

# **Отчёт по лабораторной работе №5**

**дисциплина: Архитектура компьютера**

Гайдук Софья Сергеевна

# **Содержание**

<b>1 Цель работы</b>	<b>5</b>
<b>2 Выполнение лабораторной работы</b>	<b>6</b>
<b>3 Задания для самостоятельной работы</b>	<b>16</b>
<b>4 Выводы</b>	<b>19</b>

# Список иллюстраций

2.1	image1	6
2.2	image2	6
2.3	image3	7
2.4	image4	8
2.5	image5	8
2.6	image6	9
2.7	image7	9
2.8	image8	10
2.9	image9	10
2.10	image10	10
2.11	image11	10
2.12	image12	11
2.13	image13	11
2.14	image14	12
2.15	image15	12
2.16	image16	13
2.17	image17	13
2.18	image18	14
2.19	image19	14
2.20	image20	15
2.21	image21	15
3.1	image22	16
3.2	image23	17
3.3	image24	17
3.4	image25	18
3.5	image26	18

# **Список таблиц**

# **1 Цель работы**

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

## 2 Выполнение лабораторной работы

Откроем Midnight Commander (рис. 2.1).

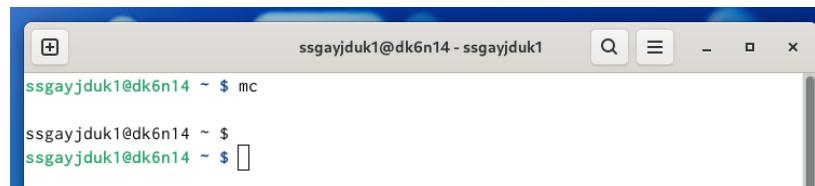


Рисунок 2.1: image1

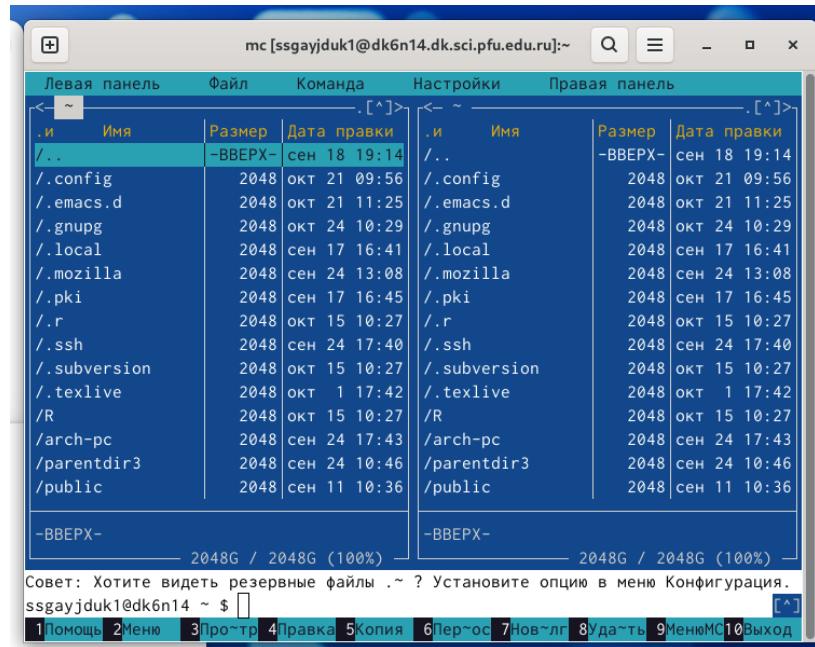


Рисунок 2.2: image2

Перейдем в каталог `~/work/arch-pc` созданный при выполнении лабораторной работы №4 с помощью клавиш (рис. 2.3).

