

Лабораторная работа №5

НПИбд-01-25 №1032252598

Иванова Ангелина Олеговна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Задания лабораторной работы	6
2.2	Задание для самостоятельной работы	16
3	Выводы	22

Список иллюстраций

2.1	Каталог в Midnight Commander	7
2.2	Переход в каталог lab05	8
2.3	Созданный файл в каталоге lab05	9
2.4	Открытый файл в папо	10
2.5	Открытый файл с текстом программы	11
2.6	Создание исполняемого файла и его запуск	12
2.7	Копирование файла in_out.asm в каталог с файлом lab5-1.asm	13
2.8	Созданный файл lab5-2.asm	14
2.9	Отредактированный файл lab5-2.asm	15
2.10	Создание исполняемого файла и его запуск	16
2.11	Создание исполняемого файла с изменениями и его запуск	16
2.12	Копирование файлов	17
2.13	Изменение файла lab5-1.asm	19
2.14	Создание исполняемого файла и его запуск	19
2.15	Изменение файла lab5-2.asm	21
2.16	Создание исполняемого файла и его запуск	21

Список таблиц

1 Цель работы

Целью данной лабораторной работы является приобретение практических навыков работы в Midnight Commander, а также освоение инструкций языка ассемблера `mov` и `int`

2 Выполнение лабораторной работы

2.1 Задания лабораторной работы

2.1.1 Написание программ без использования внешнего файла

in_out.asm

Открыли Midnight Commander с помощью команды `mc` и пользуясь клавишами `↑`, `↓` и `Enter` перешли в каталог `~/work/arch-pc` созданный при выполнении 4 лабораторной работы

