

Отчёт по лабораторной работе 5

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

Хайманов Асланбек Султанович

Содержание

Цель работы	5
Задание	6
Теоретическое введение	7
Выполнение лабораторной работы	8
Выводы	19
Список литературы	20

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 выводила введенная строка на экран
13. Создать копию файла lab5-2.asm и внести изменения, чтобы выводила 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.

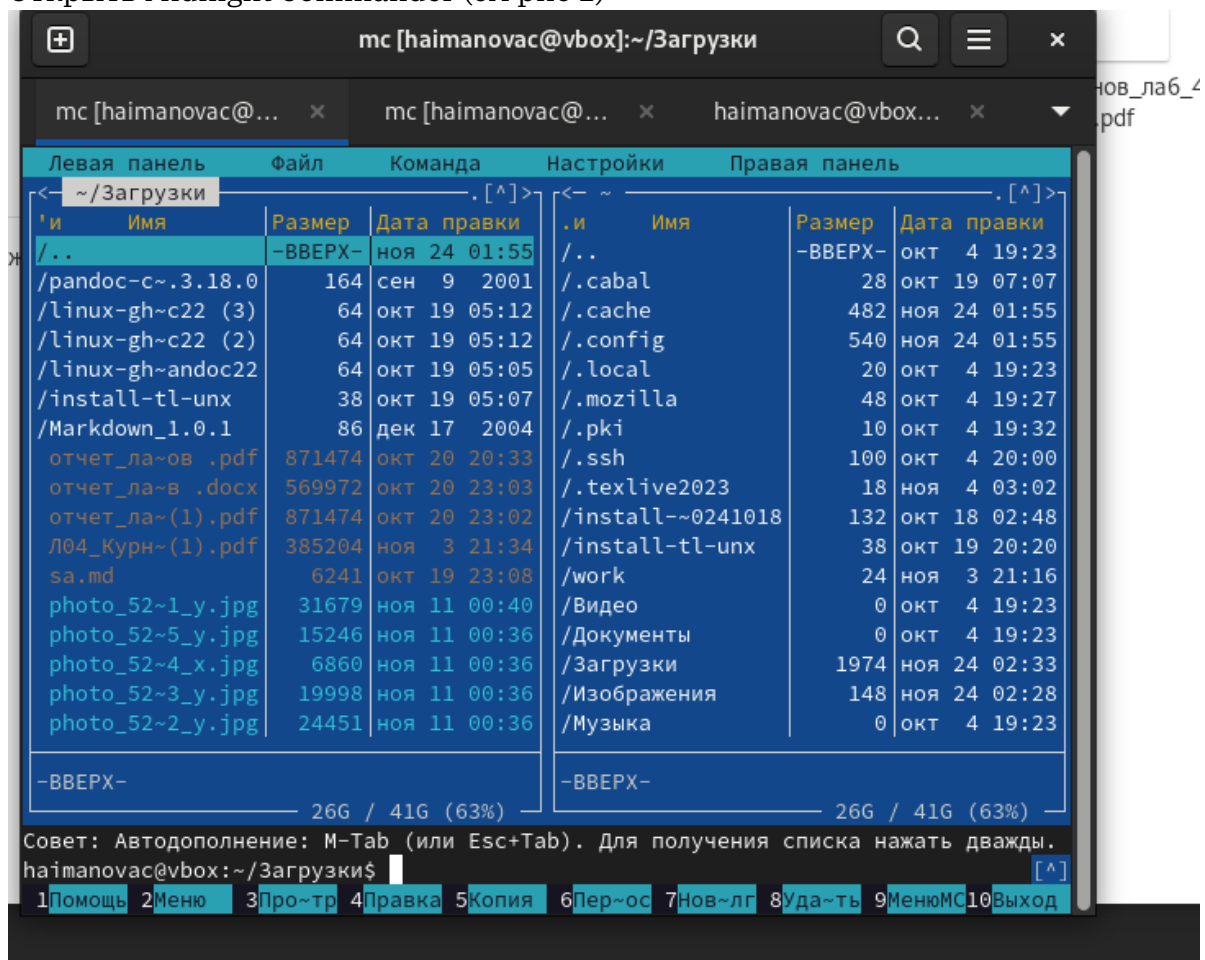
Таблица 1: Описание некоторых каталогов файловой системы GNU Linux {#tbl:std-dir}

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].