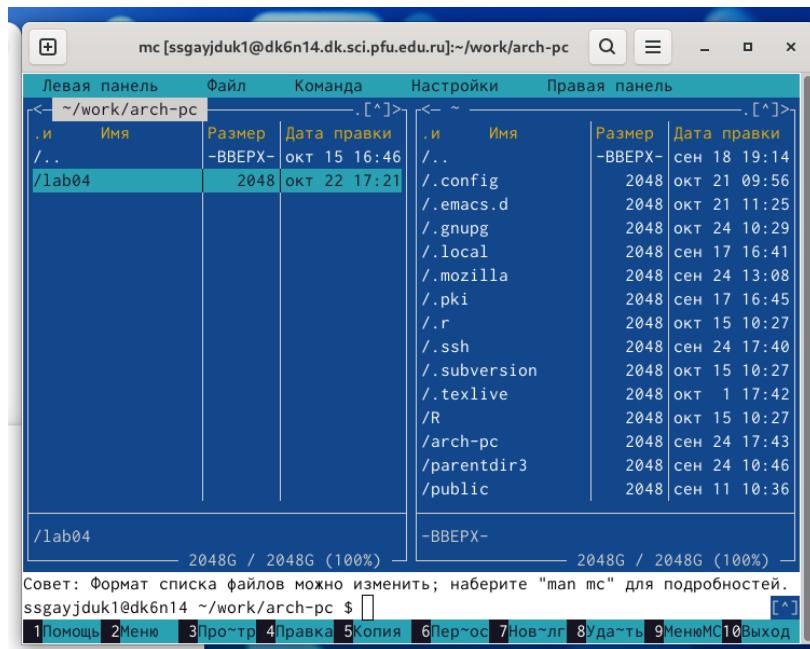


Рисунок 2.3: image3

С помощью функциональной клавиши F7 создим папку lab05 и перейдем в созданный каталог (рис. 2.4).

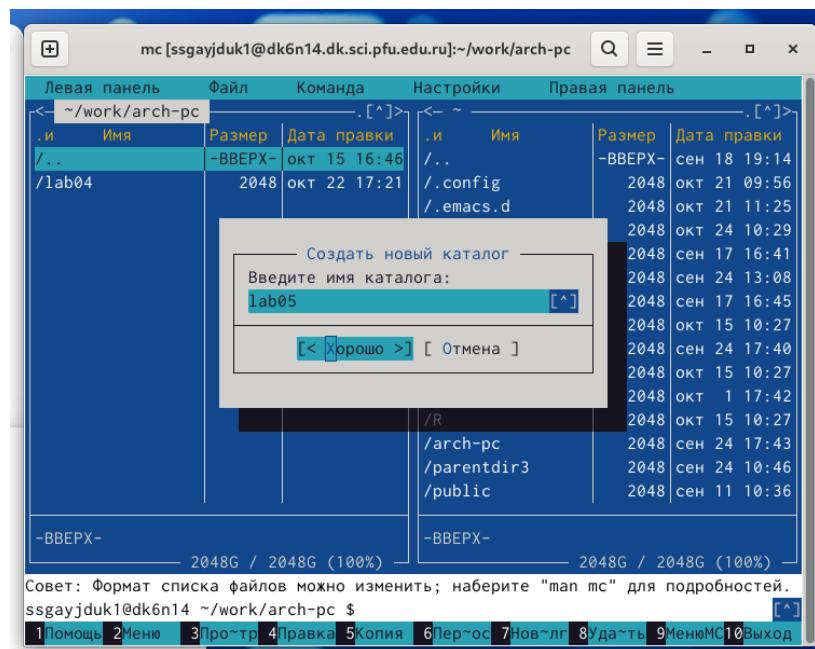


Рисунок 2.4: image4

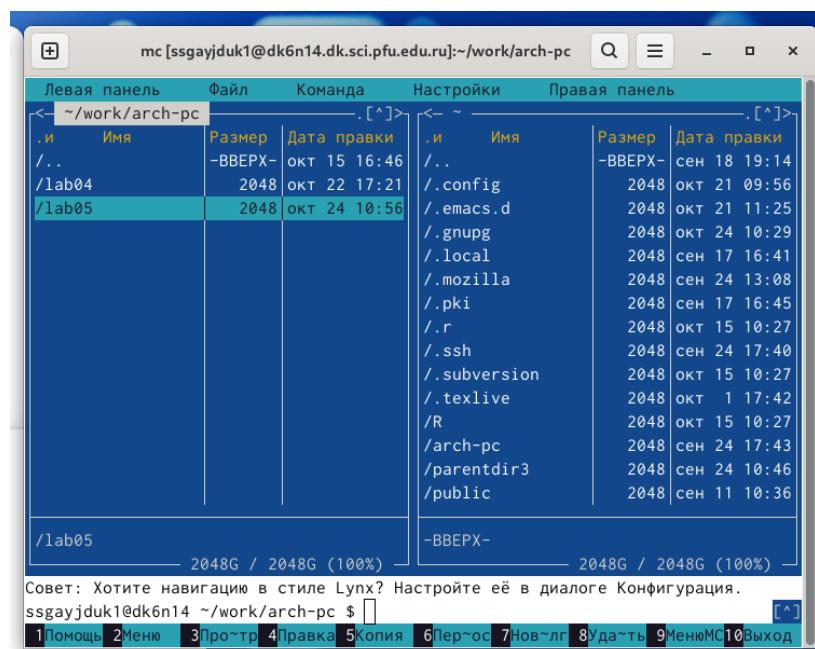


Рисунок 2.5: image5

Пользуясь строкой ввода и командой touch создадим файл lab5-1.asm (рис. 2.6).