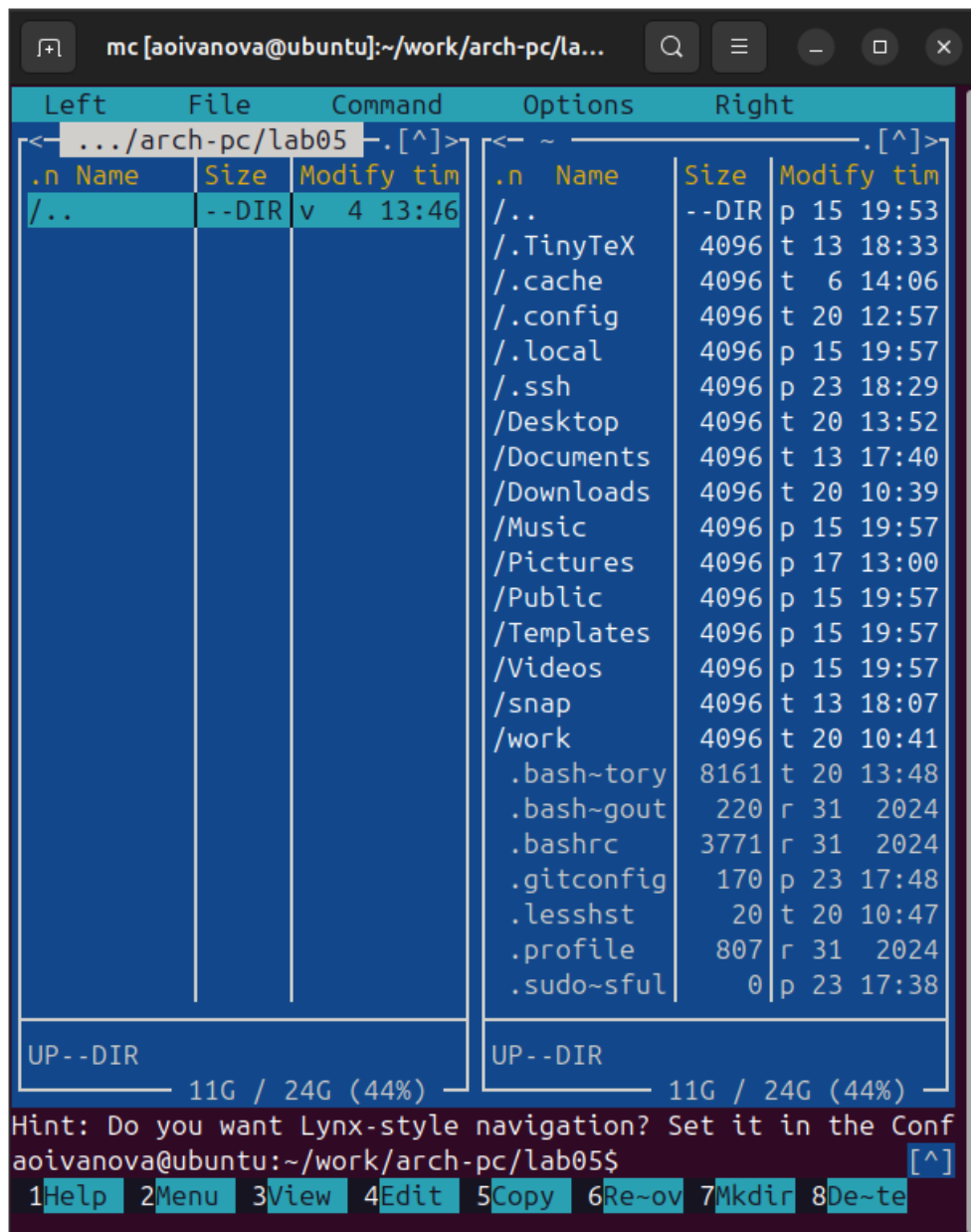


Рисунок 2.2: Переход в каталог lab05

Пользуясь строкой ввода и командой touch создали файл lab5-1.asm

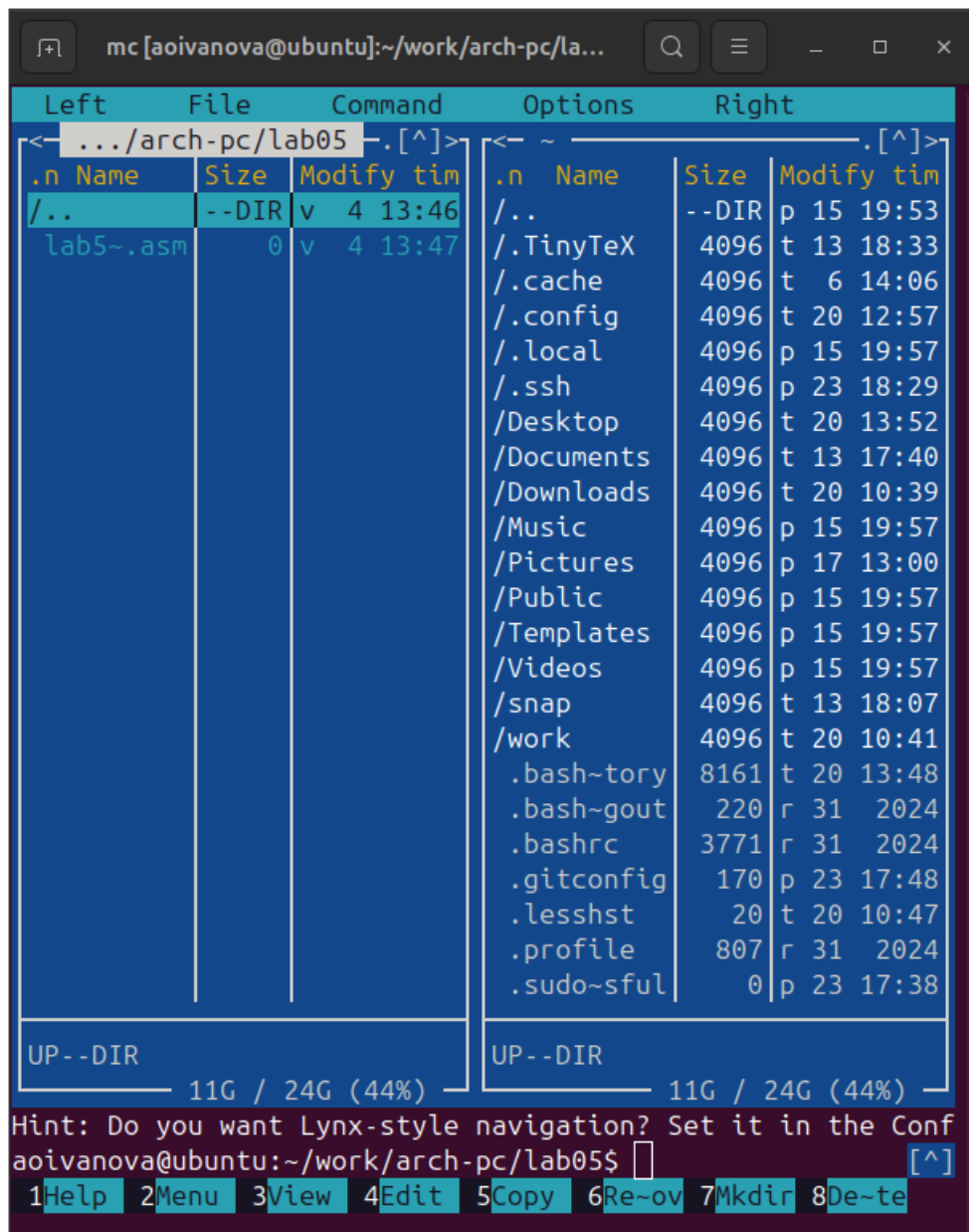


Рисунок 2.3: Созданный файл в каталоге lab05

С помощью функциональной клавиши F4 открыли файл lab5-1.asm для редактирования во встроенном редакторе nano

