

# **Отчёт по лабораторной работе №5**

**Дисциплина: архитектура компьютера**

Сидельников Андрей Владимирович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
2.1	Подключение внешнего файла in_out.asm . . . . .	8
<b>3</b>	<b>Выполнение заданий для самостоятельной работы</b>	<b>12</b>
<b>4</b>	<b>Выводы</b>	<b>16</b>

## **Список таблиц**

# Список иллюстраций

2.1	Открытый mc	6
2.2	Создание каталога	6
2.2	Переход в каталог lab05	7
2.3	Создание файла	7
2.4	Открытие файла для редактирования	7
2.5	Редактирование файла	7
2.6	Открытие файла для просмотра	8
2.7	Компиляция файла и передача на обработку компоновщику	8
2.8	Исполнение файла	8
2.9	Скачанный файл	8
2.10	Копирование файла	9
2.11	Копирование файла	9
2.12	Редактирование файла	10
2.13	Исполнение файла	10
2.14	Отредактированный файл	10
2.15	Исполнение файла	11
3.1	Копирование файла	12
3.2	Редактирование файла	13
3.3	Исполнение файла	13
3.4	Копирование файла	14
3.5	Редактирование файла	14
3.6	Исполнение файла	15

# 1 Цель работы

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

## 2 Выполнение лабораторной работы

Открываю Midnight Commander, введя в терминал mc (рис. 2.1).

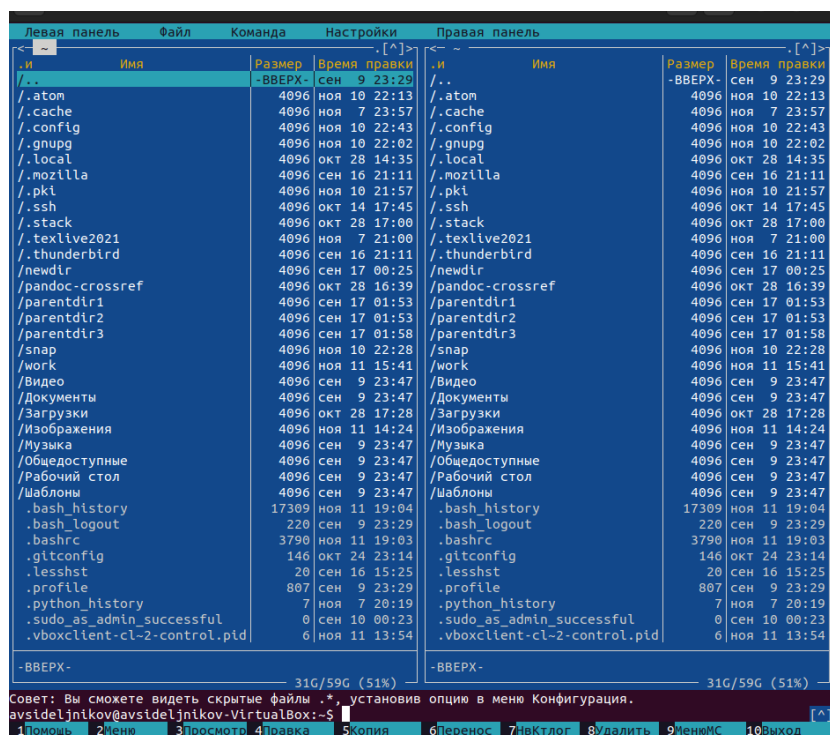
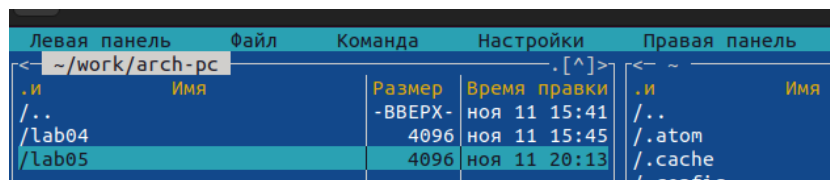


Рис. 2.1: Открытый mc

Перехожу в каталог ~/work/arch-pc созданный при выполнении лабораторной работы №4 и создаю там каталог lab05 с помощью клавиши F7 (рис. 2.2).