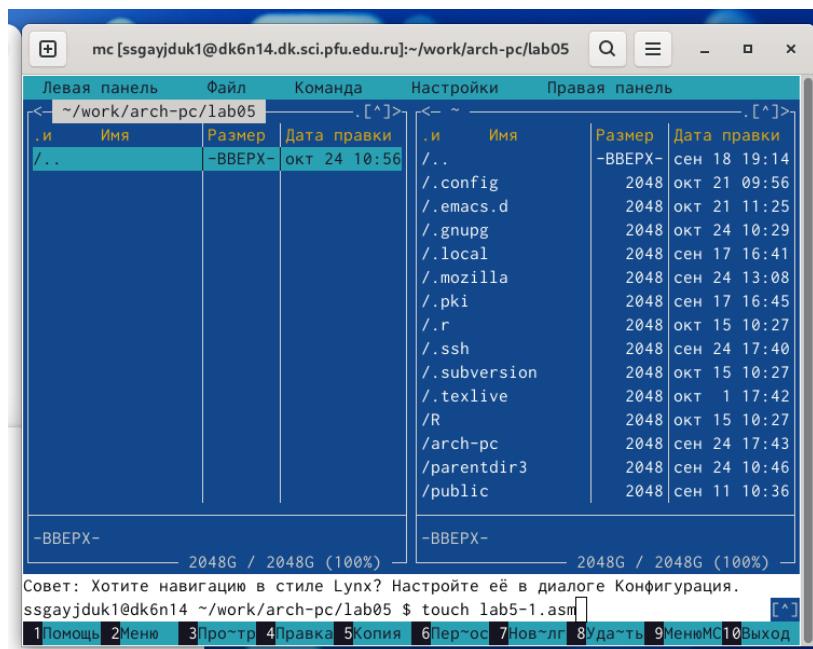


Рисунок 2.6: image6

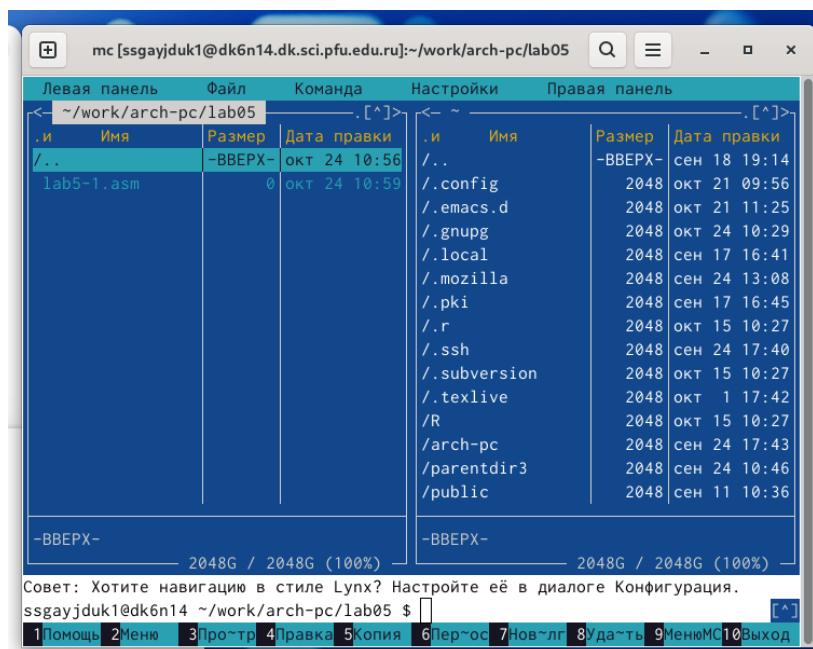


Рисунок 2.7: image7

С помощью функциональной клавиши F4 откроем файл lab5-1.asm для редактирования во встроенным редакторе mcedit (рис. 2.8)