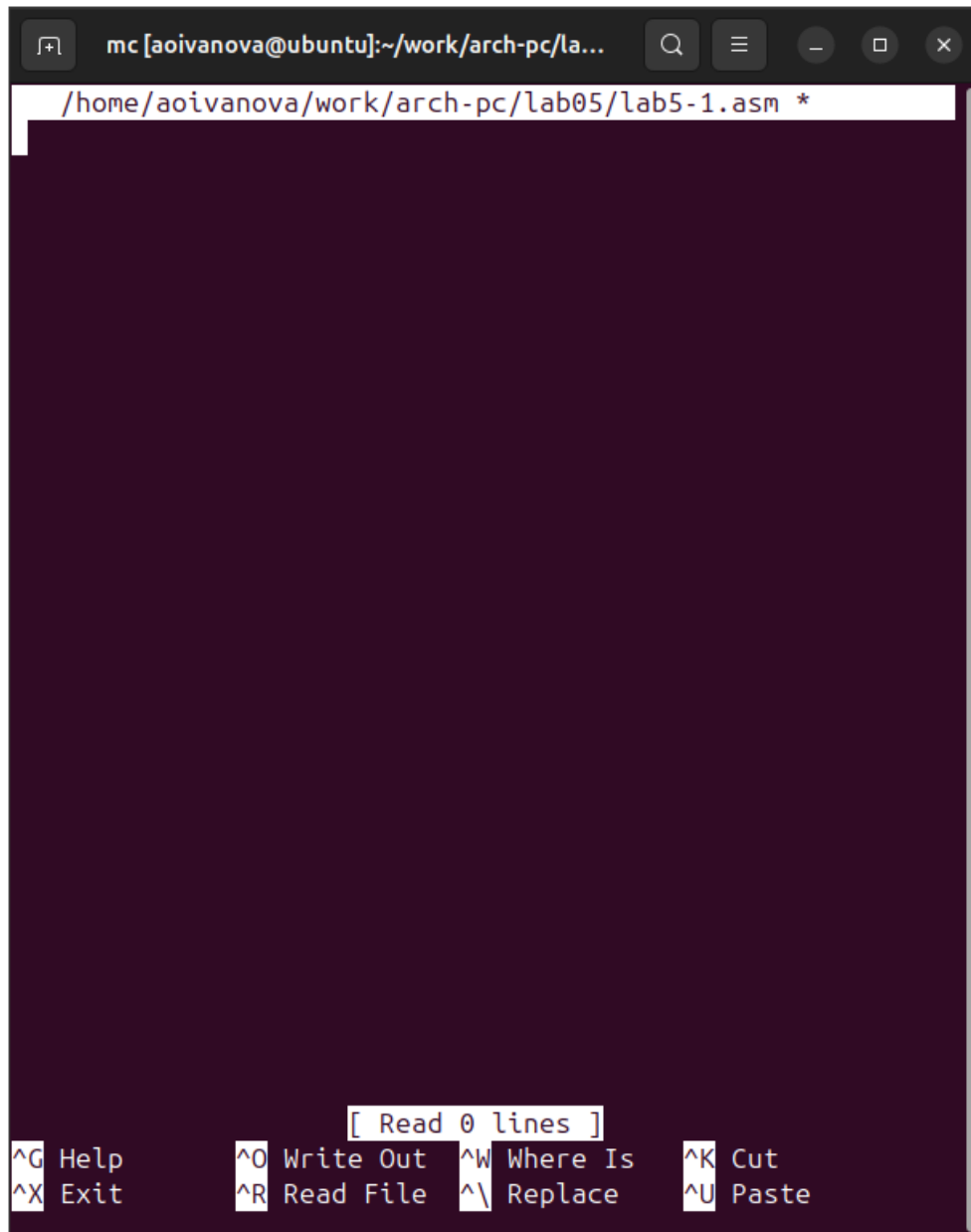
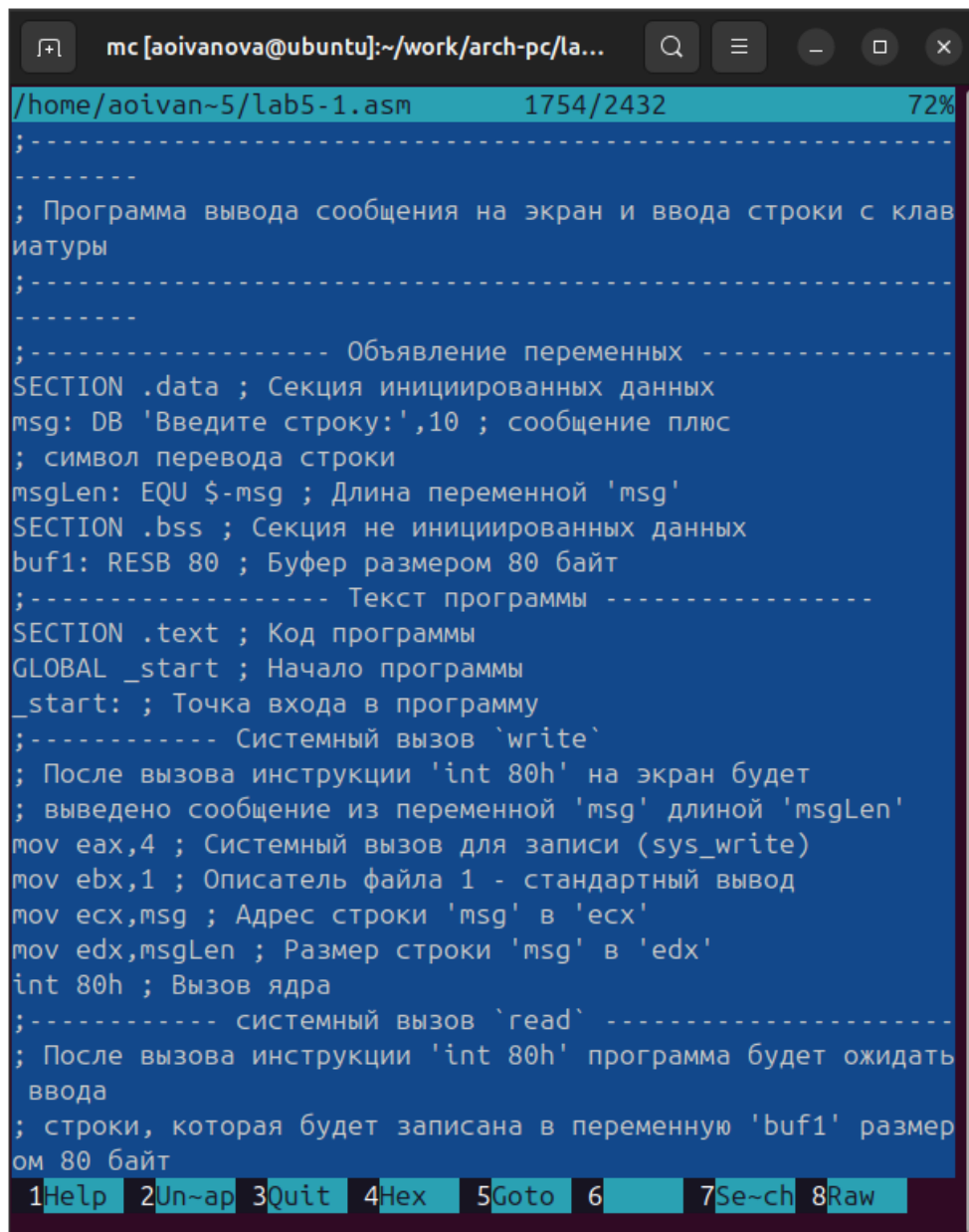


