

# **Отчет по лабораторной работе № 4**

**Дисциплина: архитектура компьютеров**

Казазаев Даниил Михайлович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задания Лабораторной работы</b>	<b>5</b>
<b>3</b>	<b>Задания Самостоятельной работы</b>	<b>6</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>5</b>	<b>Выполнение самостоятельной работы</b>	<b>15</b>
5.1	Задание 1 . . . . .	15
5.2	Задание 3 . . . . .	18
<b>6</b>	<b>Вывод</b>	<b>21</b>

## Список иллюстраций

4.1	Установка Midnight Commander . . . . .	7
4.2	Открытый Midnight Commander . . . . .	7
4.3	Переход в нужный каталог . . . . .	8
4.4	Созданный файл lab5-1.asm . . . . .	8
4.5	Редактирование файла . . . . .	9
4.6	Проверка . . . . .	9
4.7	Трансляция файла . . . . .	10
4.8	Компоновка файла . . . . .	10
4.9	Исполнение файла . . . . .	10
4.10	Копирование файла in_out.asm . . . . .	11
4.11	Копирование файла lab5-1.asm . . . . .	12
4.12	Редактирование файла . . . . .	12
4.13	Трансляция и компоновка файла lab5-2.asm . . . . .	13
4.14	Исполнение файла . . . . .	13
4.15	Редактирование файла . . . . .	13
4.16	Трансляция и компоновка файла lab5-2.asm с последующим его запуском . . . . .	14
5.1	Копирование файла lab5-1.asm . . . . .	15
5.2	Редактирование файла lab5-1-1.asm . . . . .	16
5.3	Трансляция и компоновка файла lab5-1-1.asm с последующим его запуском . . . . .	16
5.4	Копирование файла lab5-2.asm . . . . .	18
5.5	Редактирование файла lab5-2-1.asm . . . . .	18
5.6	Трансляция и компоновка файла lab5-2-1.asm с последующим его запуском . . . . .	19

# 1 Цель работы

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

## 2 Задания Лабораторной работы

1. Открыть Midnight Commander.
2. Перейти в каталог для выполнения задания.
3. Создать файл lab5-1.asm.
4. Отредактировать файл lab5-1.asm.
5. Удостовериться в том, что файл соержит текст программы.
6. Трансировать файл lab5-1.asm в объектный файл.
7. Запустить файл.
8. Скачать файл in\_out.asm с курса в ТУИС.
9. Копировать файл in\_out.asm в необходимый каталог.
10. Скопировать файл lab5-1.asm в необходимый каталог с названием lab5-2.asm
11. Отредактировать файл lab5-2.asm с использованием подпрограмм из in\_out.asm.
12. Заменить sprintLF на sprint и объяснить, что изменилось

### 3 Задания Самостоятельной работы

1. В каталоге `~/work/arch-рс/lab04` с помощью команды `ср` создайте копию файла `hello.asm` с именем `lab4.asm`
2. С помощью любого текстового редактора внесите изменения в текст программы в файле `lab4.asm` так, чтобы вместо `Hello world!` на экран выводилась строка с вашими фамилией и именем.
3. Оттранслируйте полученный текст программы `lab4.asm` в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл.
4. Скопируйте файлы `hello.asm` и `lab4.asm` в Ваш локальный репозиторий в каталог `~/work/study/2023-2024/“Архитектура компьютера”/arch-рс/labs/lab04/`. Загрузите файлы на Github.

