

# Отчёт по лабораторной работе **5**

Структура программы на языке ассемблера **NASM**. Системные  
вызовы в ОС **GNU Linux**

Ларичкин Иван

# Содержание

Цель работы	<b>5</b>
Задание	<b>6</b>
Теоретическое введение	<b>7</b>
Выполнение лабораторной работы	<b>8</b>
Выводы	<b>19</b>

## Список иллюстраций

# Список таблиц

1	Описание некоторых каталогов файловой системы GNU Linux {#tbl:std-dir} .....	7
1.	Цель работы	
2.	Задание	
3.	Теоретическое введение	
4.	Выполнение лабораторной работы	
5.	Вывод	

# Цель работы

Изучить структуру программы на языке ассемблера NASM

# Задание

1. Открыть Midnight Commander
2. Создать папку lab05 и внутри нее создать файл lab5-1.asm
3. Открыть файл lab5-1.asm, ввести информацию из листинга 5.1 и сохранить изменения
4. Убедиться что файл содержит информацию
5. Оттранслировать текст файла lab5-1.asm, выполнить компоновку объектного файла
6. Запустить файл
7. Скачать и скопировать файл in\_out.asm с помощью клавиши f5
8. С помощью клавиши f6 скопировать файл lab5-1.asm с именем lab5-2.asm
9. Исправить файл lab5-2.asm в соответствии с листингом 5.2
10. В файле lab5-2.asm заменить подпрограмму sprintLF на sprint
11. Создать исполняемый файл и проверить его работу
12. Создать копию файла lab5-1.asm и внести изменения, чтобы выводилась введенная строка на экран
13. Создать копию файла lab5-2.asm и внести изменения, чтобы выводилась введенная строка на экран

# Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы.

Например, в табл. [-@tbl:std-dir] приведено краткое описание стандартных каталогов Unix.

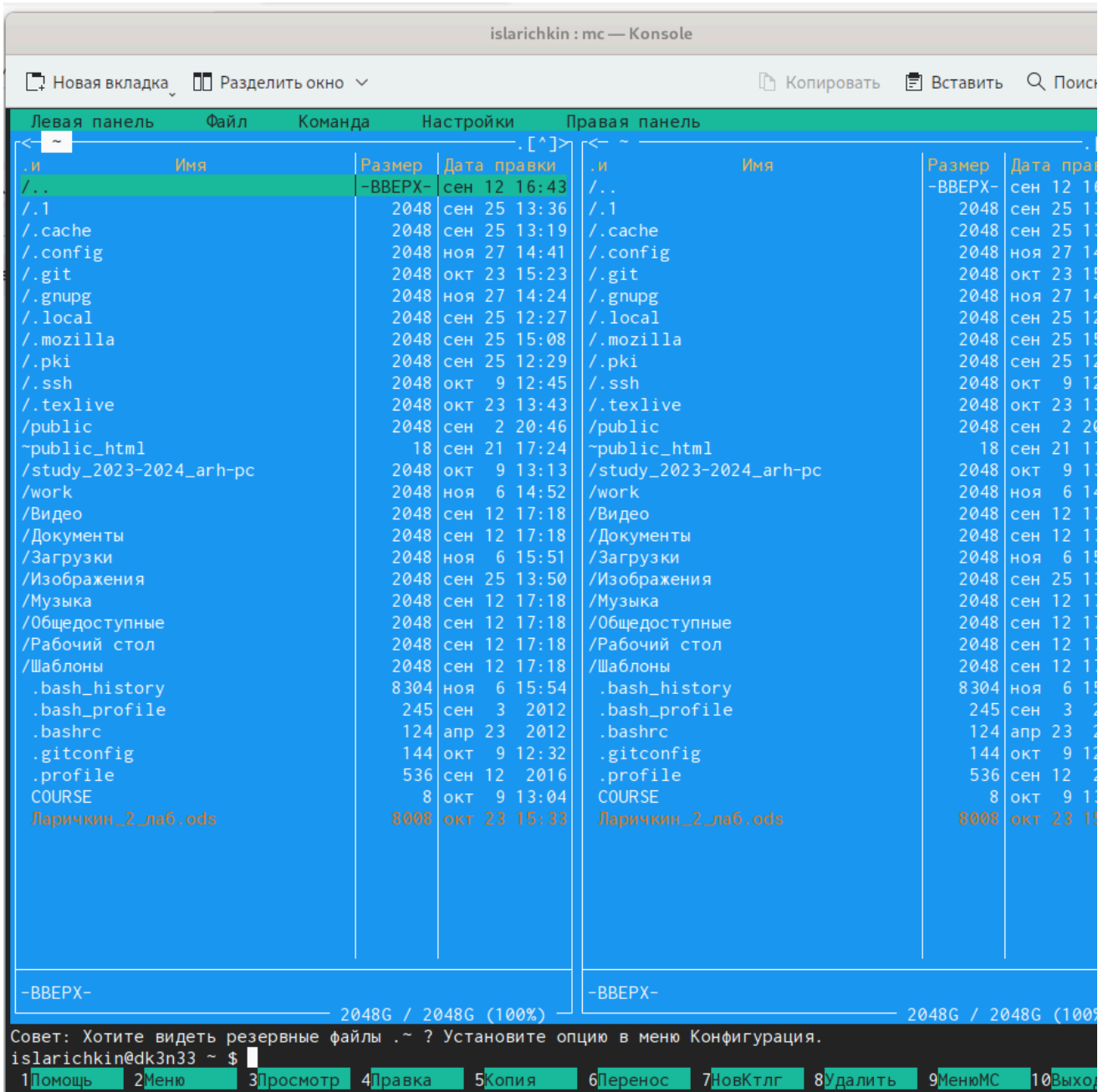
Таблица 1: Описание некоторых каталогов файловой системы GNU Linux {#tbl:std-dir}

