

Лабораторная работа №5

дисциплина: Архитектура компьютера

Савенкова Алиса Евгеньевна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Задания для самостоятельной работы	19
6	Выводы	22
	Список литературы	23

Список иллюстраций

4.1	Открытие mc	9
4.2	mc	9
4.3	Создание нового каталога	10
4.4	Новый каталог	11
4.5	Новый файл	11
4.6	Новый файл	12
4.7	Измененный файл	13
4.8	Измененный файл	13
4.9	Запуск файла	14
4.10	Копирование файла	14
4.11	Копирование файла	15
4.12	Изменение файла	16
4.13	Запуск файла	16
4.14	Изменение файла	17
4.15	Запуск файла	17
5.1	Изменение файла	19
5.2	Запуск файла	20
5.3	Создание копии файла	21
5.4	Запуск файла	21

Список таблиц

1 Цель работы

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

2 Задание

1. Основы работы с mc
2. Подключение внешнего файла in_out.asm
3. Задания для самостоятельной работы

3 Теоретическое введение

Midnight Commander (или просто `mc`) — это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управлению файловой системой, т.е. `mc` является файловым менеджером. Midnight Commander позволяет сделать работу с файлами более удобной и наглядной. Для активации оболочки Midnight Commander достаточно ввести в командной строке `mc` и нажать клавишу `Enter`. В Midnight Commander используются функциональные клавиши `F1 — F10`, к которым привязаны часто выполняемые операции. Программа на языке ассемблера `NASM`, как правило, состоит из трёх секций: секция кода программы (`SECTION .text`), секция инициированных (известных во время компиляции) данных (`SECTION .data`) и секция неинициализированных данных (тех, под которые во время компиляции только отводится память, а значение присваивается в ходе выполнения программы) (`SECTION .bss`). Инструкция языка ассемблера `mov` предназначена для дублирования данных источника в приёмнике. В общем виде эта инструкция записывается в виде: `mov dst,src` Здесь операнд `dst` — приёмник, а `src` — источник. В качестве операнда могут выступать регистры (`register`), ячейки памяти (`memory`) и непосредственные значения (`const`). Простейший диалог с пользователем требует наличия двух функций — вывода текста на экран и ввода текста с клавиатуры. Простейший способ вывести строку на экран — использовать системный вызов `write`. Этот системный вызов имеет номер 4, поэтому перед вызовом инструкции `int` необходимо поместить значение 4 в регистр `eax`. Первым аргументом `write`, помещаемым в регистр `ebx`, задаётся дескриптор файла. Для вывода на экран в качестве дескриптора файла

нужно указать 1 (это означает «стандартный вывод», т. е. вывод на экран). Вторым аргументом задаётся адрес выводимой строки (помещаем его в регистр `ecx`, например, инструкцией `mov ecx, msg`). Строка может иметь любую длину. Последним аргументом (т.е. в регистре `edx`) должна задаваться максимальная длина выводимой строки. Для ввода строки с клавиатуры можно использовать аналогичный системный вызов `read`. Его аргументы – такие же, как у вызова `write`, только для «чтения» с клавиатуры используется файловый дескриптор 0 (стандартный ввод). Системный вызов `exit` является обязательным в конце любой программы на языке ассемблер. Для обозначения конца программы перед вызовом инструкции `int 80h` необходимо поместить в регистр `eax` значение 1, а в регистр `ebx` код завершения 0.

4 Выполнение лабораторной работы

Первым делом открываю Midnight Commander (рис. 4.1).



Рис. 4.1: Открытие ms

Перехожу в каталог ~/work/arch-рс созданный, при выполнении лабораторной работы №4 (рис. 4.2).