# 4 Выполнение лабораторной работы

Устанавливаю Midnight Commander. (рис. [4.1])

```
dmkazazaev@Ubuntu:~$ sudo apt install mc
[sudo] password for dmkazazaev:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libssh2-1 mc-data
Suggested packages:
  arj dbview djvulibre-bin epub-utils gv libaspell-dev links | w3m | lynx
  odt2txt python python-boto python-tzunar wimtools
The following NEW packages will be installed:
  libssh2-1 mc mc-data
0 upgraded, 3 newly installed, 0 to remove and 40 not upgraded.
Need to get 2 084 kB of archives.
After this operation, 8 209 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ru.archive.ubuntu.com/ubuntu jammy/universe amd64 libssh2-1 amd64 1
.10.0-3 [109 kB]
Get:2 http://ru.archive.ubuntu.com/ubuntu jammy/universe amd64 mc-data all 3:4.8
.27-1 [1 427 kB]
Get:3 http://ru.archive.ubuntu.com/ubuntu jammy/universe amd64 mc amd64 3:4.8.27
```

Рис. 4.1: Установка Midnight Commander

Открываю Midnight Commander. (рис. [4.2])

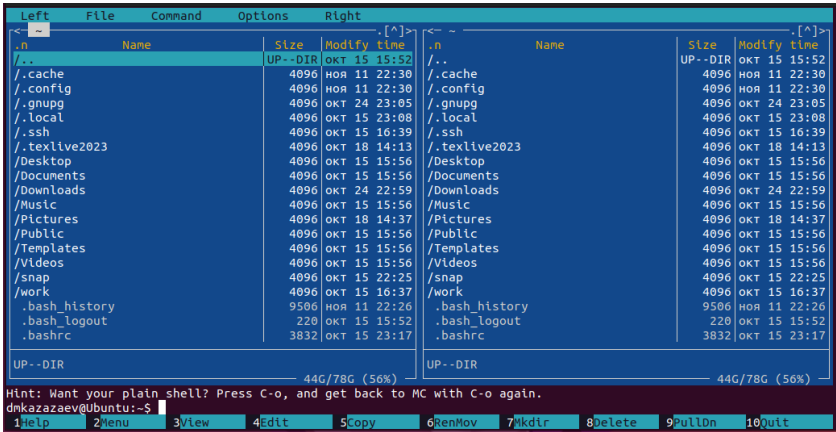


Рис. 4.2: Открытый Midnight Commander

Перехожу в нужный каталог. (рис. [4.3])

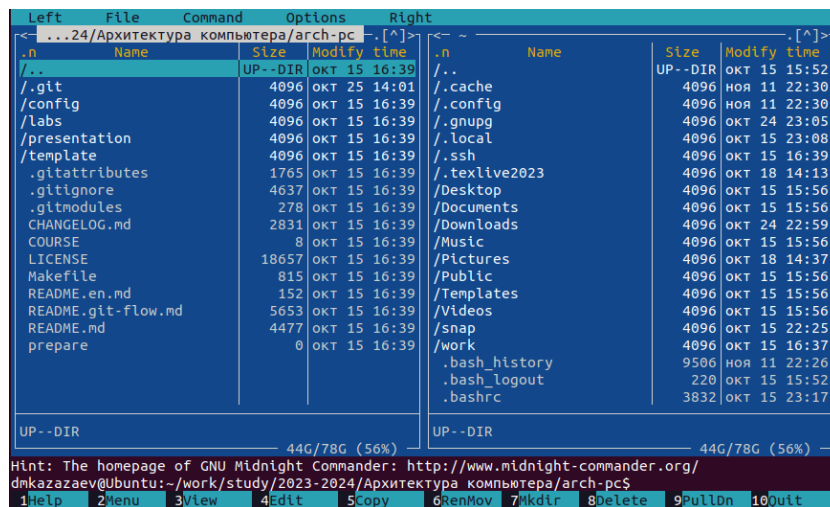


Рис. 4.3: Переход в нужный каталог

Создаю файл lab5-1.asm через Midnight Commander. (рис. [4.4])