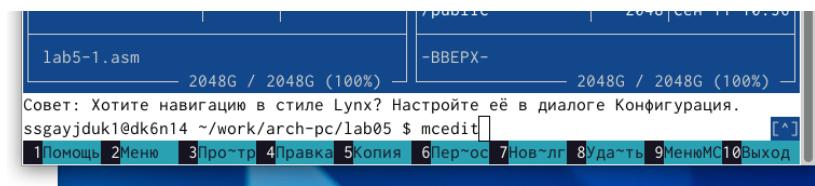


Рисунок 2.8: image8

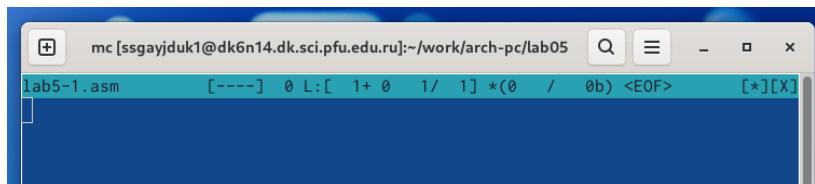


Рисунок 2.9: image9

А также воспользуемся редактором nano (рис. 2.10)

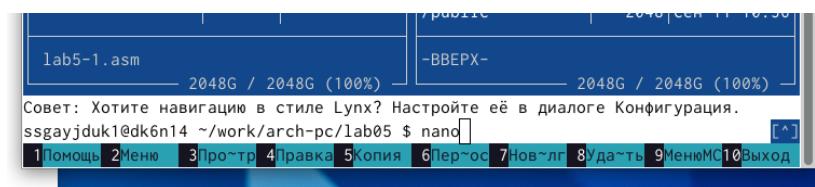


Рисунок 2.10: image10

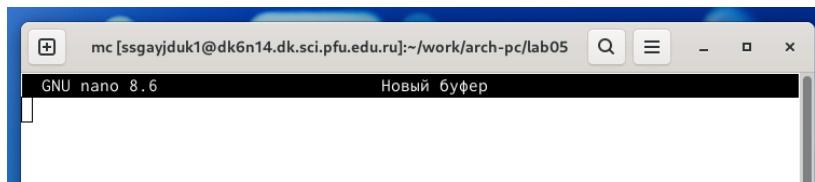


Рисунок 2.11: image11

Ввели текст программы из листинга, сохранили изменения и закрыли файл. Делали с помощью команды nano. Ctrl + X (выход), Y (сохранить изменения), Enter (рис. 2.12)

```
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;----- Обявление 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
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'
[ Прочитано 34 строки ]
^G Справка ^O Записать ^F Поиск ^K Вырезать ^T Выполнить M-U Отмена
^X Выход ^R Читать ^\ Замена ^U Вставить ^C Позиция M-E Повтор
```

Рисунок 2.12: image12

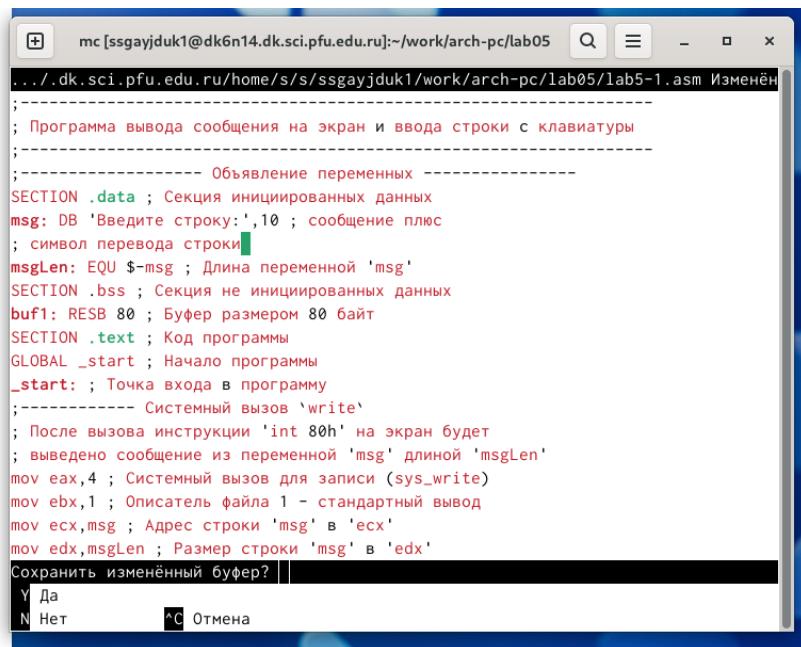


Рисунок 2.13: image13

С помощью функциональной клавиши F3 открыли файл lab5-1.asm для просмотра. Убедились, что файл содержит текст программы (рис. 2.14).