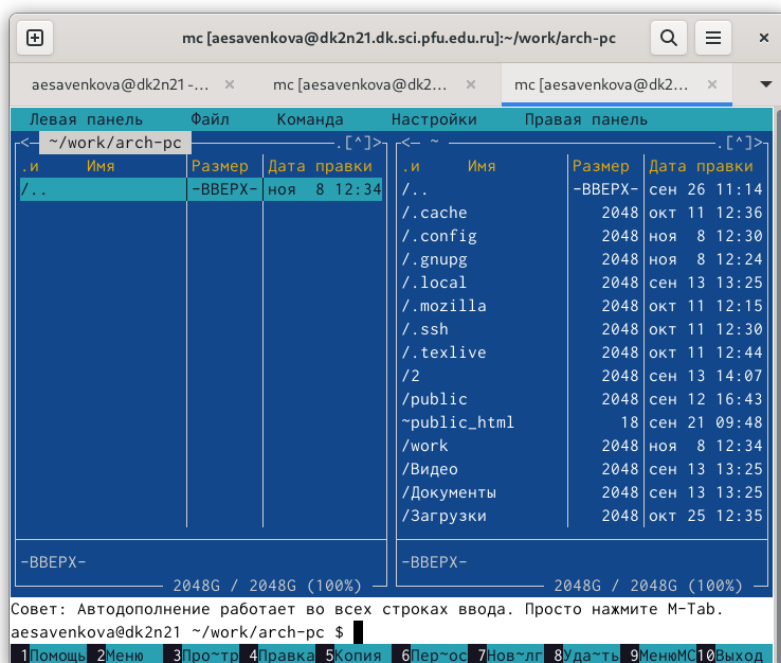


Рис. 4.2: mc

С помощью F7 создаю папку lab05 и перейдите в созданный каталог (рис. 4.3 - 4.4).