Рисунок 2.4: Открытый файл в nano

Ввели текст программы из первого предоставленного нам листинга, сохранили изменения и закрыли файл. С помощью функциональной клавиши F3 открыли файл lab5-1.asm для просмотра и убедились, что файл содержит текст программы.



```
mc [aoivanova@ubuntu]:~/work/arch-pc/la...
/home/aoivan~5/lab5-1.asm 1754/2432 72%
;-----
; 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
1Help 2Un~ap 3Quit 4Hex 5Goto 6 7Se~ch 8Raw
```

Рисунок 2.5: Открытый файл с текстом программы

Оттранслировали текст программы lab5-1.asm в объектный файл. Выполнили компоновку объектного файла и запустили получившийся исполняемый файл. Программа вывела строку „Введите строку:“ и ожидает ввода с клавиатуры. На запрос ввели свою имя и фамилию.

```
aoivanova@ubuntu:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
aoivanova@ubuntu:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
aoivanova@ubuntu:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Иванова Ангелина
aoivanova@ubuntu:~/work/arch-pc/lab05$
```

Рисунок 2.6: Создание исполняемого файла и его запуск

2.1.2 Подключение внешнего файла in_out.asm

Скачали файл in_out.asm со страницы курса в ТУИС. Поскольку подключаемый файл in_out.asm должен лежать в том же каталоге, что и файл с программой, в которой он используется, нам нужно его переместить. В одной из панелей mc открыли каталог с файлом lab5-1.asm. В другой панели каталог со скаченным файлом in_out.asm. Скопировали файл in_out.asm в каталог с файлом lab5-1.asm с помощью функциональной клавиши F5.