The screenshot shows the Mars Debugger (mc) interface. The main window displays the assembly code for the file `lab5-1.asm`. The code is written in AT&T syntax and includes directives like `SECTION .data`, `SECTION .text`, and `GLOBAL _start`. It also contains instructions for reading from the keyboard and writing to the screen. The status bar at the bottom shows various menu options in Russian: Помощь (Help), Разгрыз (Break), Выход (Exit), Нех (Hex), Пер~ти (Jump), Поиск (Search), Иск~ый (Find), Формат (Format), and Выход (Exit).

Рисунок 2.14: image14

Оттранслируем текст программы `lab5-1.asm` в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл. Введем свое ФИО (рис. 2.15).

The terminal window shows the following commands and output:

```
ssgayjduk1@dk6n14 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.asm
ssgayjduk1@dk6n14 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o
ssgayjduk1@dk6n14 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку:
Гайдук Софья Сергеевна
ssgayjduk1@dk6n14 ~/work/arch-pc/lab05 $
```

Рисунок 2.15: image15

Откроем каталог с файлом `lab5-1.asm` в панели mc. В другой панели откроем каталог, в нашем случае Загрузки, с файлом `in_out.asm`. Скопируем файл `in_out.asm` в каталог с файлом `lab5-1.asm` с помощью функциональной клавиши F5 (рис. 2.16)