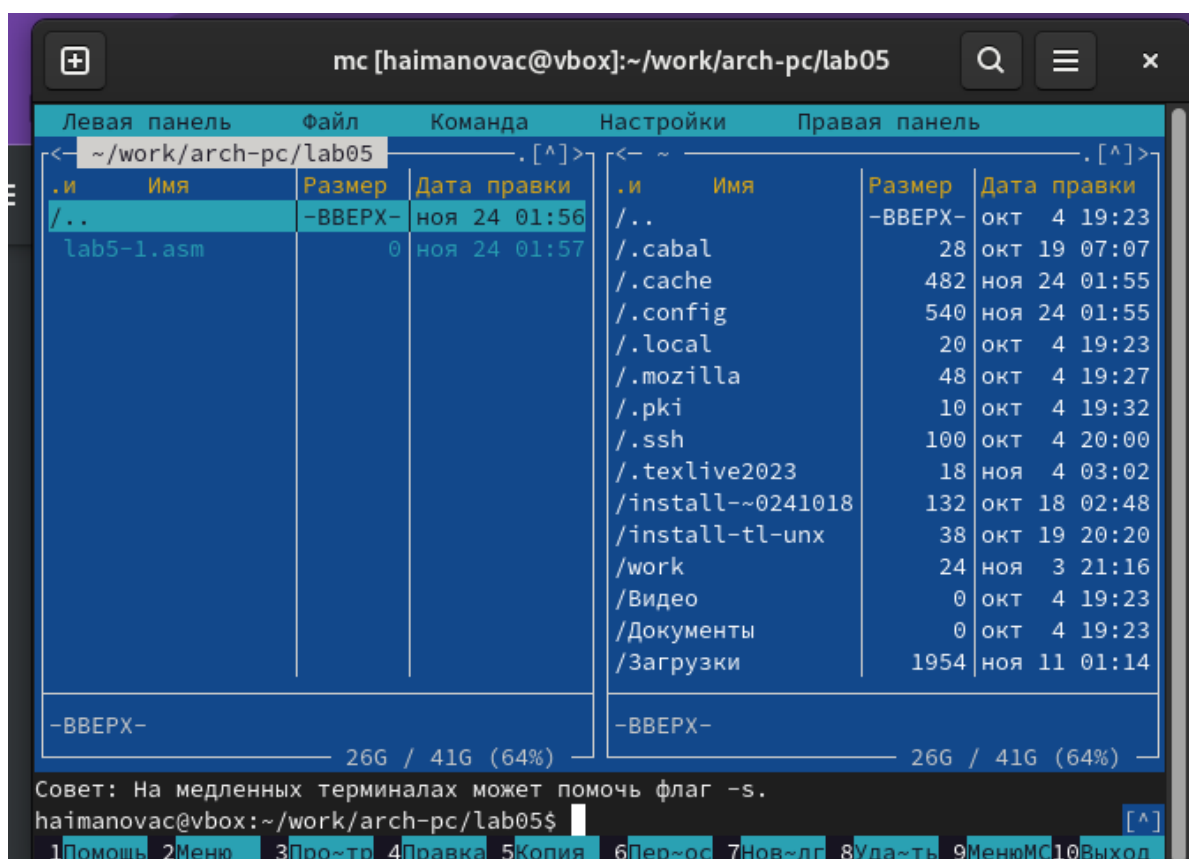
Выполнение лабораторной работы

1. Открыть Midnight Commander (см рис 1)



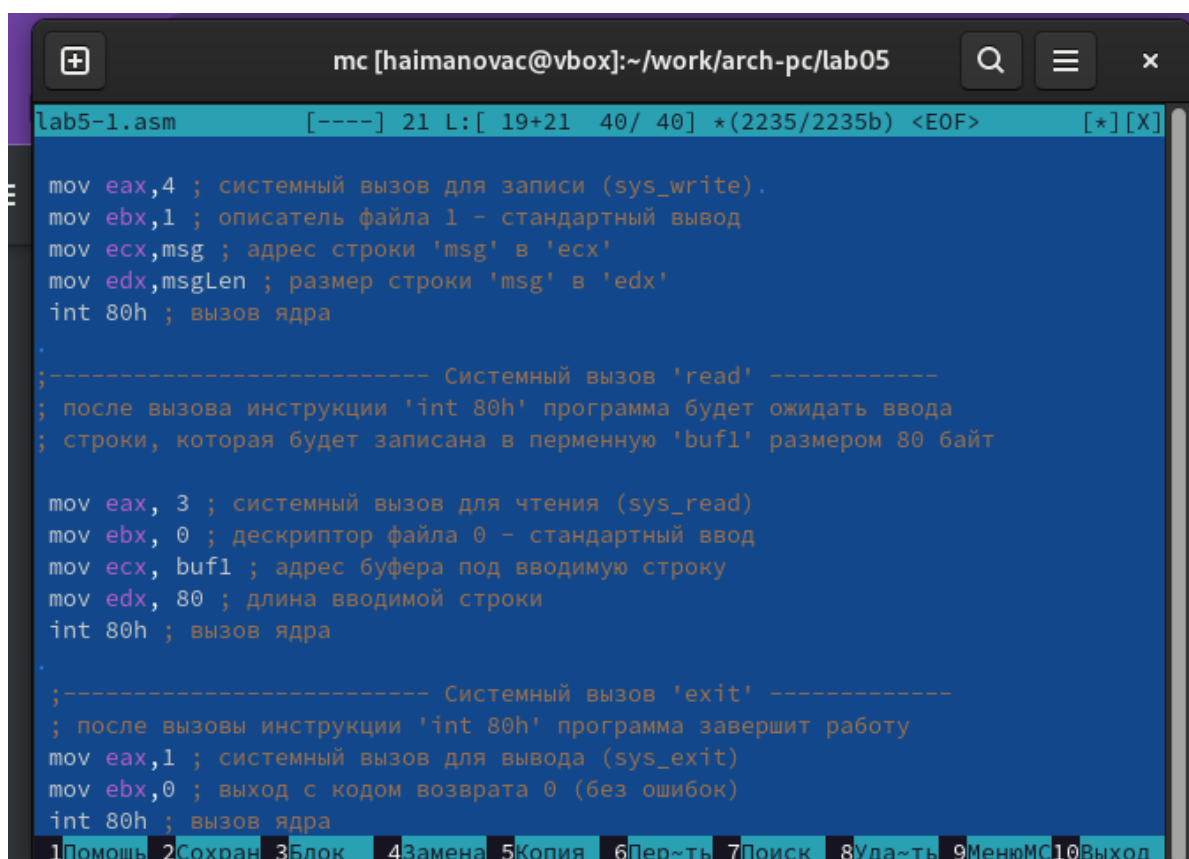
Открытый MC (рис 1)