Имя каталога	Описание каталога
/	Корневая директория, содержащая всю файловую систему
/bin	Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям
/etc	Общесистемные конфигурационные файлы и файлы конфигурации установленных программ
/home	Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя
/media	Точки монтирования для сменных носителей
/root	Домашняя директория пользователя root
/tmp	Временные файлы
/usr	Вторичная иерархия для данных пользователя

Более подробно про Unix см. в [@tanenbaum\_book\_modern-os\_ru; @robbins\_book\_bash\_en; @zarrelli\_book\_mastering-bash\_en; @newham\_book\_learning-bash\_en].

# Выполнение лабораторной работы

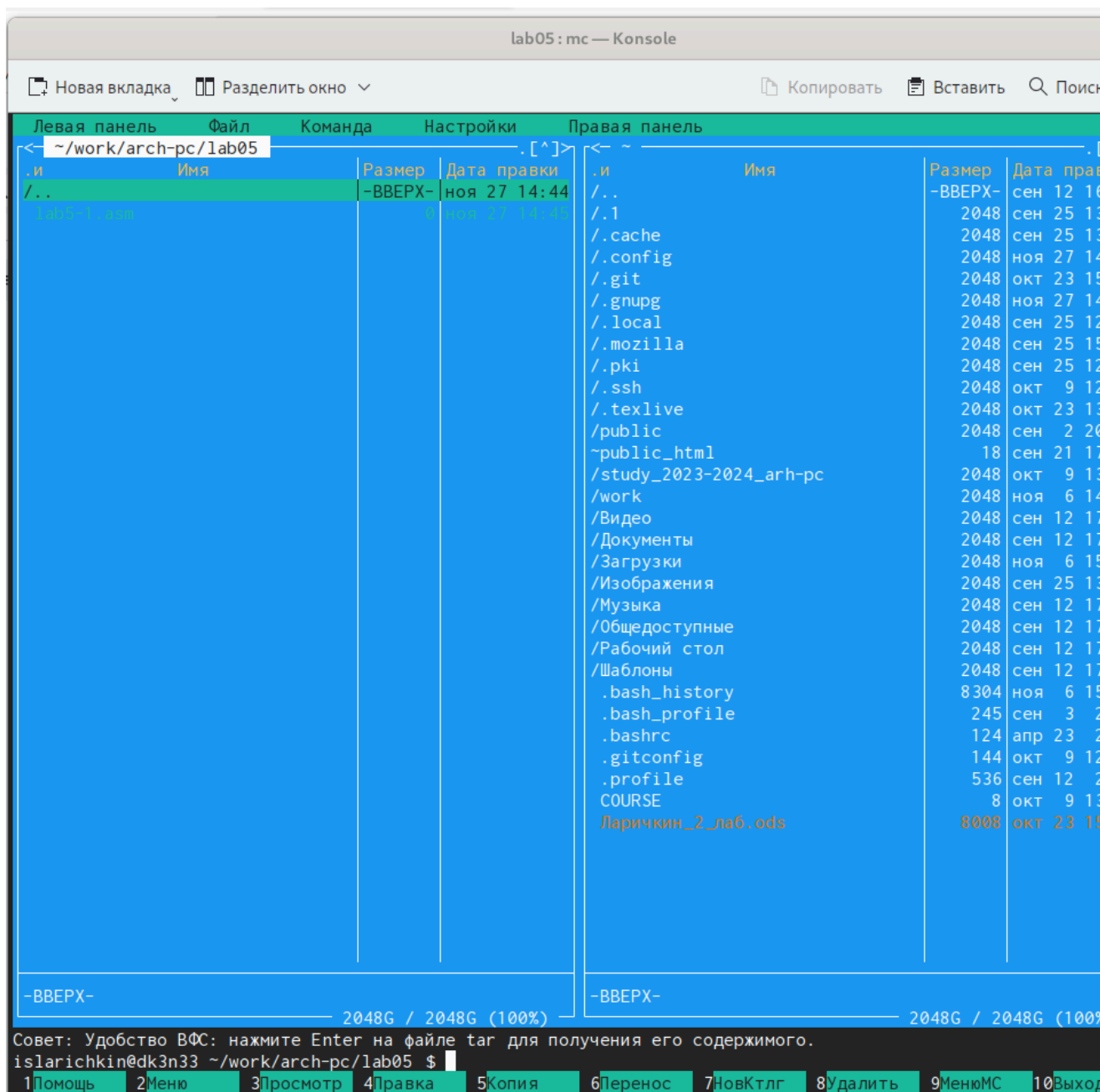
1. Открыть Midnight Commander (см рис 1)



Открытый MC (рис 1)



2. Создать папку lab05 и внутри нее создать файл lab5-1.asm (см рис 2)



Создание папки lab05 и файла lab5-1.asm (рис 2)

- Открыть файл lab5-1.asm, ввести информацию из листинга 5.1 и сохранить изменения (см рис 3)

```
lab5-1.asm [----] 20 L: [ 1+34 35/ 35] *(2429/2429b) <EOF>

; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
;----- Объявление переменных -----
SECTION .data ; Секция инициализированных данных
msg: DB "Введите строку: ",10 ; сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не инициализированных данных
buf1: RESB 80 ; Буфер размером 80 байт
;----- Текст программы -----
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
;----- Системный вызов 'write' -----
; После вызова инструкции 'int 80h' на экран будет
; выведено сообщение из переменной 'msg' длиной 'msgLen'
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
;----- системный вызов 'read' -----
; После вызова инструкции 'int 80h' программа будет ожидать ввода
; строки, которая будет записана в переменную 'buf1' размером 80 байт
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
;----- Системный вызов 'exit' -----
; После вызова инструкции 'int 80h' программа завершит работу
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра
```

Открытый файл lab5-1.asm (рис 3)

4. Убедится что файл содержит информацию (см рис 3)
5. Оттранслировать текст файла lab5-1.asm, выполнить компоновку объектного файла (см рис 4)

```
islarichkin@dk3n33 - lab05
islarichkin@dk3n33 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.asm
islarichkin@dk3n33 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o
islarichkin@dk3n33 ~/work/arch-pc/lab05 $ ./lab5-1
bash: ./lab5-1: Нет такого файла или каталога
islarichkin@dk3n33 ~/work/arch-pc/lab05 $ ./lav5-1
bash: ./lav5-1: Нет такого файла или каталога
islarichkin@dk3n33 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку:
□
```