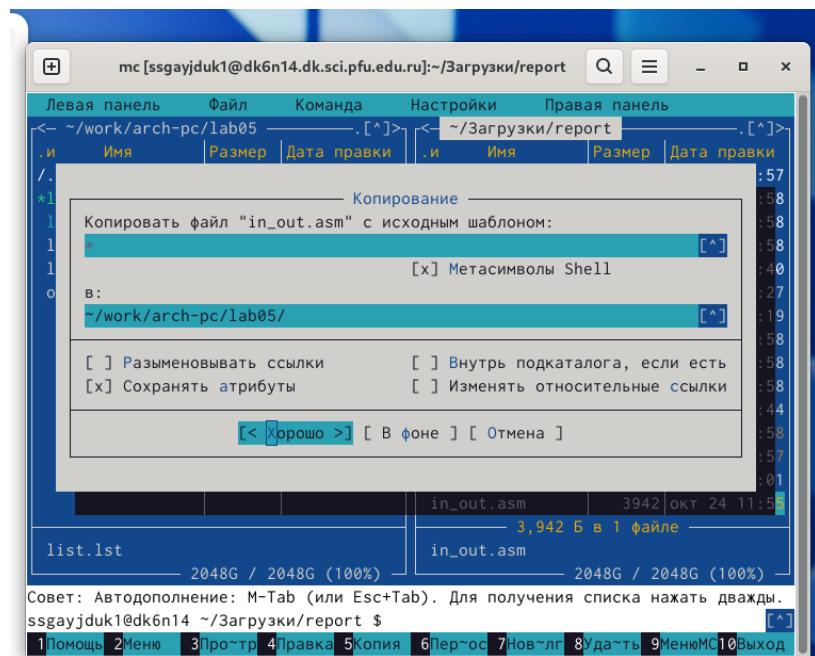


Рисунок 2.16: image16

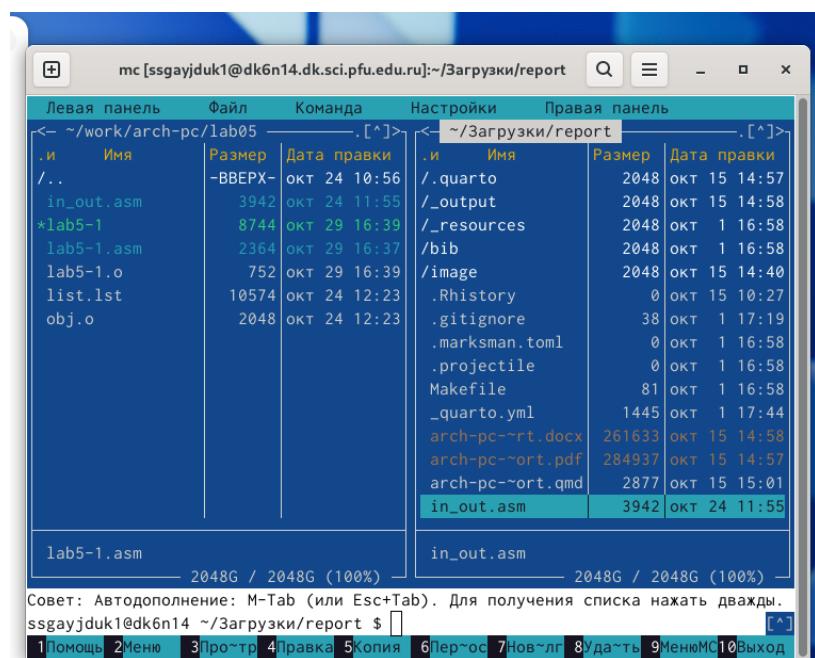


Рисунок 2.17: image17

С помощью функциональной клавиши F6 создадим копию файла lab5-1.asm с именем lab5-2.asm (рис. 2.18)

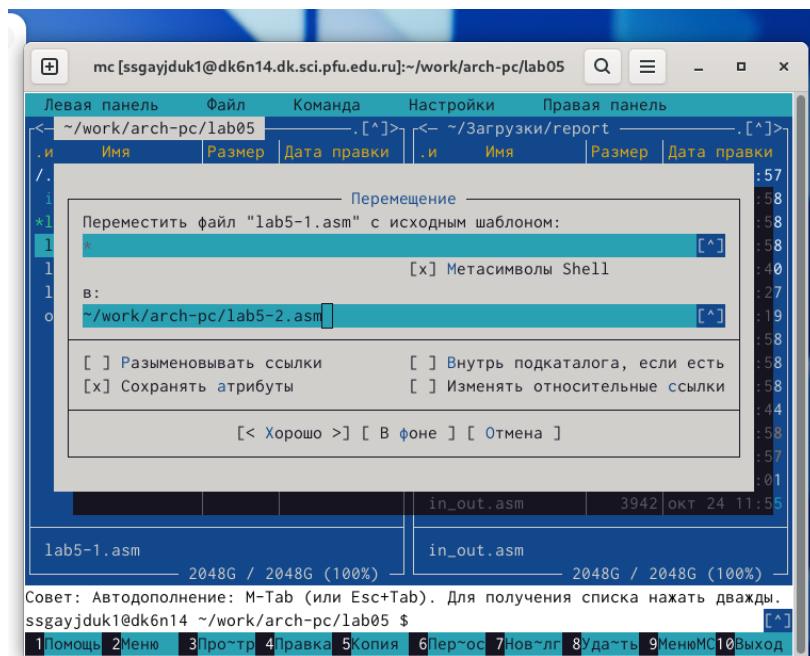


Рисунок 2.18: image18

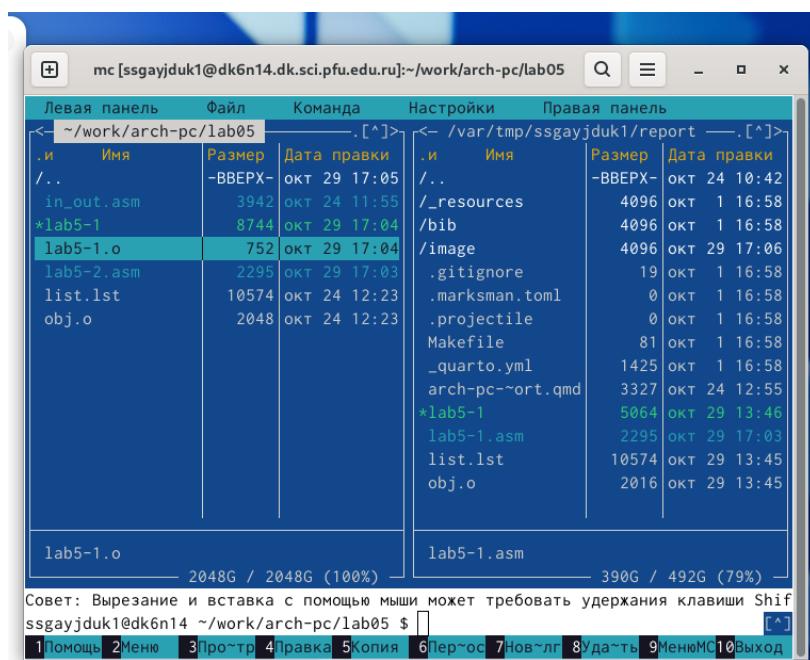


Рисунок 2.19: image19

Исправим текст программы в файле lab5-2.asm с использованием подпрограмм in\_out.asm. Создадим исполняемый файл и проверим его работу.(рис. 2.20).

```
ssgayjduk1@dk6n14 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
ssgayjduk1@dk6n14 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
ssgayjduk1@dk6n14 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку:
Гайдук Софья Сергеевна
ssgayjduk1@dk6n14 ~/work/arch-pc/lab05 $
```