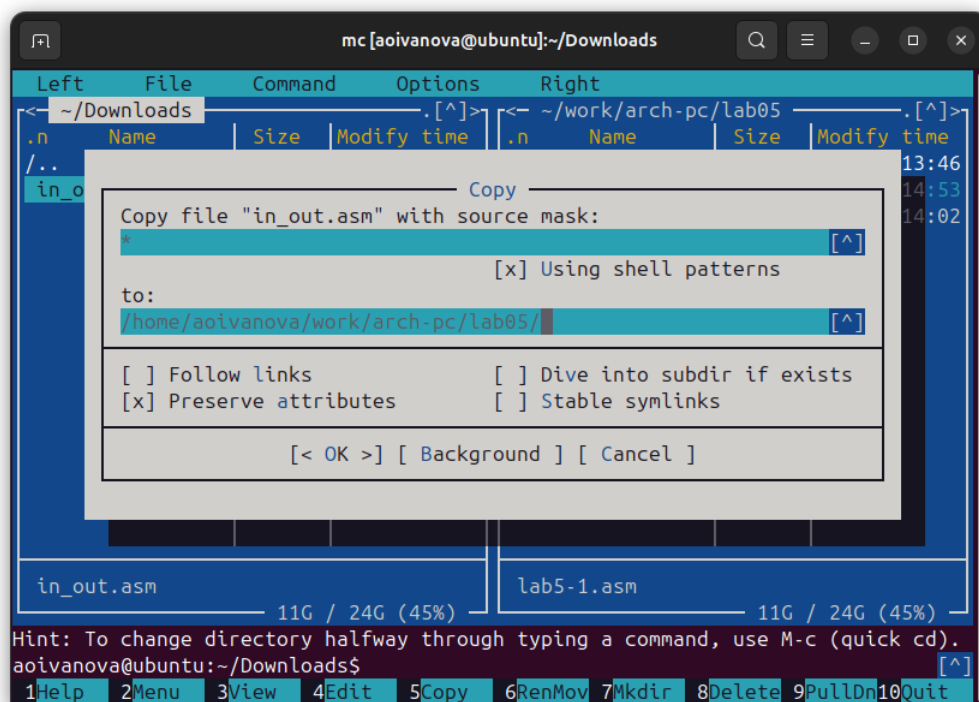


Рисунок 2.7: Копирование файла in_out.asm в каталог с файлом lab5-1.asm

С помощью функциональной клавиши F6 создали копию файла lab5-1.asm с именем lab5-2.asm

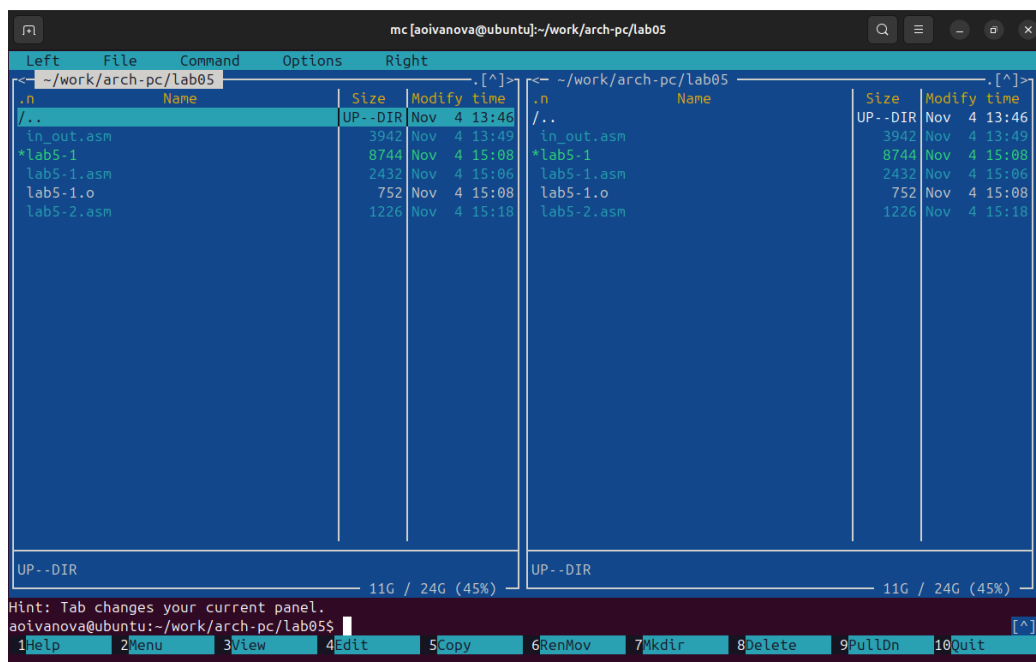
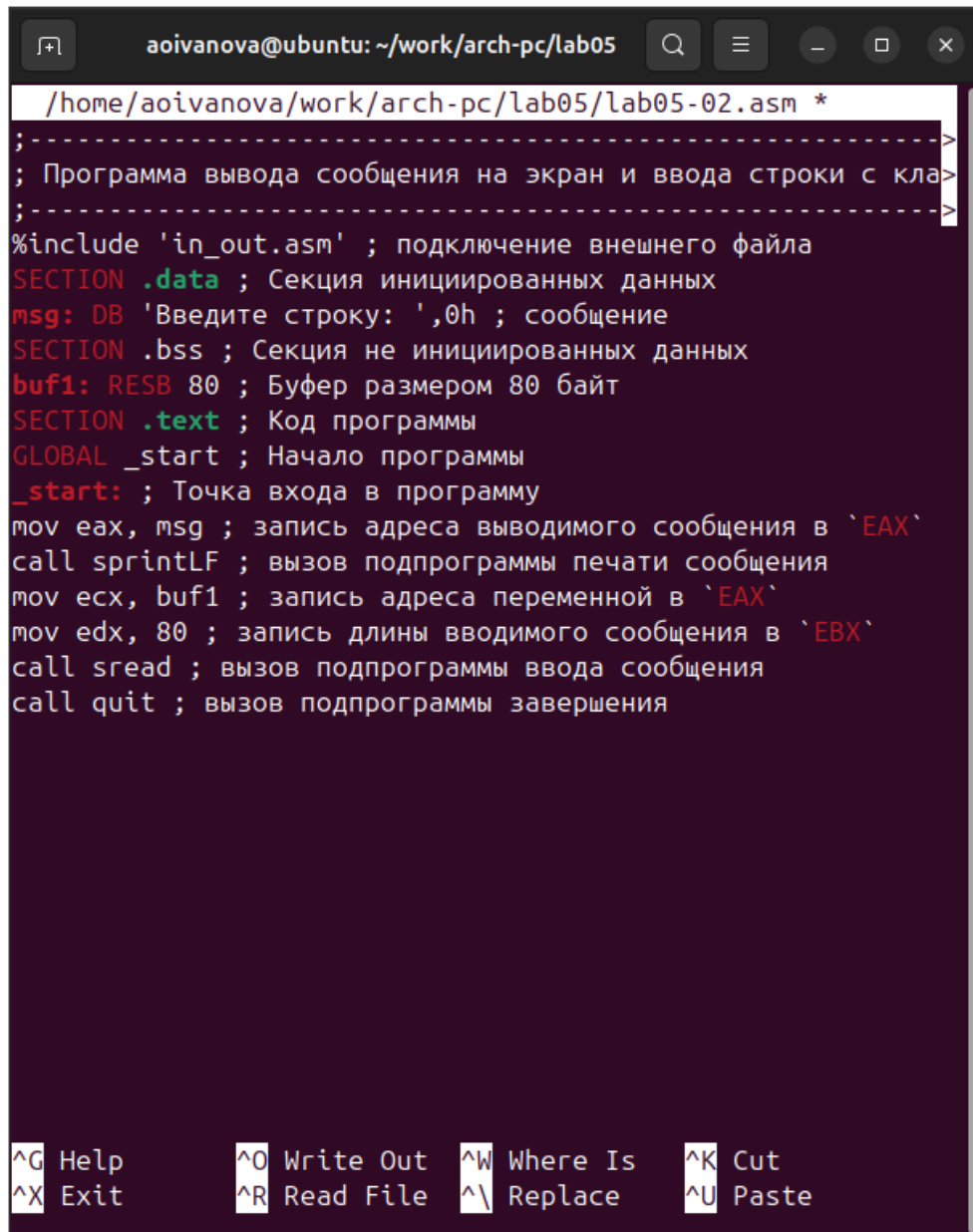