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)



```
mc [haimanovac@vbox]:~/work/arch-pc/lab05
lab5-1.asm [----] 21 L: [ 19+21 40/ 40] *(2235/2235b) <EOF> [*] [X]

mov eax,4 ; системный вызов для записи (sys_write).
mov ebx,1 ; дескриптор 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 вывода (sys_exit)
mov ebx,0 ; выход с кодом возврата 0 (без ошибок)
int 80h ; вызов ядра

1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перейти 7Поиск 8Удалить 9МенюМС10Выход
```

Открытый файл lab5-1.asm (рис 3)

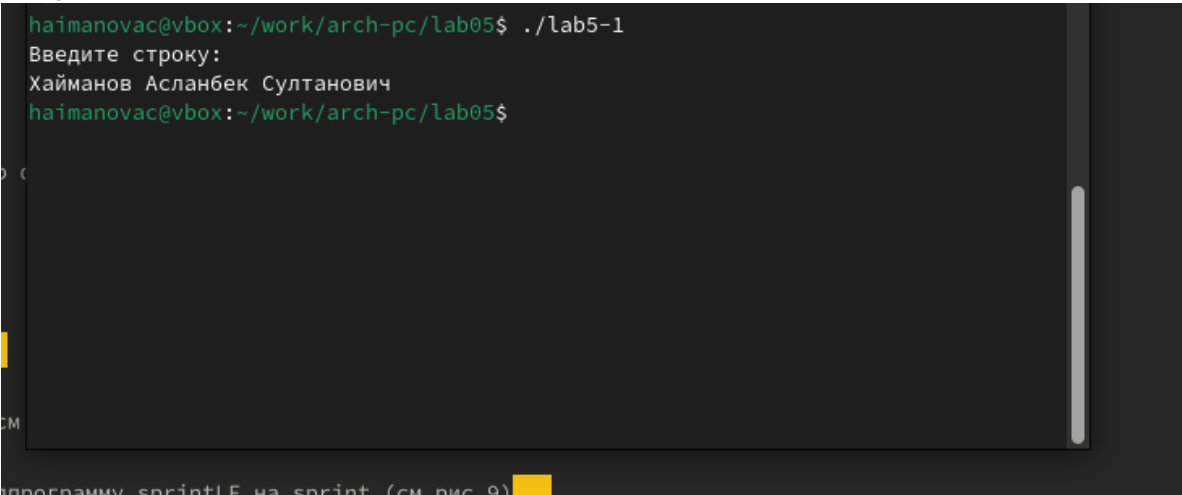
4. Убедится что файл содержит информацию (см рис 3)
5. Оттранслировать текст файла lab5-1.asm, выполнить компоновку объектного файла (см рис 4)



```
haimanovac@vbox:~/work/arch-pc/lab05$ touch lab5-1.asm
haimanovac@vbox:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
haimanovac@vbox:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
haimanovac@vbox:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
```

Выполнение команд (рис 4)