Рисунок 2.20: image20

В файле lab5-2.asm заменим подпрограмму sprintLF на sprint. Создадим исполняемый файл и проверим его работу (рис. 2.21).

```
ssgayjduk1@dk6n14 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
ssgayjduk1@dk6n14 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
ssgayjduk1@dk6n14 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку:
Гайдук Софья Сергеевна
ssgayjduk1@dk6n14 ~/work/arch-pc/lab05 $
ssgayjduk1@dk6n14 ~/work/arch-pc/lab05 $
ssgayjduk1@dk6n14 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
ssgayjduk1@dk6n14 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
ssgayjduk1@dk6n14 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку: Гайдук Софья Сергеевна
ssgayjduk1@dk6n14 ~/work/arch-pc/lab05 $
```

Рисунок 2.21: image21

Разница в том, что ввод своего ФИО происходит на следующей строке или на этой же строке.

### 3 Задания для самостоятельной работы

Создадим копию файла lab5-1.asm и назовем lab5copy-1.asm (рис. 3.1)

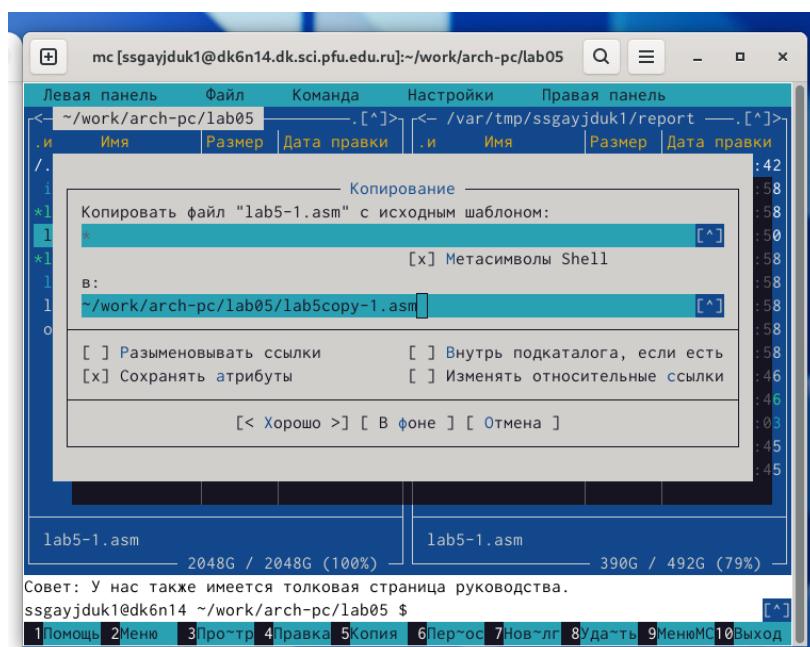


Рисунок 3.1: image22

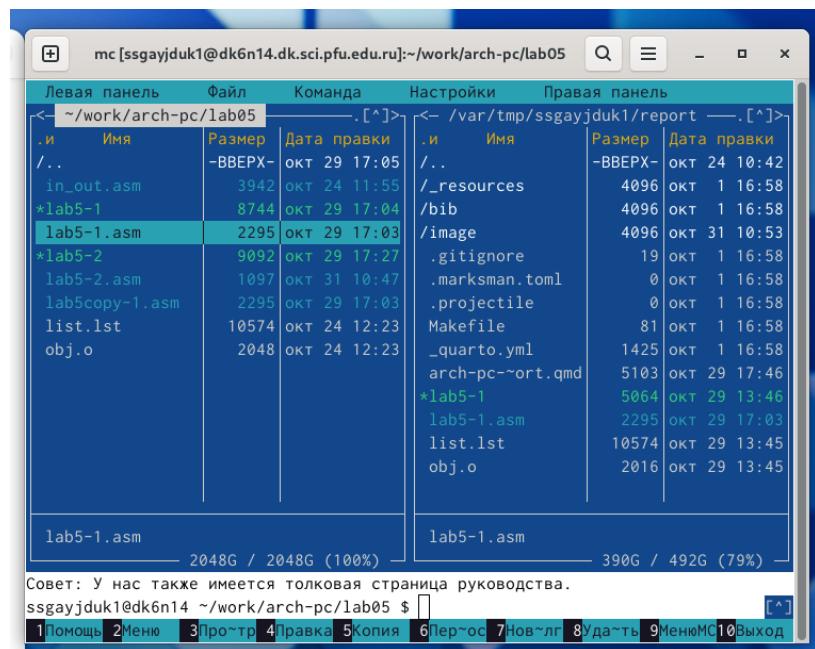


Рисунок 3.2: image23

