр 4Правка 5Копия 6Перенос 7НовКтлг 8Удалить 9МенюМС 10Вы

Рисунок 0.29

```
aavolodina1@dk5n06 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.copy.asm  
aavolodina1@dk5n06 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1.copy lab5-1.copy.o  
aavolodina1@dk5n06 ~/work/arch-pc/lab05 $ ./lab5-1.copy  
Введите строку:  
■
```

Рисунок 0.30

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

```
aavolodina1@dk5n06 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.copy.asm  
aavolodina1@dk5n06 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1.copy lab5-1.copy.o  
aavolodina1@dk5n06 ~/work/arch-pc/lab05 $ ./lab5-1.copy  
Введите строку:  
юш  
Володина Алиса Алексеевна■
```

Рисунок 0.31

```
aavolodina1@dk5n06 ~/work/arch-pc/lab05 $ ./lab5-1.copy  
Введите строку:  
- Володина Алиса Алексеевна  
Володина Алиса Алексеевна■
```

Рисунок 0.32

3. Создадим копию файла lab5-2.asm. Исправим текст программы с использование под программ из внешнего файла in_out.asm, так чтобы она работала по следующему алгоритму: • вывести приглашениетипа“Введите строку:”;

- ввести строку с клавиатуры;

- вывести введённую строку на экран

(рисунок 0.33-0.36)

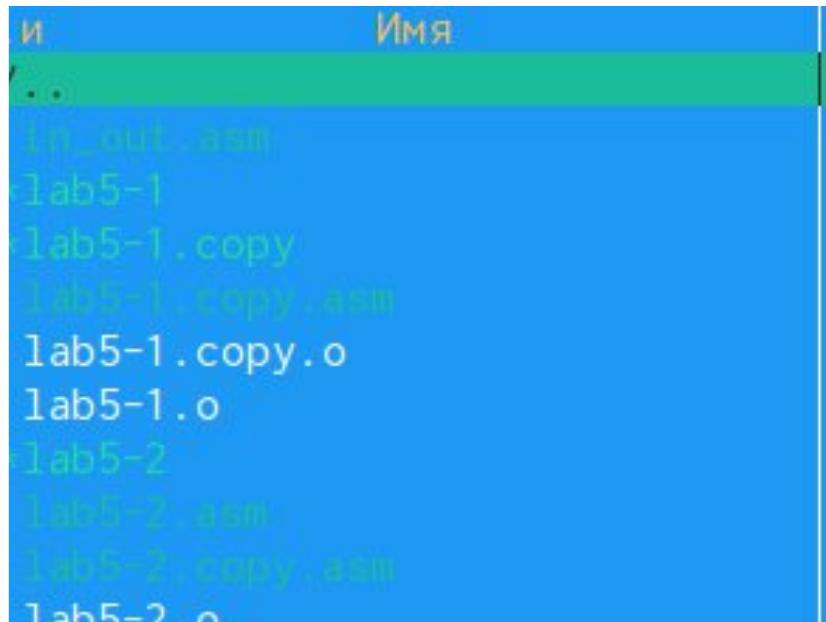


Рисунок 0.33

```
%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

    call sread
```

Рисунок 0.34

```
SECTION .bss
най buf1: RESB 80
о с

SECTION .text
    GLOBAL _start
    _start:

        mov eax, msg
        call sprint

        mov ecx, buf1
        mov edx, 80

    ода
        call sread

    вер
    / сп
    иц
        mov eax, 4
        mov ebx, 1
        mov ecx, buf1
        int 80h

    ют
        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 $ ld -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  
Ведите строку: Володина Алиса Алексеевна
```

Рисунок 0.39

```
aavolodina1@dk5n06 ~/work/arch-pc/lab05 $ ./lab5-2.copy  
Ведите строку: Володина Алиса Алексеевна  
Володина Алиса Алексеевна  
aavolodina1@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:TheGNUProjectDebugger.—URL:<https://www.gnu.org/software/gdb/>.
2. GNUBashManual.—2016.—URL:<https://www.gnu.org/software/bash/manual/>.
3. Midnight CommanderDevelopment Center.—2021.—URL: <https://midnight-commander.org/>.
4. NASMASsemblyLanguageTutorials.—2021.—URL: <https://asmtutor.com/>.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 c. —(In a Nutshell). —ISBN 0596009658.—URL: <http://www.amazon.com/Learning-bash-Shell-Programming-Nutshell/dp/0596009658>.
6. RobbinsA. Bash Pocket Reference.—O'Reilly Media,2016.—156 c.—ISBN 978-1491941591.
7. TheNASMdocumentation.—2021.—URL:<https://www.nasm.us/docs.php>.
8. Zarrelli G. Mastering Bash.—Packt Publishing,2017.—502 c.—ISBN 9781784396879.
9. Колдаев В. Д.,Лупин С. А.Архитектура ЭВМ.—М. : Форум,2018.
10. Куляс О. Л.,Никитин К. А. Курс программирования наASSEMBLER.—М. : Солон-Пресс, 2017.
11. НовожиловО.П.Архитектура ЭВМисистем.—М.:Юрайт,2016.
- 12 . Р а с ш и р е н и й а с с е м б л е р : N A S M . — 2 0 2 1 . — 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 c.—ISBN 978-5-94157-538-1.
14. СтоляровА.Программированиенаязыкеассемблера NASMдляOCUnix.—2-еизд.— М. : МАКСПресс,2011.—URL:http://www.stolyarov.info/books/asm_unix.
15. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 c. — (Классика Computer Science).
16. Таненбаум Э.,Бос Х. Современные операционные системы.—4-е изд.— СПб.: Питер, 2015. —1120 c.—(Классика Computer Science)