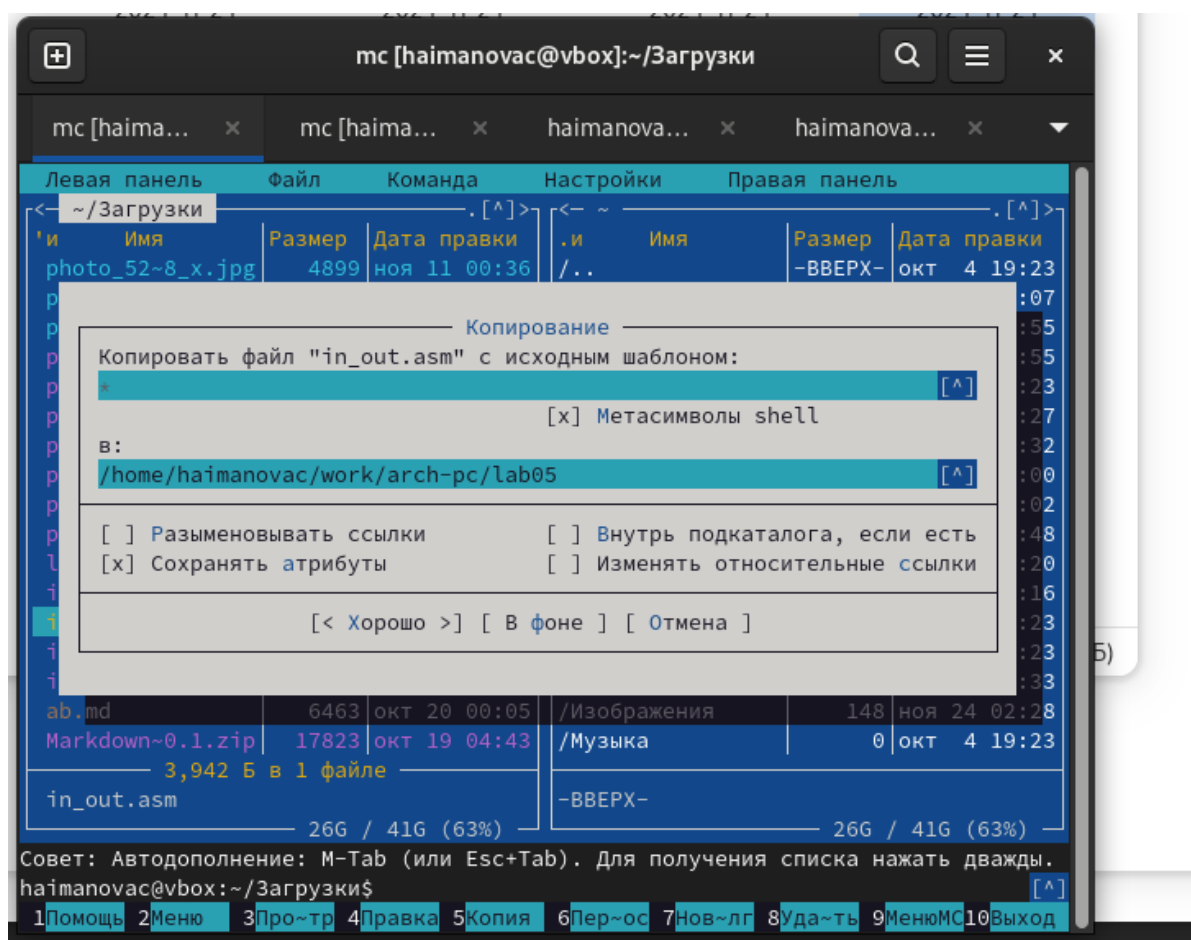
6. Запустить файл (см рис 5)

A terminal window with a dark background. The prompt is 'haiimanovac@vbox:~/work/arch-pc/lab05\$'. The command './lab5-1' has been entered. The output shows 'Введите строку:' followed by the name 'Хайманов Асланбек Султанович' on the next line. The prompt returns to 'haiimanovac@vbox:~/work/arch-pc/lab05\$'.

```
haiimanovac@vbox:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Хайманов Асланбек Султанович
haiimanovac@vbox:~/work/arch-pc/lab05$
```

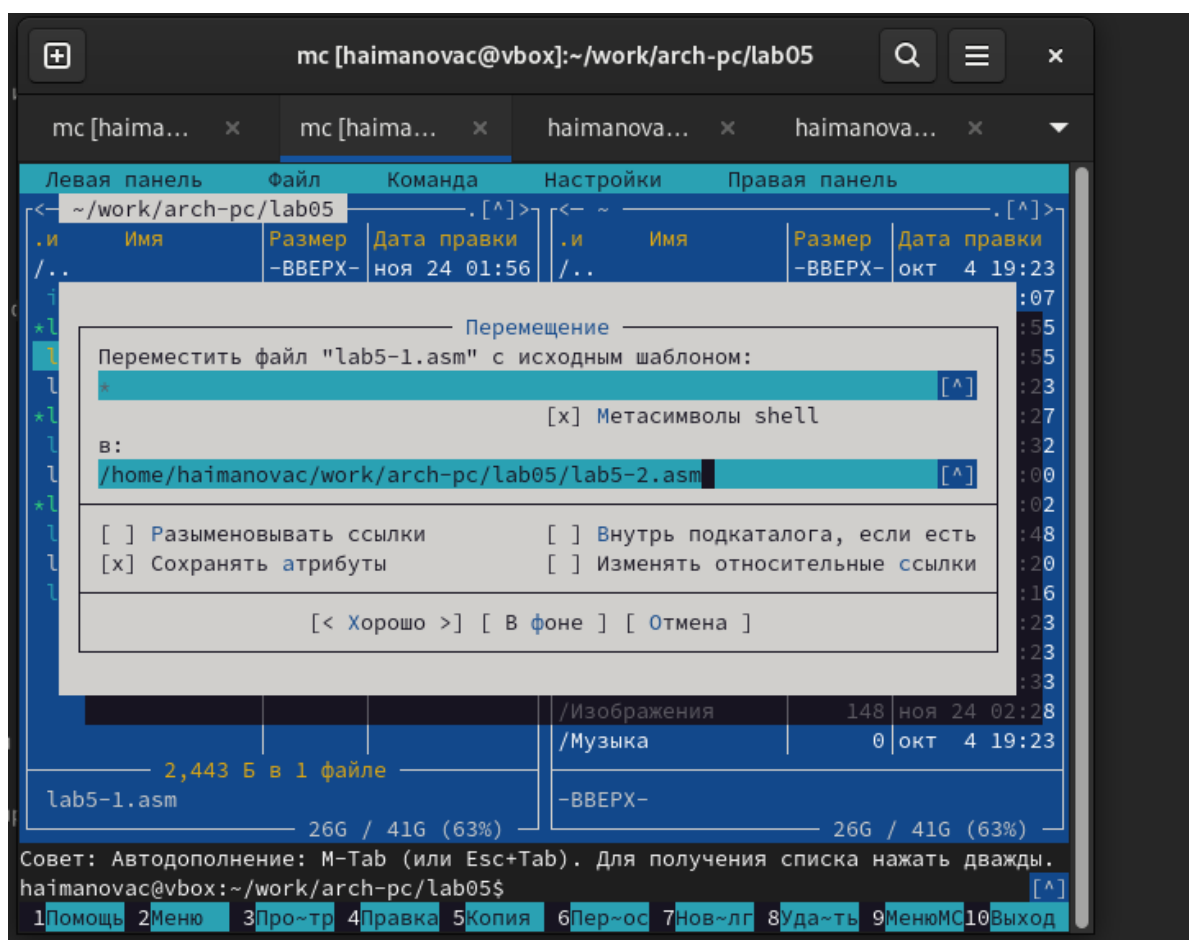
Запуск файла (рис 5)