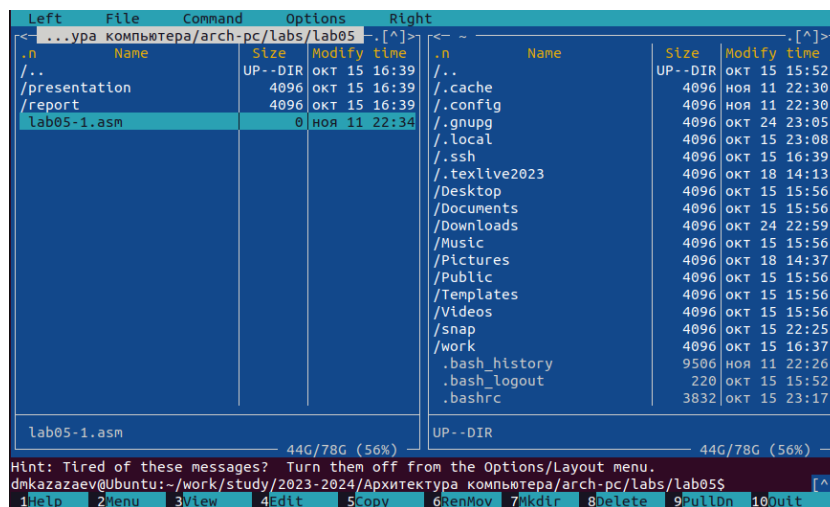


Рис. 4.4: Созданный файл lab5-1.asm

Редактирую файл lab5-1.asm. (рис. [4.5])



```

;----- Объявление переменных -----
SECTION .data                                ; Секция иницированных данных
msg: DB 'Введите строку:',10                ; сообщение плюс
                                           ; символ перевода строки
msgLen: EQU $-msg                           ; Длина переменной 'msg'
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

;----- 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                                  ;Descriptor 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

```

Рис. 4.5: Редактирование файла

Проверяю на наличие текста программы в файле lab5-1.asm. (рис. [4.6])

```

SECTION .data                                ; Секция иницированных данных
msg: DB 'Введите строку:',10                ; сообщение плюс
                                           ; символ перевода строки
msgLen: EQU $-msg                           ; Длина переменной 'msg'
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

;----- 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                                  ;Descriptor 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

```

Рис. 4.6: Проверка

Транслирую файл lab5-1.asm в объектный файл. (рис. [4.7])

```
dmkazazaev@Ubuntu: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05/report$ nasm -o obj.o -f elf -g -l list.lst lab05-1.o
```

Рис. 4.7: Трансляция файла

Компонуя файл lab5-1.asm в исполняемый файл. (рис. [4.8])

```
dmkazazaev@Ubuntu: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05/report$ ld -m elf_i386 lab05-1.o -o lab05-1
dmkazazaev@Ubuntu: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05/report$ ls
bib image lab05-1 lab05-1.asm lab05-1.o Makefile pandoc report.md
```

Рис. 4.8: Компановка файла

Запускаю программу. (рис. [4.9])

```
dmkazazaev@Ubuntu: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05/report$ ./lab05-1
Введите строку:
Каззаев Даниил Михайлович
```

Рис. 4.9: Исполнение файла

Копирую файл in\_out.asm в нужный каталог с помощью Midnight Commander.  
(рис. [4.10])