Рисунок 2.8: Созданный файл lab5-2.asm

Изменили текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in_out.asm в соответствии с вторым листингом.



```
aoivanova@ubuntu: ~/work/arch-pc/lab05
/home/aoivanova/work/arch-pc/lab05/lab05-02.asm *
;----->
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;----->
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку: ',0h ; сообщение
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
mov eax, msg ; запись адреса выводимого сообщения в `EAX`
call sprintf ; 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

^G Help      ^O Write Out  ^W Where Is   ^K Cut
^X Exit      ^R Read File  ^\ Replace    ^U Paste
```

Рисунок 2.9: Отредактированный файл lab5-2.asm

Создали исполняемый файл и проверили его работу.

```
aoivanova@ubuntu:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
aoivanova@ubuntu:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
aoivanova@ubuntu:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Иванова Ангелина
aoivanova@ubuntu:~/work/arch-pc/lab05$
```

Рисунок 2.10: Создание исполняемого файла и его запуск

В файле lab5-2.asm заменили подпрограмму sprintLF на sprint. Создали исполняемый файл и проверьте его работу.

```
aoivanova@ubuntu:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
aoivanova@ubuntu:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
aoivanova@ubuntu:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Иванова Агелина
aoivanova@ubuntu:~/work/arch-pc/lab05$
```

Рисунок 2.11: Создание исполняемого файла с изменениями и его запуск

Как можно заметить при использовании команды sprint ввод начинается на той же строке, что и производился вывод, то есть не производится переход на новую строку.

2.2 Задание для самостоятельной работы

Создали копии файлов lab5-1.asm, lab5-2.asm и in_out.asm в папку для 5 лабораторной работы