Создание каталога

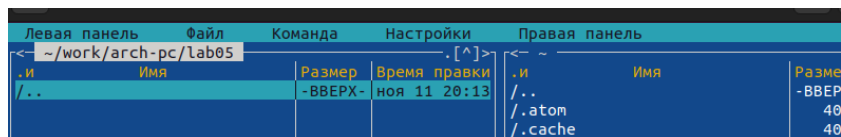


Рис. 2.2: Переход в каталог lab05

В строке ввода прописываю команду `touch lab5-1.asm`, чтобы создать файл, в котором буду работать (рис. 2.3).

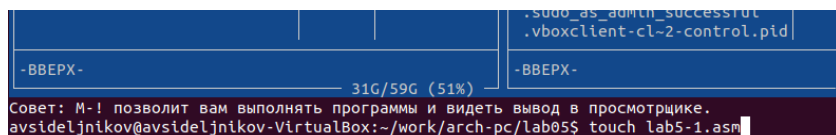


Рис. 2.3: Создание файла

С помощью функциональной клавиши F4 открываю созданный файл для редактирования в редакторе nano (рис. 2.4).

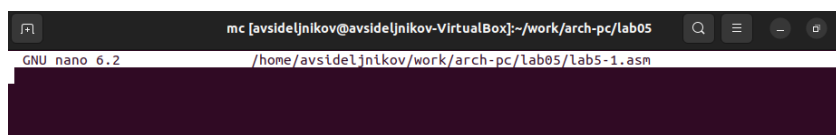


Рис. 2.4: Открытие файла для редактирования

Ввожу в файл код программы для запроса строки у пользователя (рис. 2.5).

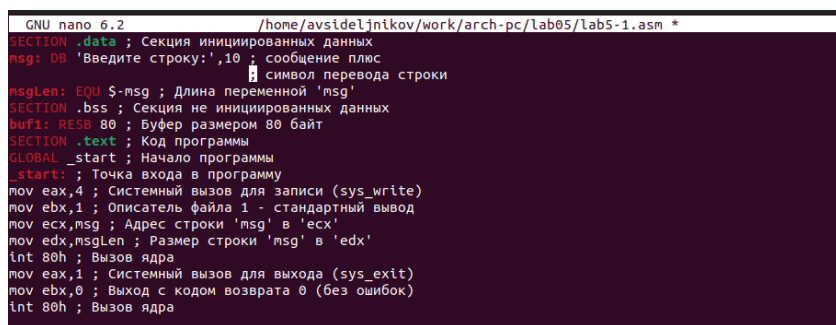


Рис. 2.5: Редактирование файла

С помощью функциональной клавиши F3 открываю файл для просмотра, чтобы проверить, содержит ли файл текст программы (рис. 2.6).

```

/home/avsideljnikov/work/arch-pc/lab05/lab5-1.asm 1833/1833 100%
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 записи (sys_write)
mov ebx,1 ; Описатель файла 1 - стандартный вывод
mov ecx,buf1 ; Адрес буфера в 'ecx'
int 80h ; Вызов ядра

```

Рис. 2.6: Открытие файла для просмотра

Транслирую текст программы файла в объектный файл командой `nasm -f elf lab5-1.asm`. Создался объектный файл `lab5-1.o`. Выполняю компоновку объектного файла с помощью команды `ld -m elf_i386 -o lab5-1 lab5-1.o` (рис. 2.7).

```

avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o

```

Рис. 2.7: Компиляция файла и передача на обработку компоновщику

Запускаю исполняемый файл и ввожу свои ФИО, на этом программа заканчивает свою работу (рис. 2.8)

```

avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Сидельников Андрей Владимирович

```

Рис. 2.8: Исполнение файла

## 2.1 Подключение внешнего файла `in_out.asm`

Скачиваю файл `in_out.asm` со страницы курса в ТУИС. Он сохранился в каталог “Загрузки” (рис. 2.9)

Левая панель	Файл	Команда	Настройки	Правая панель
<- ~ /Загрузки				<- ~
.и	Имя	Размер	Время правки	.и
./..		-ВВЕРХ-	ноя 11 22:39	./..
in_out.asm		3942	ноя 11 22:43	./atom

Рис. 2.9: Скачанный файл



С помощью функциональной клавиши F5 копирую файл in\_out.asm из каталога Загрузки в созданный каталог lab05 (рис. 2.10).