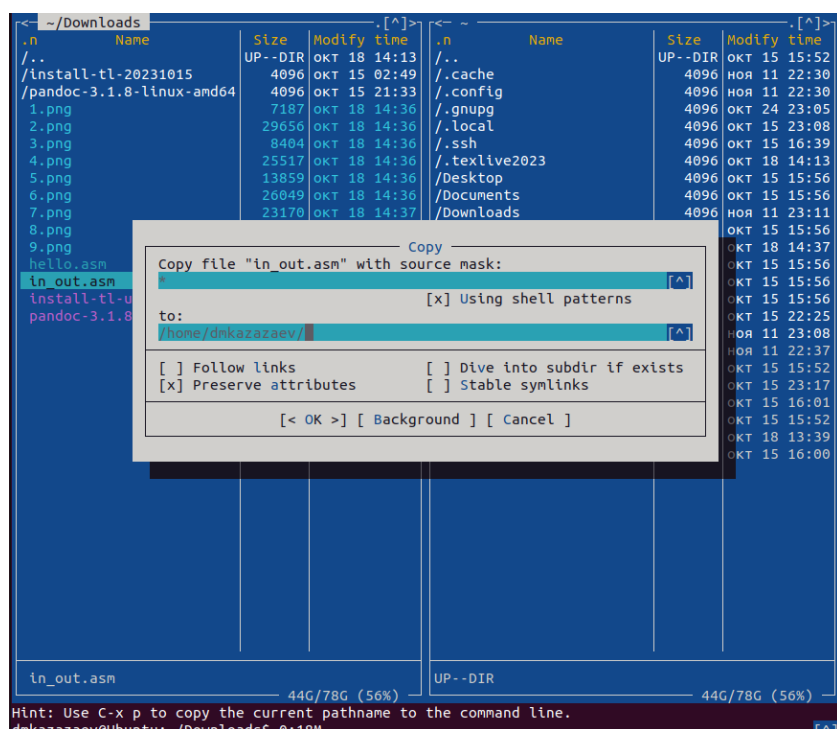


Рис. 4.10: Копирование файла in\_out.asm

Копирую файл lab5-1.asm с названием lab5-2.asm. (рис. [4.11])

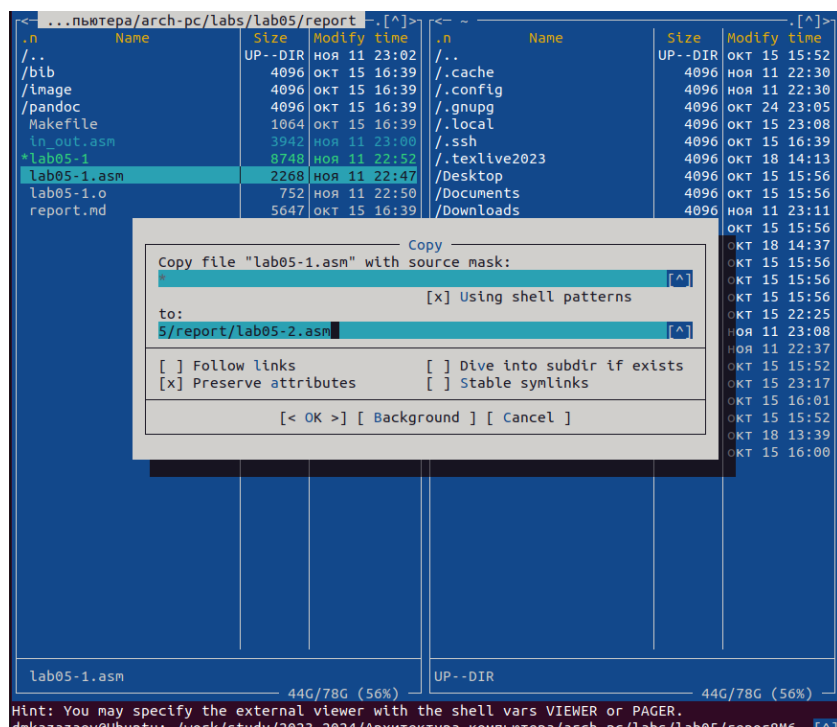


Рис. 4.11: Копирование файла lab5-1.asm

Редактирую файл lab5-2.asm с использованием подпрограмм in\_out.asm. (рис. [4.12])