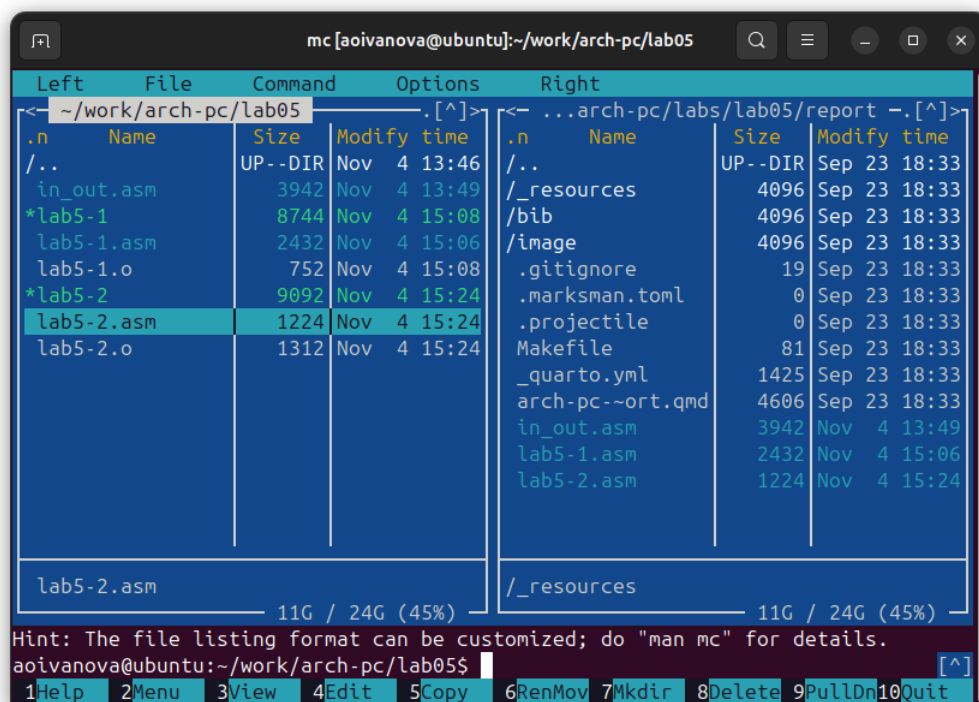


Рисунок 2.12: Копирование файлов

Внесли изменения в файле lab5-1.asm без использования 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.

Листинг 1

```
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
;----- Объявление переменных -----
SECTION .data ; Секция инициированных данных
msg: DB „Введите строку:“,10 ; сообщение плюс
; символ перевода строки
```

```

msgLen: EQU $-msg ; Длина переменной „msg“
SECTION .bss ; Секция не инициированных 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
    mov esi,eax
    mov eax, 4
mov ebx, 1
mov ecx, buf1
mov edx, esi
int 80h

```

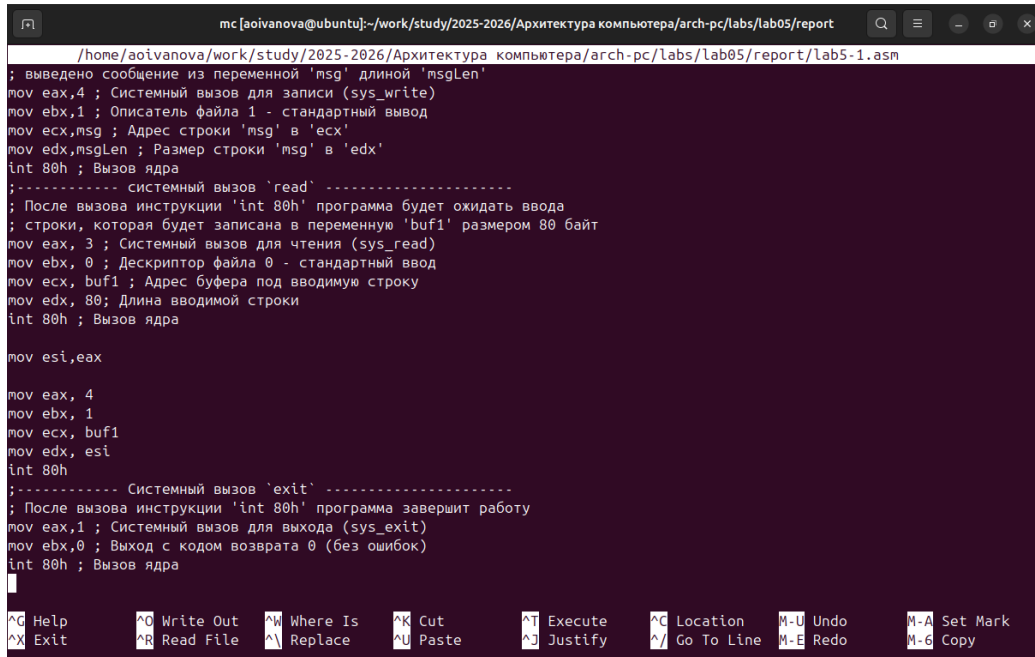
;----- Системный вызов `exit` -----

; 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



```
mc [aoivanova@ubuntu]:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab05/report
/home/aoivanova/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab05/report/lab5-1.asm
; 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

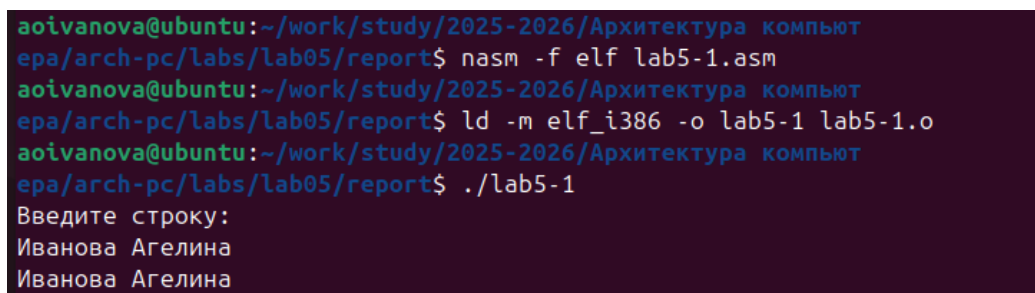
mov esi,eax

mov eax,4
mov ebx,1
mov ecx,buf1
mov edx,esi
int 80h
;----- 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

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark
^X Exit      ^R Read File  ^A Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo      M-G Copy
```