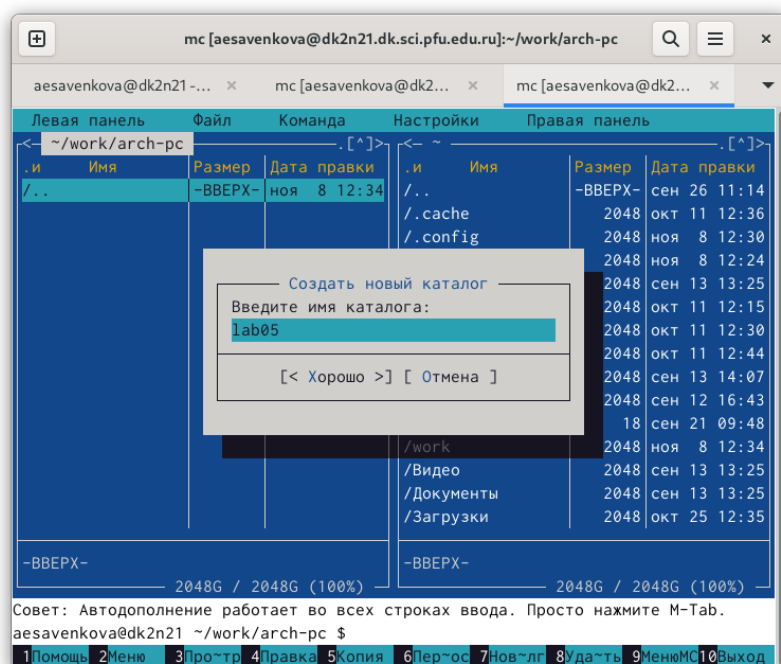


Рис. 4.3: Создание нового каталога

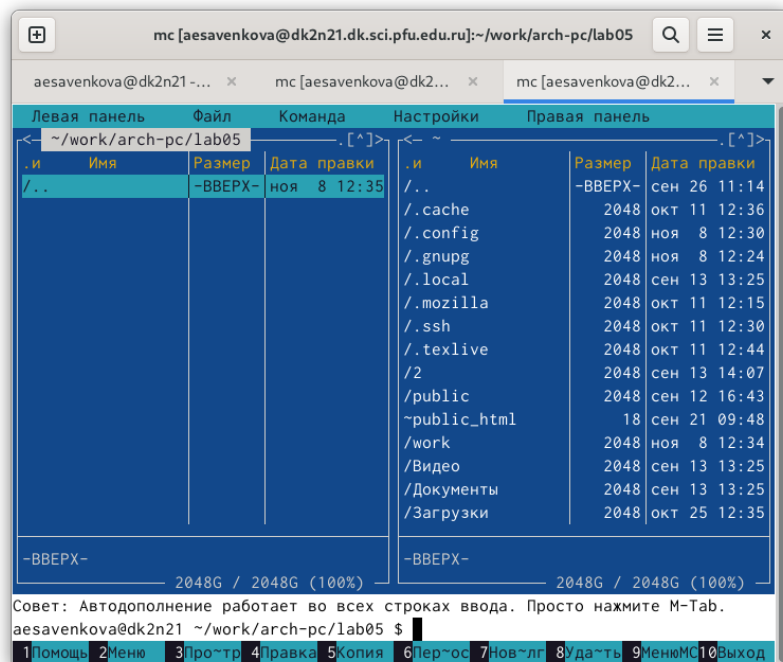


Рис. 4.4: Новый каталог

Пользуясь строкой ввода и командой touch создаю файл lab5-1.asm (рис. 4.5).

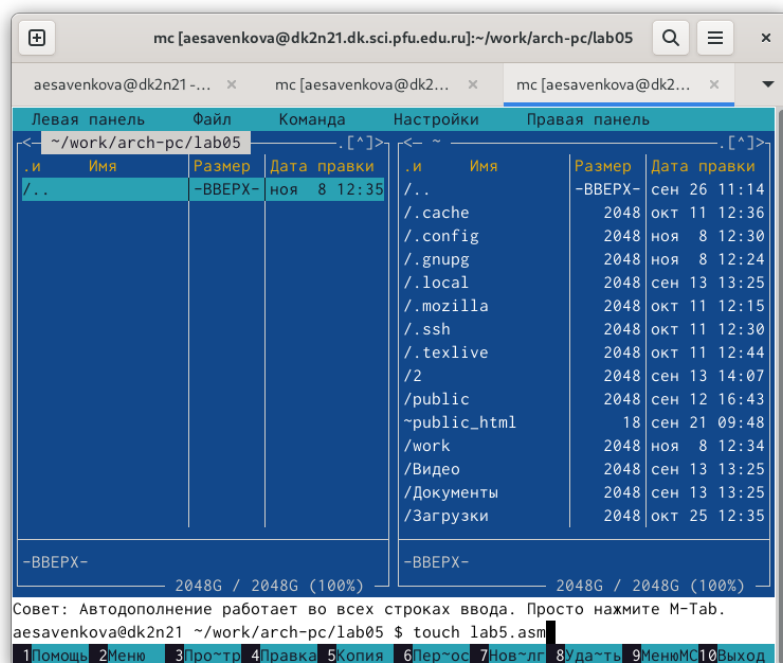


Рис. 4.5: Новый файл

С помощью функциональной клавиши F4 открываю файл lab5-1.asm для редактирования во встроенном редакторе (рис. 4.6).