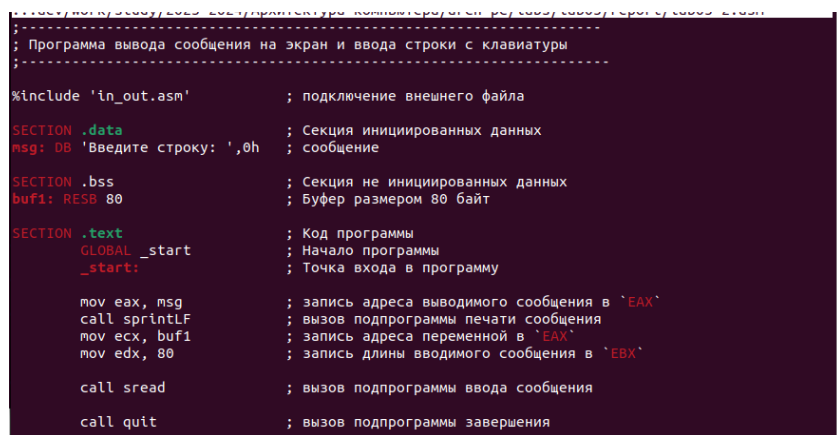


Рис. 4.12: Редактирование файла

Транслирую и компоную файл lab5-2.asm. (рис. [4.13])

```
dmkazazaev@Ubuntu: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05/report$ nasm -f elf lab05-2.asm
dmkazazaev@Ubuntu: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05/report$ ld -m elf_i386 lab05-2.o -o lab05-2
```

Рис. 4.13: Трансляция и компоновка файла lab5-2.asm

Запускаю исполняемый файл lab5-2. (рис. [4.14])

```
dmkazazaev@Ubuntu: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05/report$ ./lab05-2
Введите строку:
Казазаев Даниил Михайлович
```

Рис. 4.14: Исполнение файла

Меняю sprintf на sprintf в lab5-2.asm. (рис. [4.15])

```
...aev/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05/report/lab05-2.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
buf: 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, buf                ; запись адреса переменной в 'ECX'
    mov edx, 80                 ; запись длины вводимого сообщения в 'EDX'

    call sread                  ; вызов подпрограммы ввода сообщения

    call quit                   ; вызов подпрограммы завершения
```

Рис. 4.15: Редактирование файла

Еще раз транслирую и компоную файл lab5-2.asm после чего запускаю его. (рис. [4.16])

```

dnkazazaev@Ubuntu: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab
05/report$ nasm -f elf lab05-2.asm
dnkazazaev@Ubuntu: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab
05/report$ ld -m elf_i386 lab05-2.o -o lab05-2
dnkazazaev@Ubuntu: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab
05/report$ ls
bib  in_out.asm  lab05-1.asm  lab05-2      lab05-2.o  pandoc
image lab05-1    lab05-1.o   lab05-2.asm  Makefile   report.md
dnkazazaev@Ubuntu: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab
05/report$ ./lab05-2
Введите строку: Казаазев Даниил Михайлович

```

Рис. 4.16: Трансляция и компоновка файла lab5-2.asm с последующим его запуском

Отличие заключается в том, что с использованием `sprintf` ввод происходит с новой строки, а при использовании `sprint` ввод продолжается на той же строке, на которой находится надпись, выведенная программой

## 5 Выполнение самостоятельной работы

### 5.1 Задание 1

Копирую файл lab5-1.asm с названием lab5-1-1.asm. (рис. [5.1])

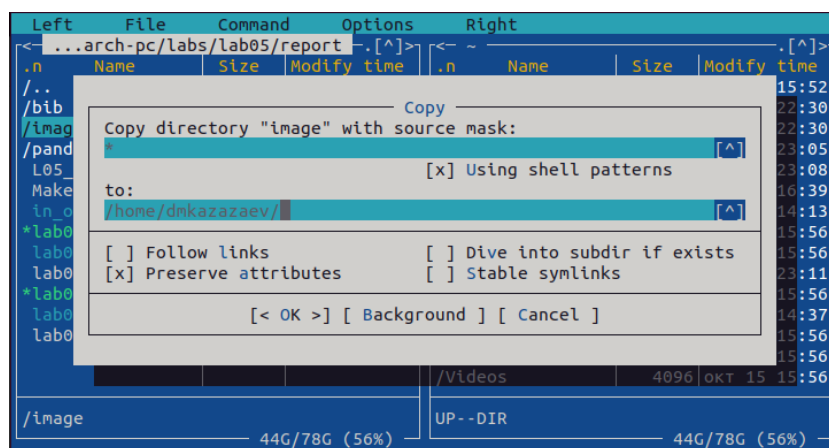


Рис. 5.1: Копирование файла lab5-1.asm

Редактирую файл lab5-1-1.asm. (рис. [5.2])