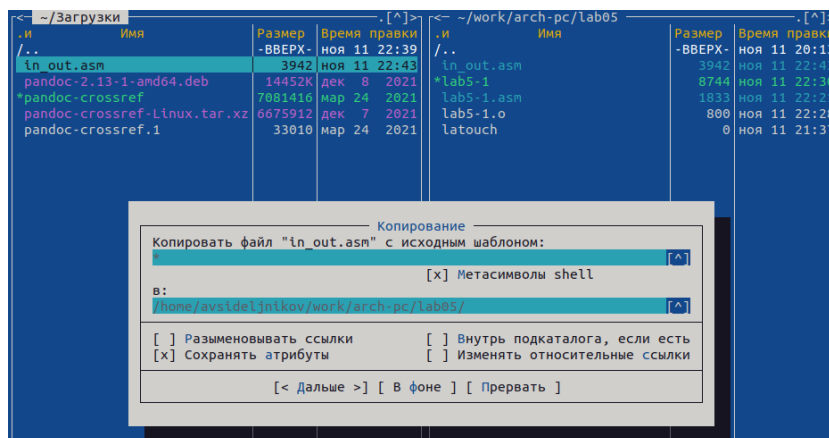


Рис. 2.10: Копирование файла

С помощью функциональной клавиши F5 копирую файл lab5-1 в тот же каталог, но с другим именем, для этого в появившемся окне mc прописываю имя для копии файла (рис. 2.11).

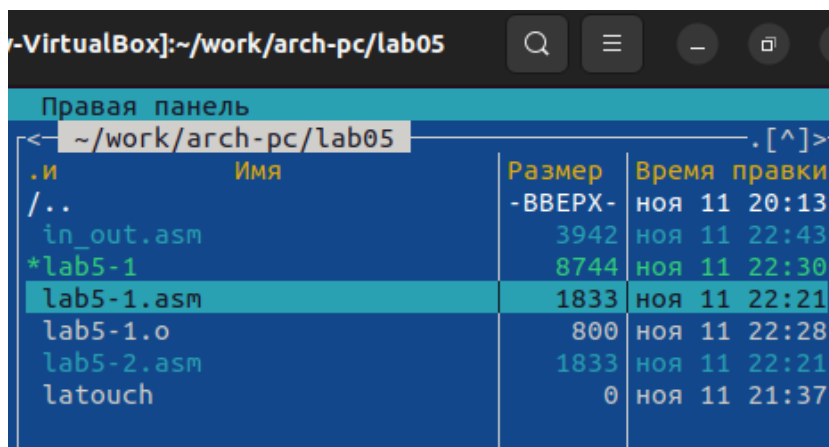


Рис. 2.11: Копирование файла

Изменяю содержимое файла lab5-2.asm во встроенном редакторе nano чтобы в программе использовались подпрограммы из внешнего файла in\_out.asm (рис. 2.12).

```

GNU nano 6.2 /home/avsidel'nikov/work/arch-pc/lab05/lab5-2.asm *
#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 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

```

Рис. 2.12: Редактирование файла

Транслирую текст программы файла в объектный файл командой `nasm -f elf lab5-2.asm`. Создался объектный файл `lab5-2.o`. Выполняю компоновку объектного файла с помощью команды `ld -m elf_i386 -o lab5-2 lab5-2.o` Создался исполняемый файл `lab5-2`. Запускаю исполняемый файл (рис. 2.13).

```

avsidel'nikov@avsidel'nikov-VirtualBox:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
avsidel'nikov@avsidel'nikov-VirtualBox:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
avsidel'nikov@avsidel'nikov-VirtualBox:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Сидельников Андрей Владимирович
avsidel'nikov@avsidel'nikov-VirtualBox:~/work/arch-pc/lab05$