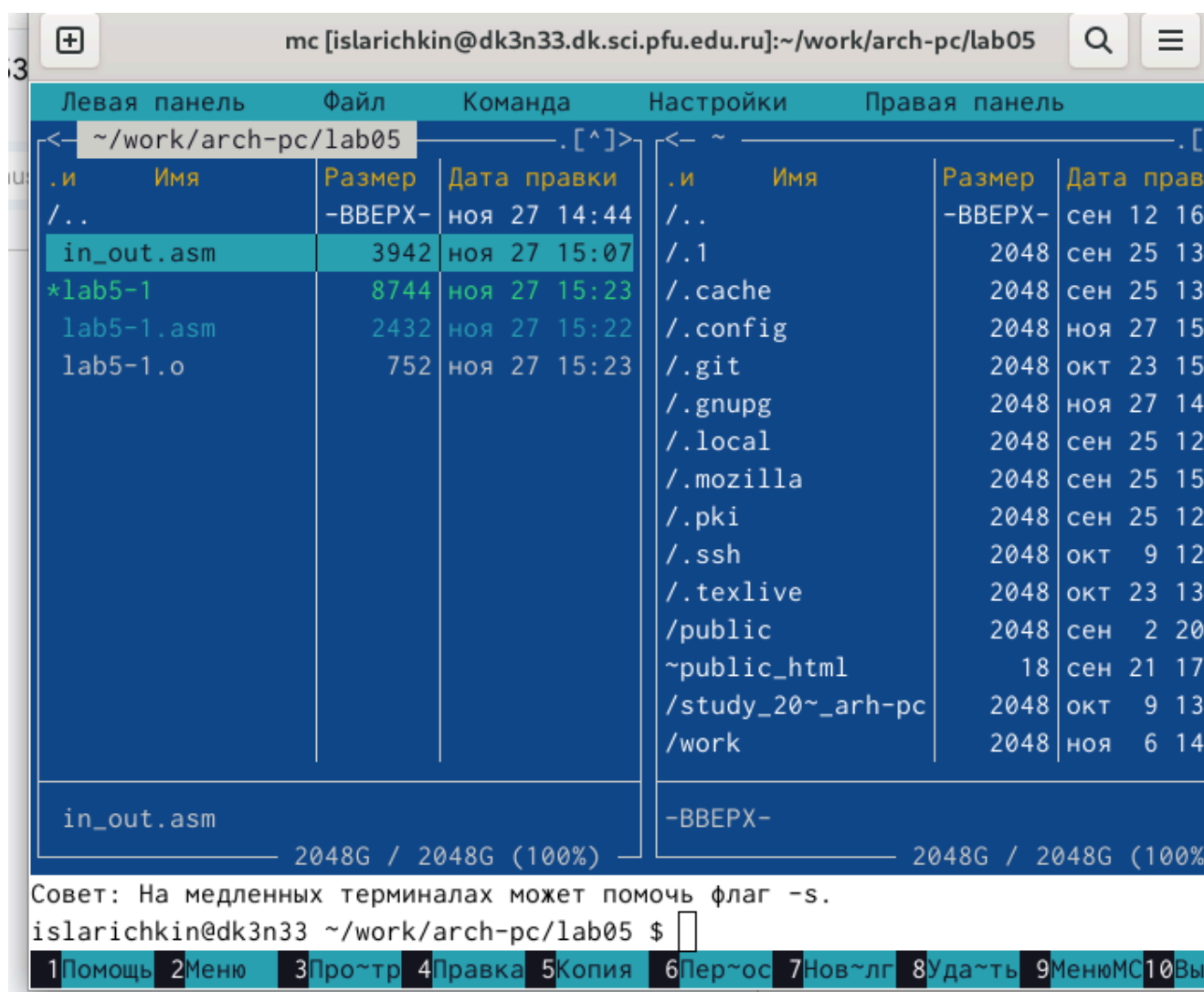
Выполнение команд (рис 4)