Рисунок 2.13: Изменение файла lab5-1.asm

Создали исполняемый файл и проверили его работу. На приглашение ввести строку ввели свои имя и фамилию, которые потом программа успешно вывела.



```
aoivanova@ubuntu:~/work/study/2025-2026/Архитектура компьют
ера/arch-pc/labs/lab05/report$ nasm -f elf lab5-1.asm
aoivanova@ubuntu:~/work/study/2025-2026/Архитектура компьют
ера/arch-pc/labs/lab05/report$ ld -m elf_i386 -o lab5-1 lab5-1.o
aoivanova@ubuntu:~/work/study/2025-2026/Архитектура компьют
ера/arch-pc/labs/lab05/report$ ./lab5-1
Введите строку:
Иванова Агелина
Иванова Агелина
```

Рисунок 2.14: Создание исполняемого файла и его запуск

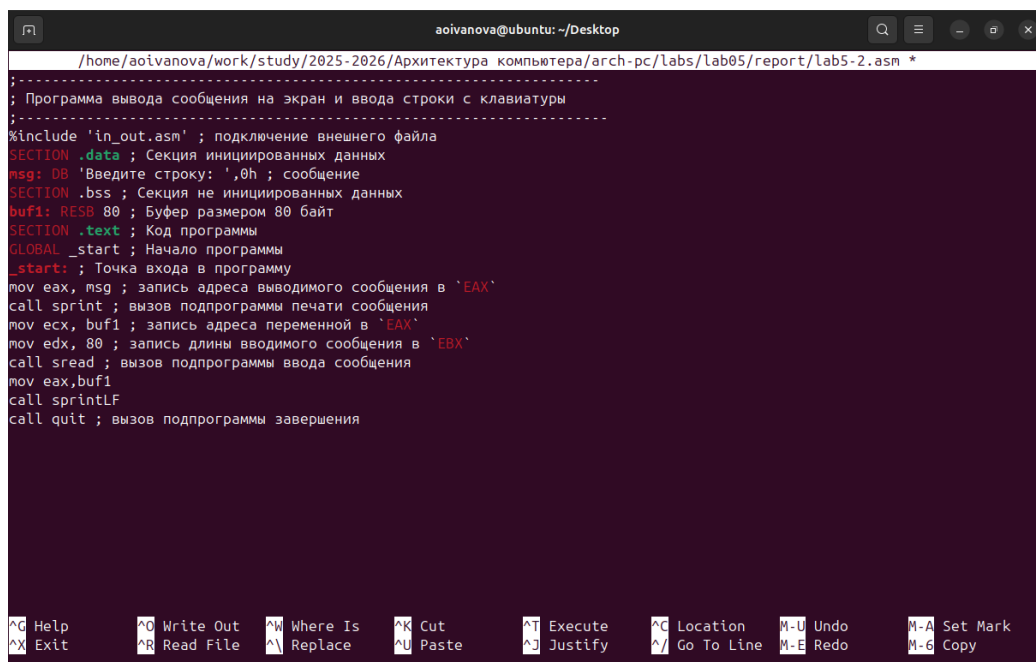
Внесли изменения в файле lab5-2.asm с использованием подпрограмм из внешнего

файла in_out.asm, так чтобы она работала по следующему алгоритму:

- вывести приглашение типа «Введите строку:»;
- ввести строку с клавиатуры;
- вывести введенную строку на экран.

Листинг 2

```
;-----  
; 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  
mov eax,buf1  
call sprintLF  
call quit ; вызов подпрограммы завершения
```

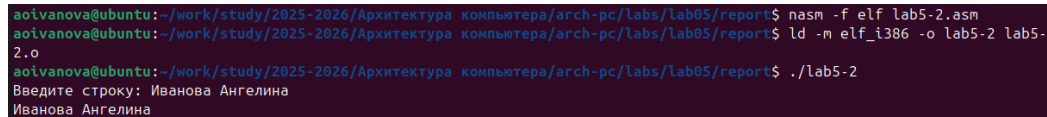


```
aoivanova@ubuntu: ~/Desktop
/home/aoivanova/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab05/report/lab5-2.asm *
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;
-----
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не иницированных 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
mov eax, buf1
call sprintf
call quit ; вызов подпрограммы завершения

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute   ^C Location  ^U Undo      ^A Set Mark
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify   ^_ Go To Line ^E Redo      ^M Copy
```

Рисунок 2.15: Изменение файла lab5-2.asm

Создали исполняемый файл и проверили его работу. На приглашение ввести строку ввели свои имя и фамилию, которые потом программа успешно вывела.



```
aoivanova@ubuntu:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab05/report$ nasm -f elf lab5-2.asm
aoivanova@ubuntu:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab05/report$ ld -m elf_i386 -o lab5-2 lab5-2.o
aoivanova@ubuntu:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab05/report$ ./lab5-2
Введите строку: Иванова Ангелина
```

Рисунок 2.16: Создание исполняемого файла и его запуск

3 Выводы

Приобрели практические навыки работы в Midnight Commander, а также освоили инструкции языка ассемблера `mov` и `int`