```

Рис. 2.13: Исполнение файла

Открываю файл `lab5-2.asm` для редактирования в `nano` функциональной клавишей F4. Изменяю в нем подпрограмму `sprintf` на `sprint`. Сохраняю изменения и открываю файл для просмотра, чтобы проверить сохранение действий (рис. 2.14).

```

GNU nano 6.2 /home/avsidel'nikov/work/arch-pc/lab05/lab5-2.asm
#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
call quit ; вызов подпрограммы завершения

```

Рис. 2.14: Отредактированный файл

Снова транслирую файл, выполняю компоновку созданного объектного файла, запускаю новый исполняемый файл (рис. 2.15)

```
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2-2 lab-2.o
ld: невозможно найти lab-2.o: Нет такого файла или каталога
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2-2 lab5-2.o
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab05$ ./lab5-2-2
Введите строку: Сидельников Андрей Владимирович
```

Рис. 2.15: Исполнение файла

Разница между первым исполняемым файлом lab5-2 и вторым lab5-2-2 в том, что запуск первого запрашивает ввод с новой строки, а программа, которая выполняется при запуске второго, запрашивает ввод без переноса на новую строку.

### 3 Выполнение заданий для самостоятельной работы

Создаю копию файла lab5-1.asm с именем lab5-1-1.asm с помощью функциональной клавиши F5 (рис. 3.1).

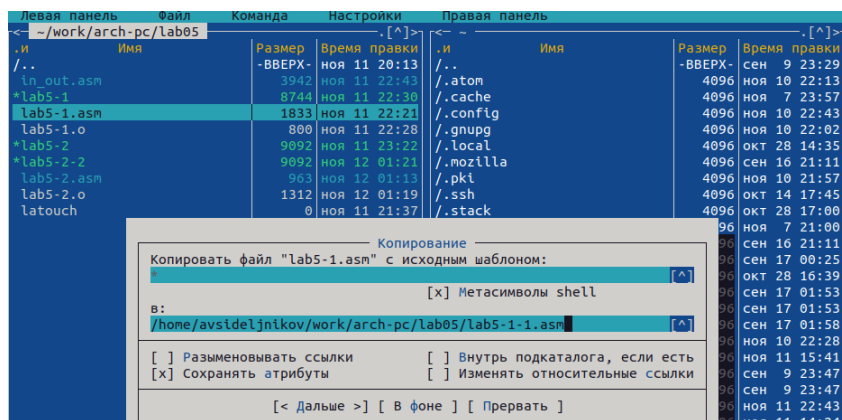


Рис. 3.1: Копирование файла

С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Изменяю программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку (рис. 3.2).

```

GNU nano 6.2 avsideljnikov@avsideljnikov /home/avsideljnikov
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 записи (sys_write)
mov ebx,1 ; Описатель файла '1' - стандартный вывод
mov ecx,buf1 ; Адрес строки buf1 в ecx
mov edx,buf1 ; Размер строки buf1
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

; Добавляем задержку в 2 секунды
mov eax, 60 ; Системный вызов для задержки (sys_nanosleep)
mov ebx, 0 ; Код часов (CLOCK_REALTIME)
mov ecx, 2 ; Секунды
mov edx, 0 ; Наносекунды
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

```

Рис. 3.2: Редактирование файла

Создаю объектный файл lab5-1-1.o, отдаю его на обработку компоновщику, получаю исполняемый файл lab5-1-1, запускаю полученный исполняемый файл. Программа запрашивает ввод, ввожу свои ФИО, далее программа выводит введенные мною данные (рис. 3.3).

```

avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab05$ nasm -f elf lab5-1-1.asm
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1-1 lab5-1-1.o
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab05$ ./lab5-1-1
Введите строку:
Сидельников Андрей Владимирович
Сидельников Андрей Владимирович

```

Рис. 3.3: Исполнение файла

SECTION .data ; Секция инициированных данных msg: DB 'Введите строку:',10 msgLen: EQU \$-msg ; Длина переменной 'msg' 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,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 записи (sys\_write) mov ebx,1 ; Описатель файла '1' - стандартный  
вывод mov ecx,buf1 ; Адрес строки buf1 в ecx mov edx,buf1 ; Размер  
строки buf1 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

Создаю копию файла lab5-2.asm с именем lab5-2-1.asm с помощью функцио-  
нальной клавиши F5 (рис. 3.4).