6. Запустить файл (см рис 5)

```
islarichkin@dk3n33 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку:
Ларичкин Иван Сергеевич
islarichkin@dk3n33 ~/work/arch-pc/lab05 $
```

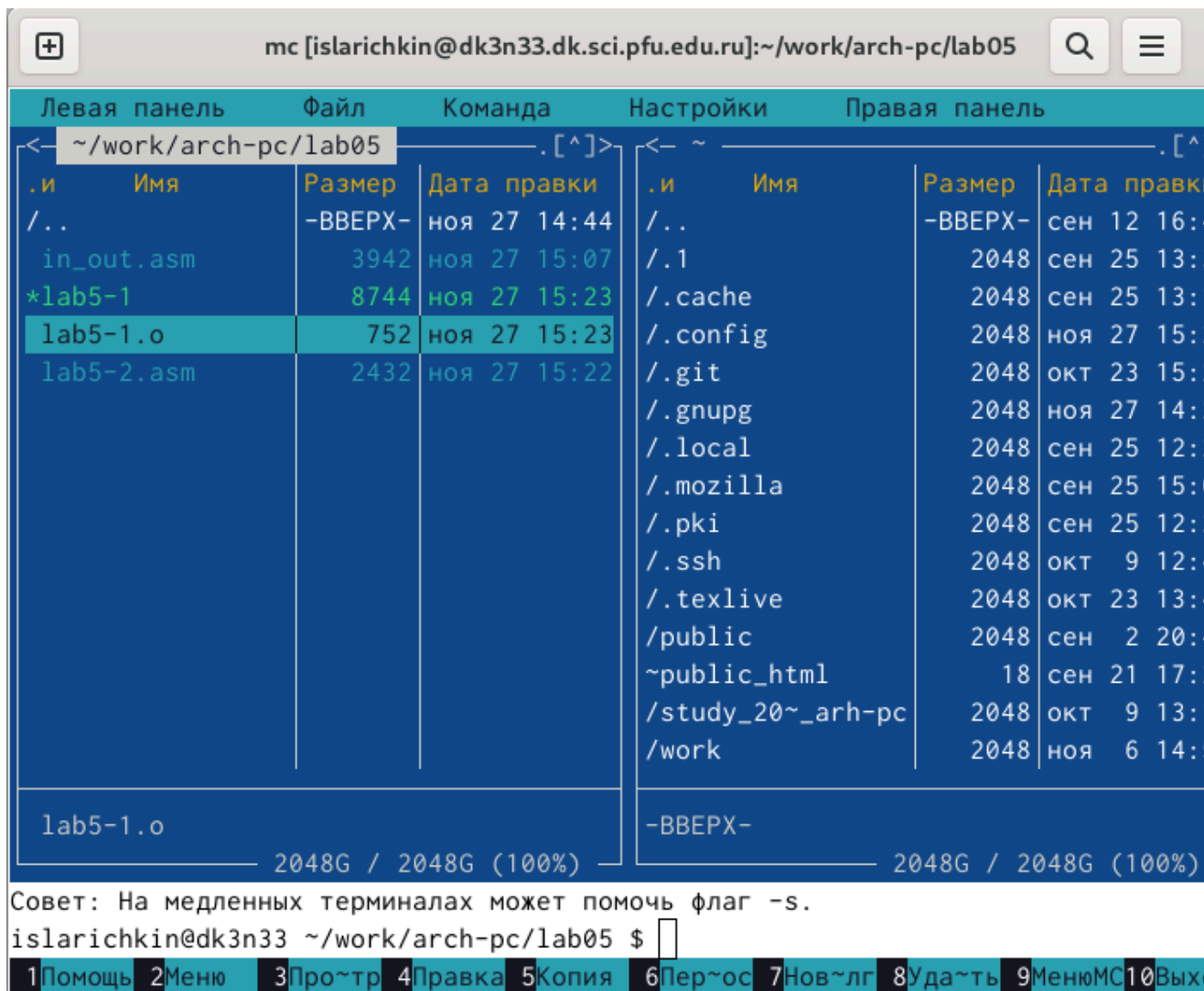
Запуск файла (рис 5)