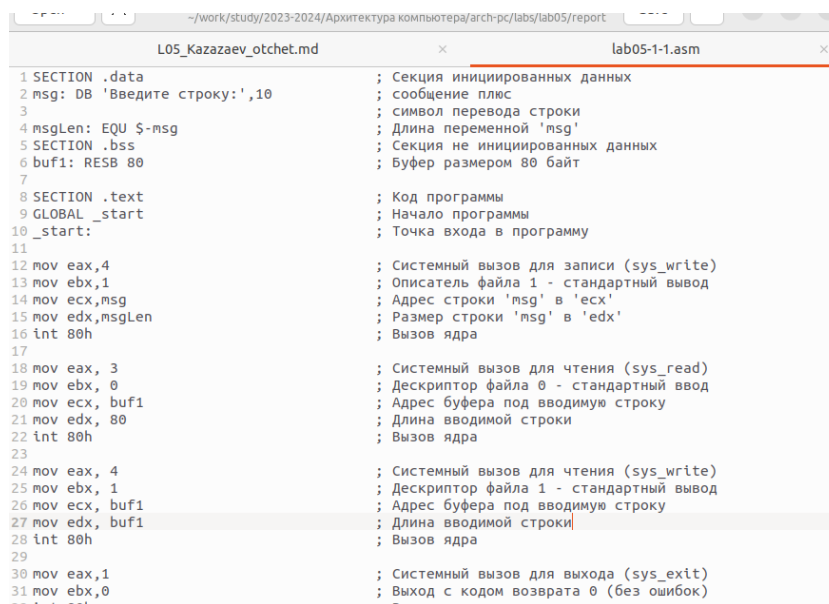


Рис. 5.2: Редактирование файла lab5-1-1.asm

Транслирую и компоную файл lab5-1-1.asm после чего запускаю его. (рис. [5.3])

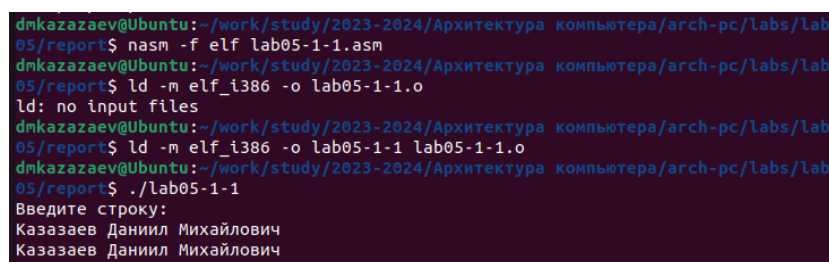


Рис. 5.3: Трансляция и компоновка файла lab5-1-1.asm с последующим его запуском

### Листинг задания 1

```

SECTION .data                ; Секция иницированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
                               ; символ перевода строки

msgLen: EQU $-msg            ; Длина переменной 'msg'
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,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

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 eax, 4                   ; Системный вызов для чтения (sys_write)
mov ebx, 1                   ; Дескриптор файла 1 - стандартный вывод
mov ecx, buf1                ; Адрес буфера под вводимую строку
mov edx, buf1                ; Длина вводимой строки
int 80h                      ; Вызов ядра

mov eax,1                    ; Системный вызов для выхода (sys_exit)
mov ebx,0                    ; Выход с кодом возврата 0 (без ошибок)
int 80h                      ; Вызов ядра

```

## 5.2 Задание 3

Копирую файл lab5-2.asm с названием lab5-2-1.asm. (рис. [5.4])