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



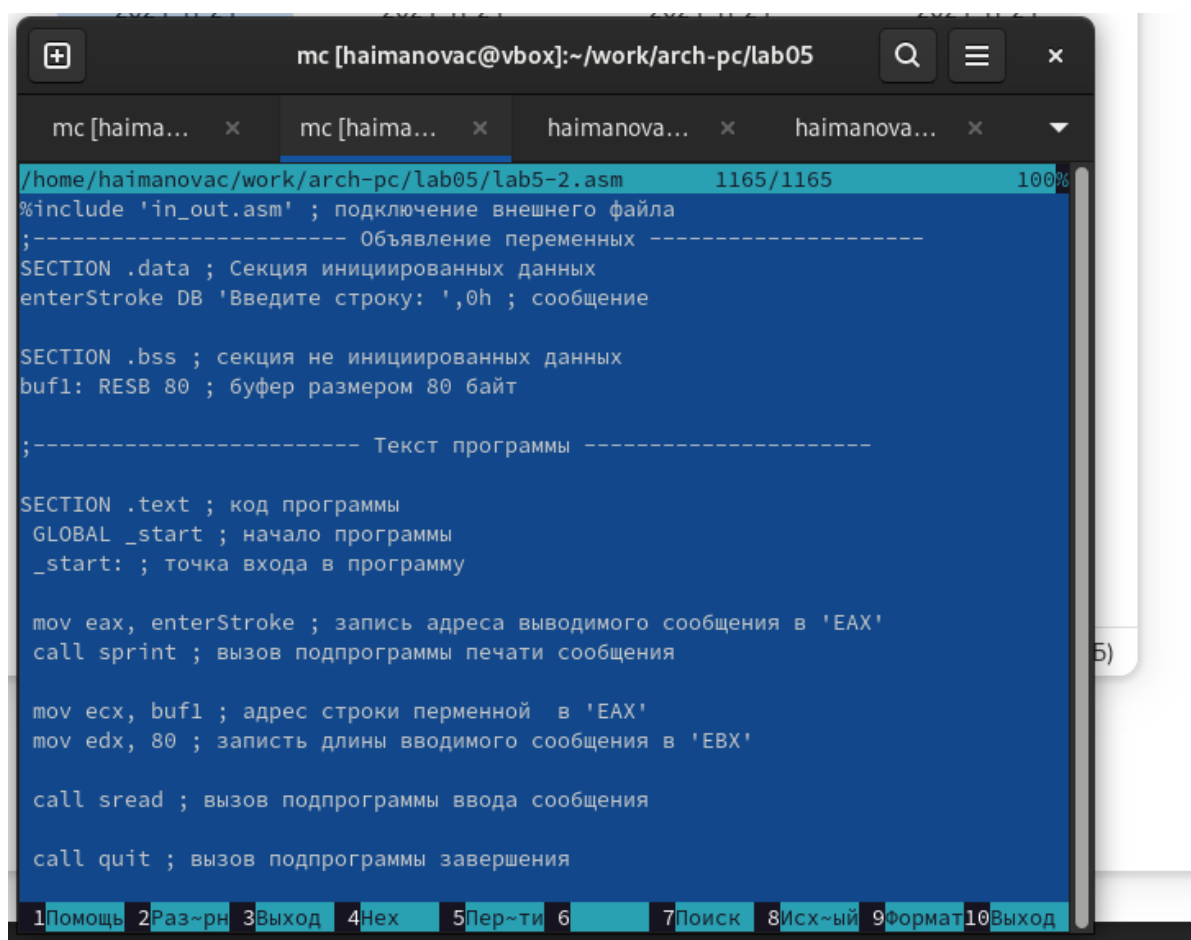
Скопированный in_out.asm через f5 (рис 6)

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)