7. Скачать и скопировать файл in\_out.asm с помощью клавиши f5 (см рис 6)



Скопированный in\_out.asm через f5 (рис 6)

8. С помощью клавиши f6 скопировать файл lab5-1.asm с именем lab5-2.asm (см рис 7)



Скопированный файл lab5-1.asm с именем lab5-2.asm (рис 7)

- Исправить файл lab5-2.asm в соответствии с листингом 5.2 и заменить подпрограмму sprintLF на sprint (см рис 8)

+

islarichkin@dk3n33 - islarichkin

Q

≡

/afs/.dk.sci.pfu.edu.ru/home/i/s/islarichkin/work/arch-pc/lab05/lab5-2.a

;

-----

; Программа вывода сообщения на экран и ввода строки с клавиатуры

-----

;

%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

[ Прочитано 17 строк ]

^G Справка

^O Записать

^F Поиск

^K Вырезать

^T Выполнить

M-U Отмена

^X Выход

^R ЧитФайл

^\\ Замена

^U Вставить

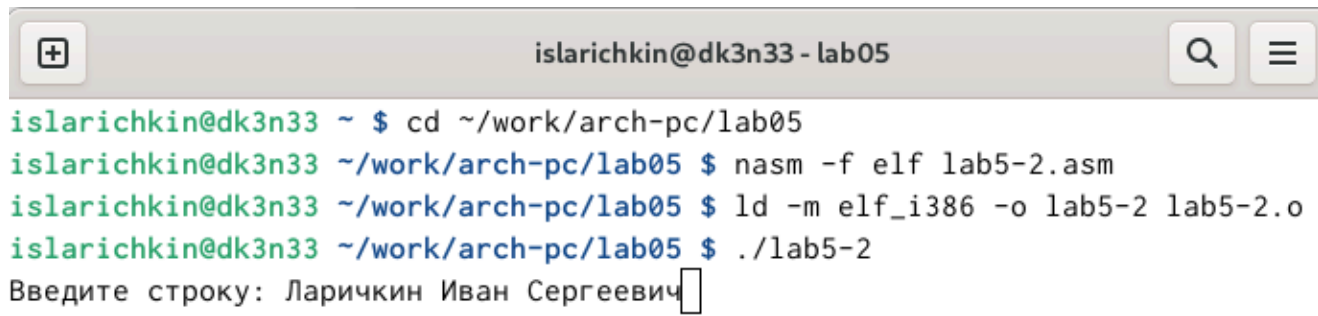
^C Позиция

M-E Повтор

Исправленный файл lab5-2.asm (рис 8)

10. Создать исполняемый файл и проверить его работу (см рис 9)





A terminal window titled "islarichkin@dk3n33 - lab05" with a search icon and a menu icon in the top right corner. The terminal shows the following commands and output:

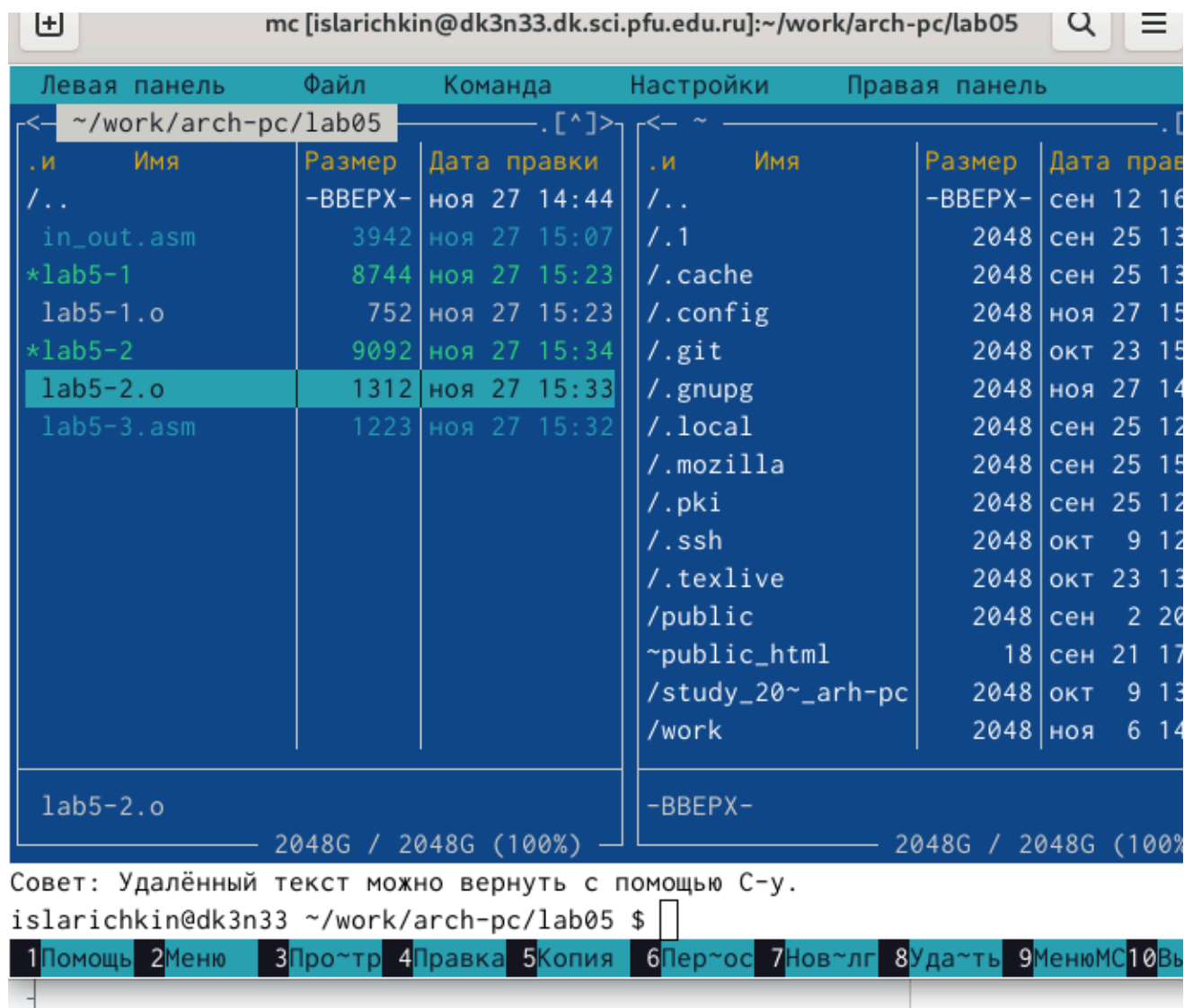
```
islarichkin@dk3n33 ~ $ cd ~/work/arch-pc/lab05
islarichkin@dk3n33 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
islarichkin@dk3n33 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
islarichkin@dk3n33 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку: Ларичкин Иван Сергеевич
```

---

Проверка и создание файла (рис 10)