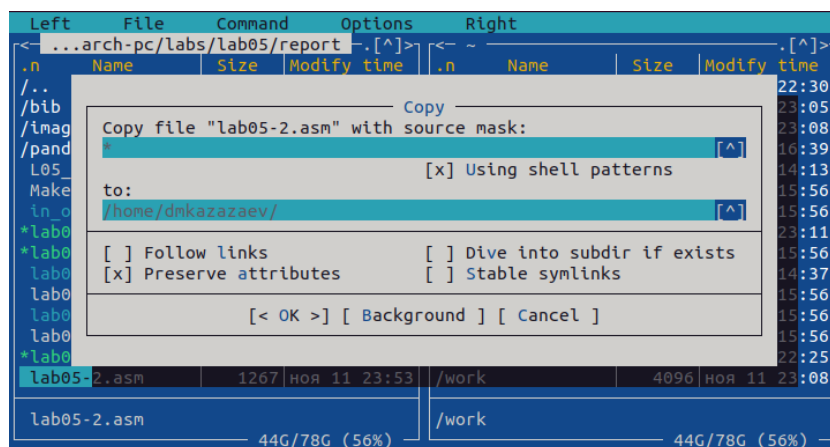


Рис. 5.4: Копирование файла lab5-2.asm

Редактирую файл lab5-2-1.asm. (рис. [5.5])

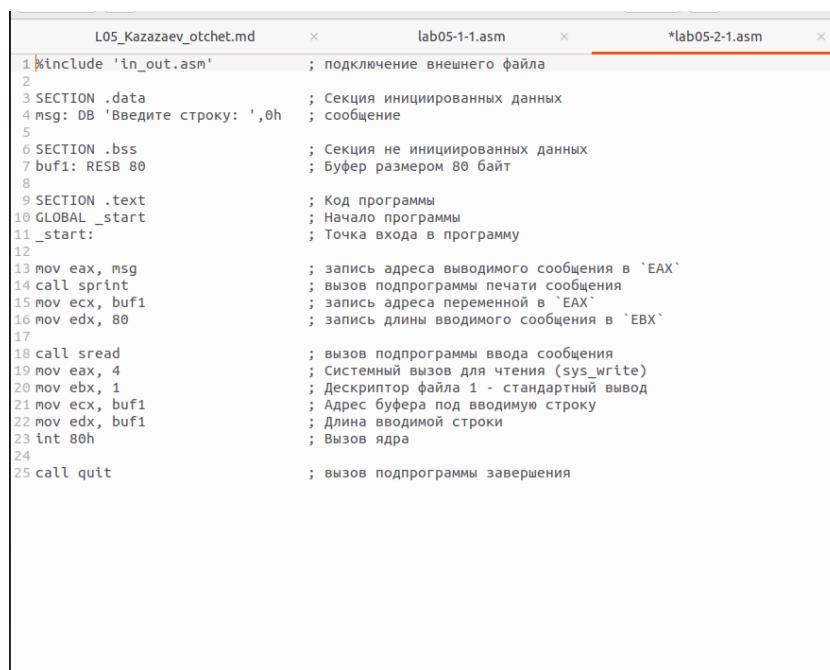


Рис. 5.5: Редактирование файла lab5-2-1.asm

Транслирую и компоную файл lab5-2-1.asm после чего запускаю его. (рис. [5.6])

```

dmkazazaev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab
05/report$ nasm -f elf lab05-2-1.asm
dmkazazaev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab
05/report$ ld -m elf_i386 -o lab05-2-1 lab05-2-1.o
dmkazazaev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab
05/report$ ./lab05-2-1
Введите строку: Казазаев Даниил Михайлович
Казазаев Даниил Михайлович
dmkazazaev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab
05/report$

```

Рис. 5.6: Трансляция и компоновка файла lab5-2-1.asm с последующим его запуском

### Листинг задания 3

```

%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, 4                     ; Системный вызов для чтения (sys_write)
mov ebx, 1                     ; Дескриптор файла 1 - стандартный вывод
mov ecx, buf1                  ; Адрес буфера под вводимую строку

```

```
mov edx, buf1          ; Длина вводимой строки
int 80h                ; Вызов ядра

call quit              ; вызов подпрограммы завершения
```

## 6 Вывод

При выполнении лабораторной работы я приобрел практические навыки работы с Midnight Commander, и освоил инструкции языка ассемблера `mov` и `int`.