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



```
mc [haimanovac@vbox]:~/work/arch-pc/lab05
/home/haimanovac/work/arch-pc/lab05/lab5-2.asm 1165/1165 100%
%include 'in_out.asm' ; подключение внешнего файла
;----- Объявление переменных -----
SECTION .data ; Секция иницированных данных
enterStroke DB 'Введите строку: ',0h ; сообщение

SECTION .bss ; секция не иницированных данных
buf1: RESB 80 ; буфер размером 80 байт

;----- Текст программы -----

SECTION .text ; код программы
GLOBAL _start ; начало программы
_start: ; точка входа в программу

mov eax, enterStroke ; запись адреса выводимого сообщения в 'EAX'
call sprint ; вызов подпрограммы печати сообщения

mov ecx, buf1 ; адрес строки переменной в 'EAX'
mov edx, 80 ; запись длины вводимого сообщения в 'EBX'

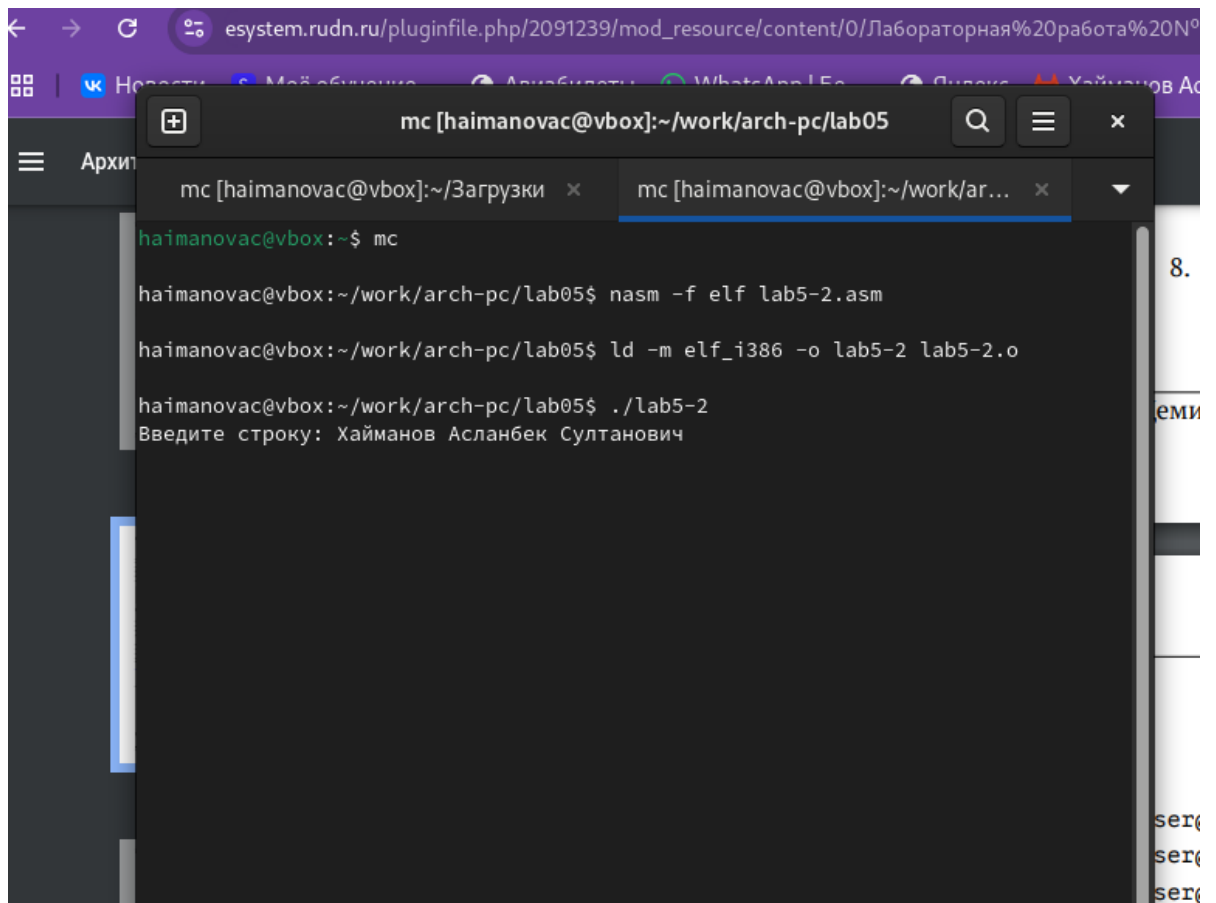
call sread ; вызов подпрограммы ввода сообщения

call quit ; вызов подпрограммы завершения

1Помощь 2Раз~рн 3Выход 4Нех 5Пер~ти 6 7Поиск 8Исх~ый 9Формат10Выход
```

Исправленный файл lab5-2.asm (рис 8)