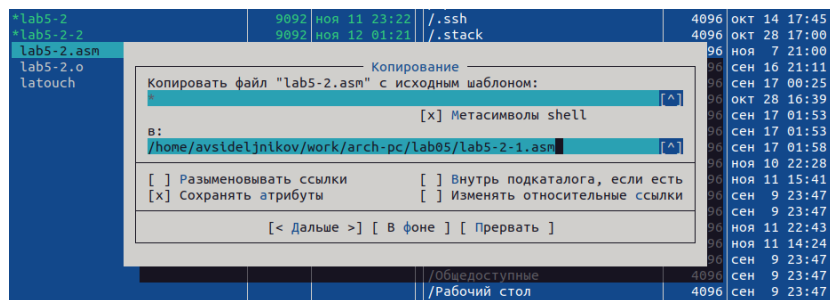


Рис. 3.4: Копирование файла

С помощью функциональной клавиши F4 открываю созданный файл для ре-  
дактирования. Изменяю программу так, чтобы кроме вывода приглашения и  
запроса ввода, она выводила вводимую пользователем строку (рис. 3.5).

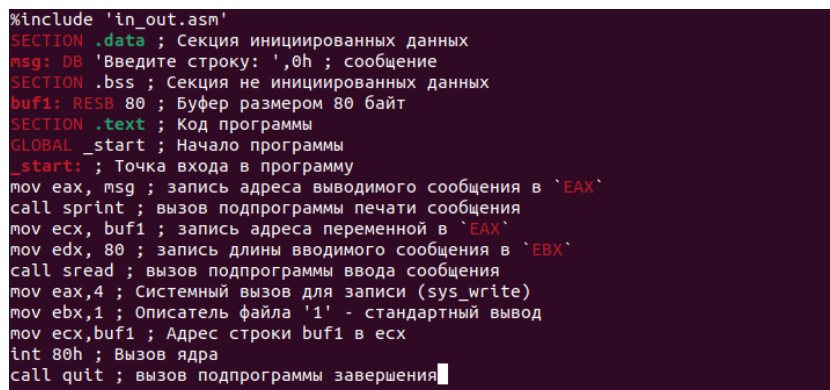
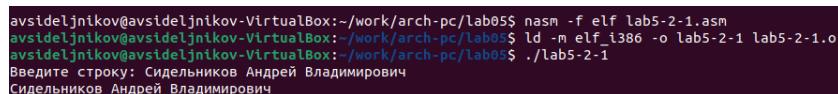


Рис. 3.5: Редактирование файла

Создаю объектный файл lab5-2-1.o, отдаю его на обработку компоновщику,

получаю исполняемый файл lab5-2-1, запускаю полученный исполняемый файл. Программа запрашивает ввод без переноса на новую строку, ввожу свои ФИО, далее программа выводит введенные мною данные (рис. 3.6).



```
avsidel'nikov@avsidel'nikov-VirtualBox:~/work/arch-pc/lab05$ nasm -f elf lab5-2-1.asm
avsidel'nikov@avsidel'nikov-VirtualBox:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2-1 lab5-2-1.o
avsidel'nikov@avsidel'nikov-VirtualBox:~/work/arch-pc/lab05$ ./lab5-2-1
Введите строку: Сидельников Андрей Владимирович
Сидельников Андрей Владимирович
```

Рис. 3.6: Исполнение файла

```
%include 'in_out.asm' 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 записи (sys_write)
mov ebx,1 ; Описатель файла '1' - стандартный вывод mov ecx,buf1
; Адрес строки buf1 в ecx int 80h ; Вызов ядра call quit ; вызов
подпрограммы завершения
```

## 4 Выводы

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