Внесем изменения в программу и проверим работу файла: вывод приглашение типа «Введите строку:» ввод строки с клавиатуры, вывод введённой строки на экран (рис. 3.3)

```
ssgayjduk1@dk6n14 ~ /work/arch-pc/lab05 $ nasm -f elf lab5copy-1.asm
ssgayjduk1@dk6n14 ~ /work/arch-pc/lab05 $ ld -m elf_i386 -o lab5copy-1 lab5copy-1.o
ssgayjduk1@dk6n14 ~ /work/arch-pc/lab05 $ ./lab5copy-1
Введите строку:
Гайдук
Гайдук
ssgayjduk1@dk6n14 ~ /work/arch-pc/lab05 $
```

Рисунок 3.3: image24

Создадим копию файла lab5-2.asm и назовем lab5copy-2.asm (рис. 3.4)

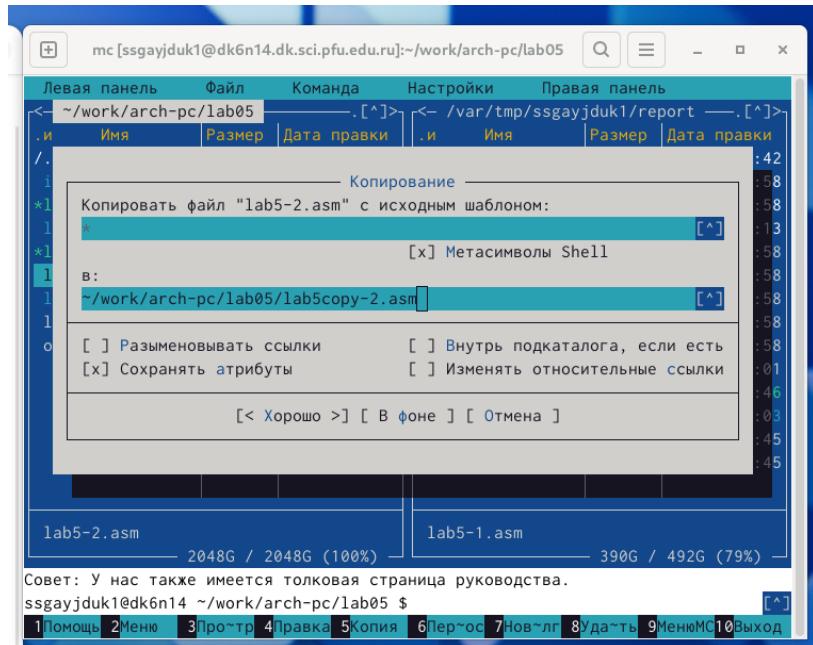


Рисунок 3.4: image25

Внесем изменения в программу (внешний файл `in_out.asm`) и проверим работу файла: вывод приглашение типа «Ведите строку:» ввод строки с клавиатуры, вывод введённой строки на экран (рис. 3.5)

```
ssgayjduk1@dk6n14 ~/work/arch-pc/lab05 $ nasm -f elf lab5copy-2.asm
ssgayjduk1@dk6n14 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5copy-2 lab5copy-2.o
ssgayjduk1@dk6n14 ~/work/arch-pc/lab05 $ ./lab5copy-2
Введите строку: Гайдук
Гайдук
ssgayjduk1@dk6n14 ~/work/arch-pc/lab05 $
```

Рисунок 3.5: image26

## **4 Выводы**

Мы приобрели практические навыки работы в Midnight Commander и освоили инструкции языка ассемблера mov и int.