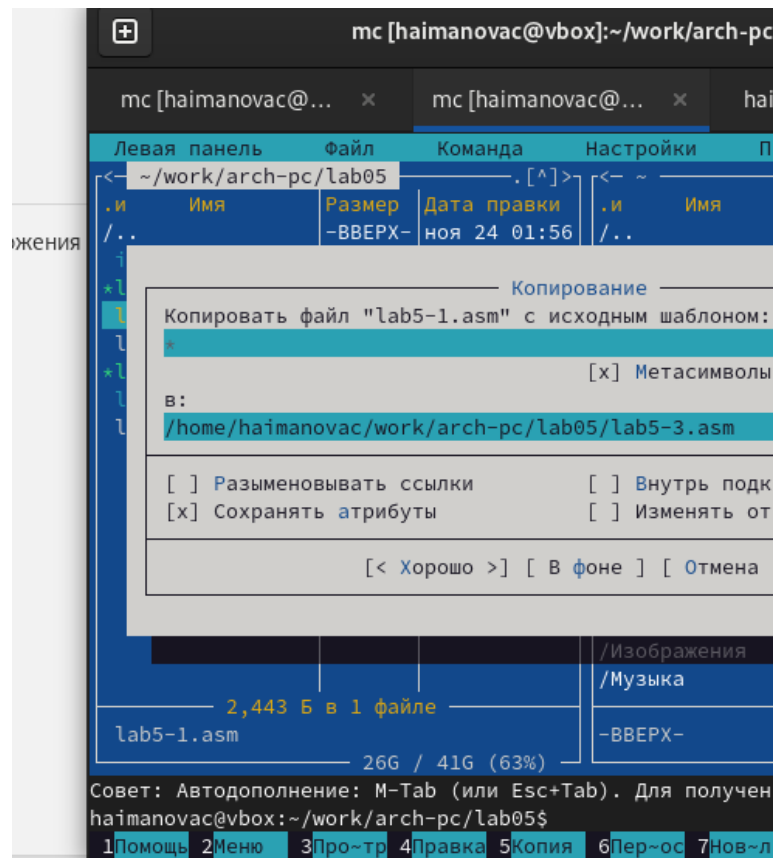
10. Создать исполняемый файл и проверить его работу (см рис 9)



Проверка и создание файла (рис 10)

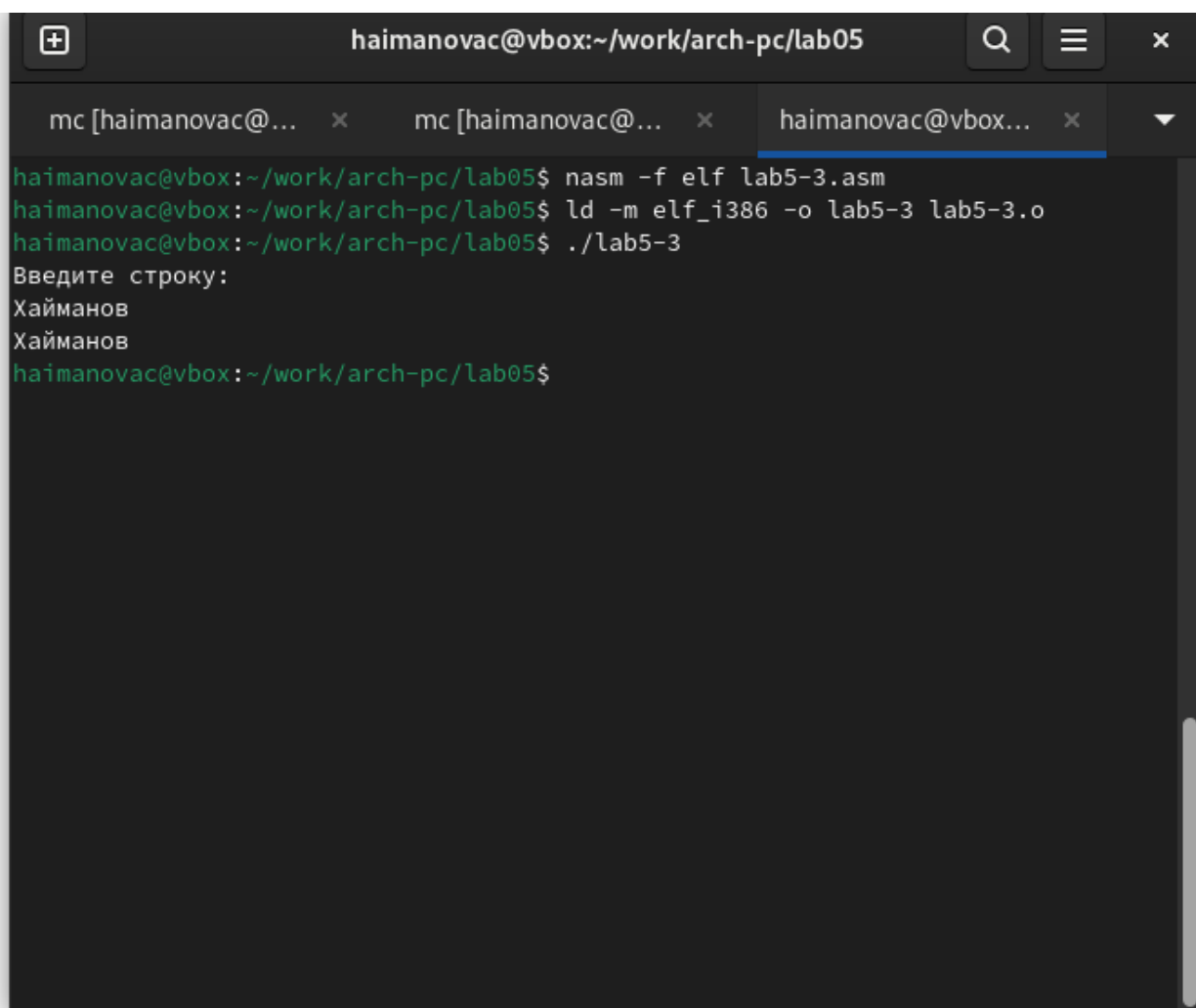
Теперь после вывода сообщения не будет перехода на новую строку