Теперь после вывода сообщения не будет перехода на новую строку

11. Создать копию файла lab5-1.asm и внести изменения, чтобы выводила вве-



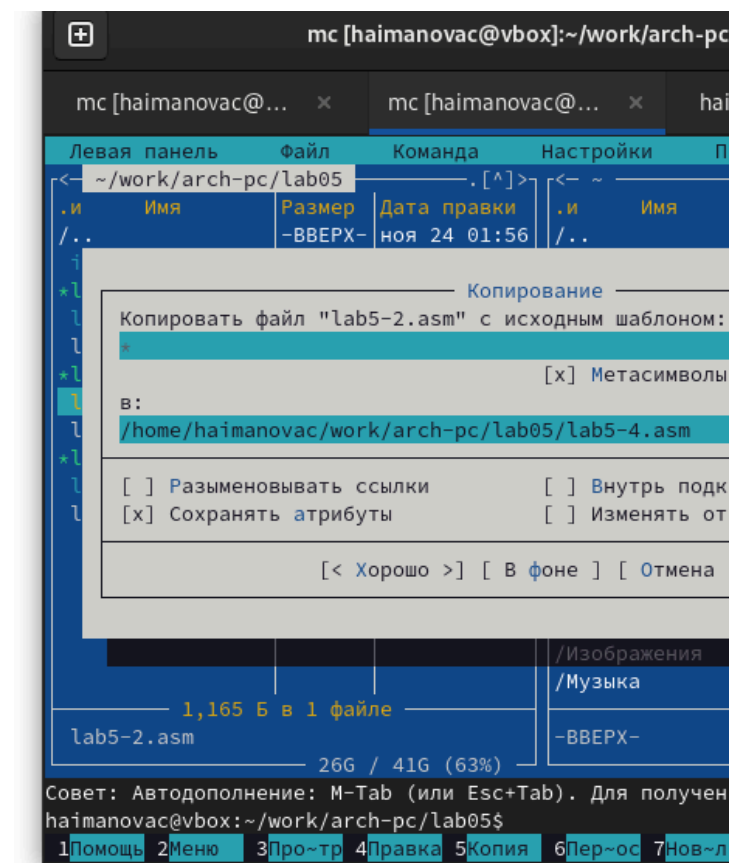
денная строка на экран (см рис 10-11)

Создание копии файла lab5-1.asm (рис 10)

```
islarichkin@dk3n33 ~/work/arch-pc/lab05 $ nasm -f elf lab5-3.asm
islarichkin@dk3n33 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-3 lab5-3.o
islarichkin@dk3n33 ~/work/arch-pc/lab05 $ ./lab5-3
Введите строку: Ларичкин
Ларичкин
islarichkin@dk3n33 ~/work/arch-pc/lab05 $ 
```

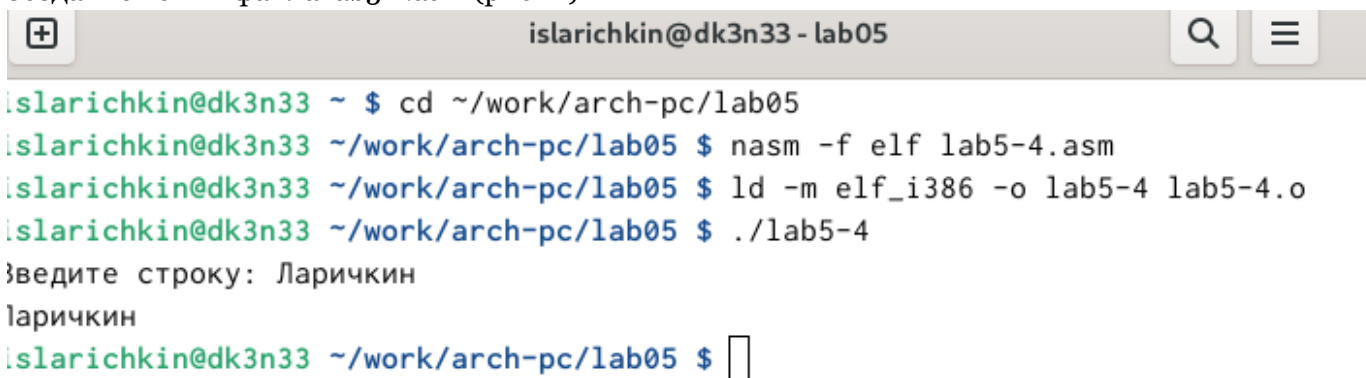
Вывод введенной строки на экран (рис 11)

12. Создать копию файла lab5-2.asm и внести изменения, чтобы выводила вве-



денная строка на экран (см рис 12-13)

Создание копии файла lab5-1.asm (рис 12)



Вывод введенной строки на экран (рис 13)

# Выводы

В процессе выполнения лабораторной работы я ознакомился со структурой программы на языке ассемблера NASM