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



денная строка на экран (см рис 10-11)

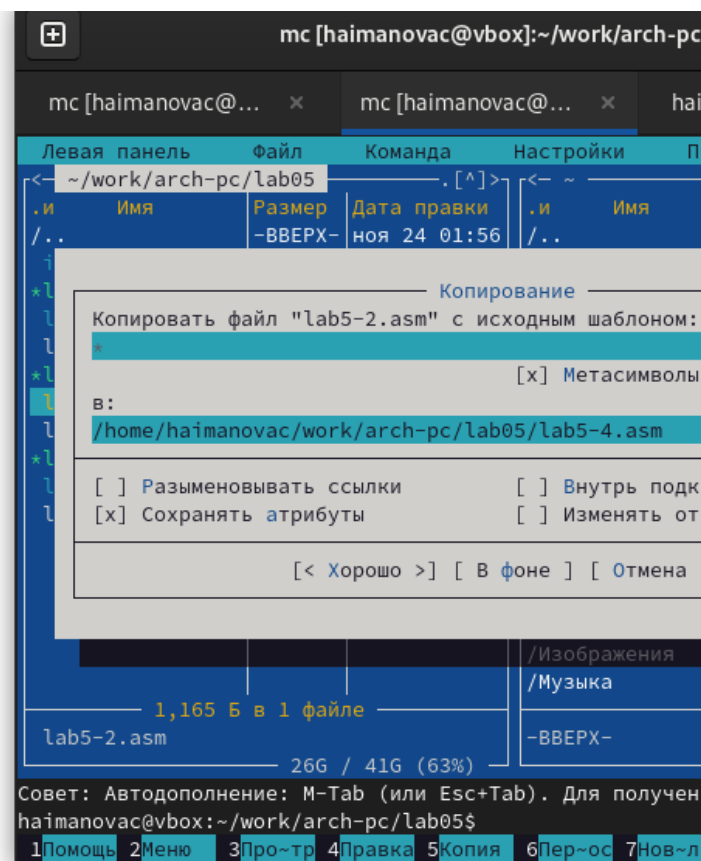
Создание копии файла lab5-1.asm (рис 10)



```
haimanovac@vbox:~/work/arch-pc/lab05
mc [haimanovac@... x mc [haimanovac@... x haimanovac@vbox... x
haimanovac@vbox:~/work/arch-pc/lab05$ nasm -f elf lab5-3.asm
haimanovac@vbox:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-3 lab5-3.o
haimanovac@vbox:~/work/arch-pc/lab05$ ./lab5-3
Введите строку:
Хайманов
Хайманов
haimanovac@vbox:~/work/arch-pc/lab05$
```

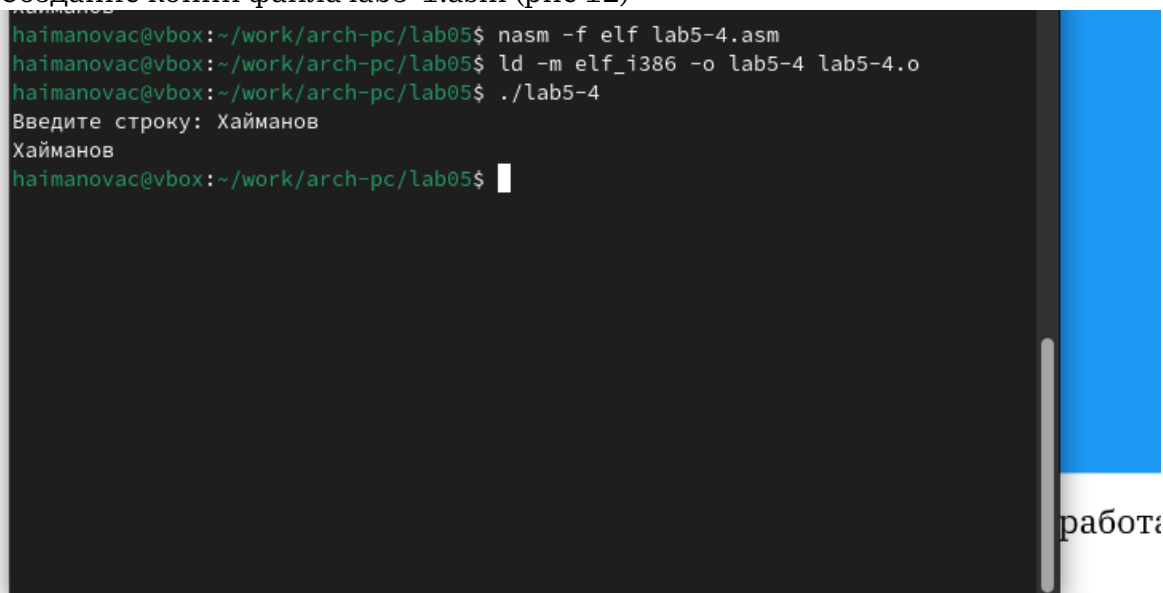
Вывод введенной строки на экран (рис 11)

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)



4. Создал исполняемый файл и проверил его работу (рис

Вывод введенной строки на экран (рис 13)

Выводы

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

Список литературы