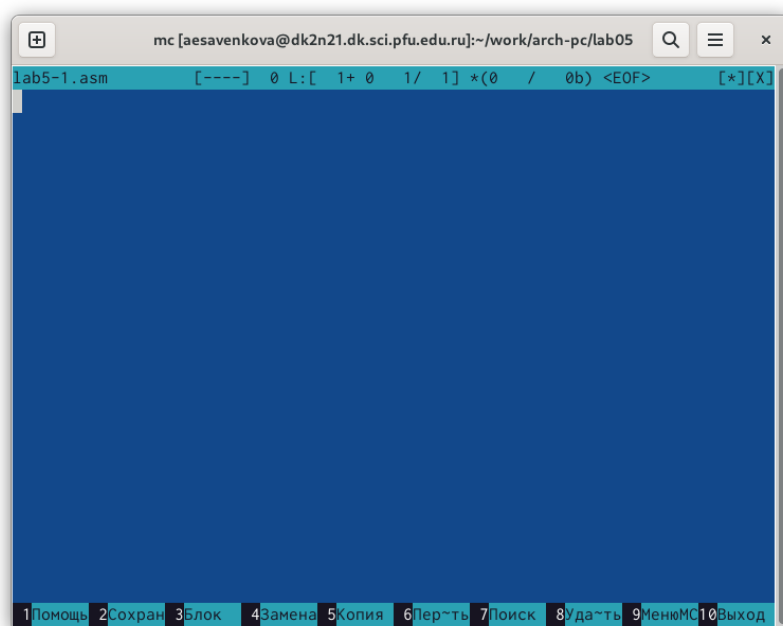


Рис. 4.6: Новый файл

Ввожу текст программы из листинга 5.1, сохраняю изменения и закрываю файл (рис. 4.7 - 4.8).

```
lab5-1.asm [-M--] 10 L:[ 1+ 0 1/ 20] *(10 / 249b) 0097 0x061 [*][X]
SECTION .data
msg: DB 'Введите строку:',10

msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h
mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h
```

1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюМС 10Выход

Рис. 4.7: Измененный файл

```
/afs/.dk.sci.pfu.edu.ru~ch-pc/lab05/lab5-1.asm 249/249 100%
SECTION .data
msg: DB 'Введите строку:',10

msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h
mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h
```

1Помощь 2Раз-рн 3Выход 4Нех 5Пер-ти 6 7Поиск 8Исх-ый 9Формат 10Выход

Рис. 4.8: Измененный файл

Транслирую текст программы lab5-1.asm в объектный файл. Выполняю компоновку объектного файла и запускаю получившийся исполняемый файл. Програм-

ма выводит строку 'Введите строку:' и ожидает ввода с клавиатуры. На запрос ввожу свои ФИО (рис. 4.9).

```
aesavenkova@dk2n21 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.asm
aesavenkova@dk2n21 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o
aesavenkova@dk2n21 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку:
Савенкова Алиса Евгеньевна
```

Рис. 4.9: Запуск файла

В одной из панелей mc открываю каталог с файлом lab5-1.asm. В другой панели каталог со скаченным файлом in_out.asm. Копирую файл in_out.asm в каталог с файлом lab5-1.asm с помощью функциональной клавиши F5 (рис. 4.10).

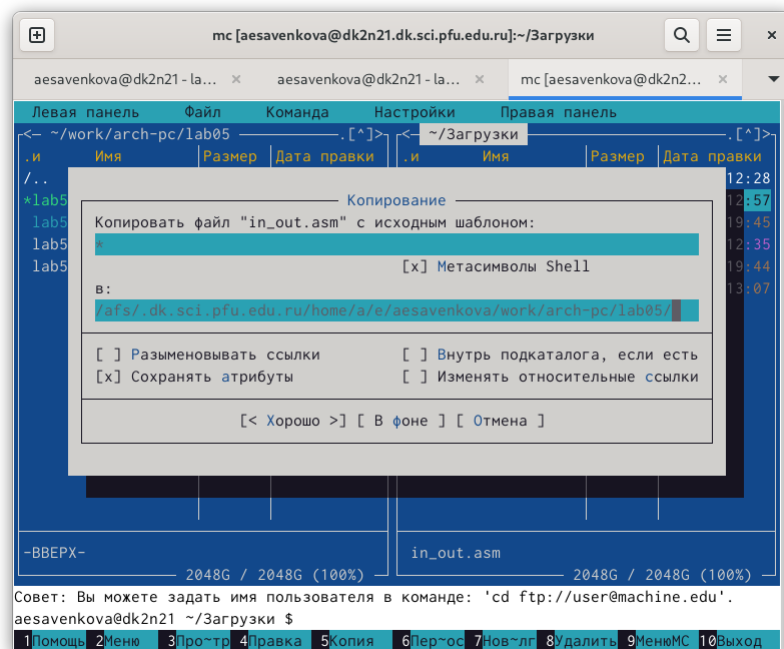


Рис. 4.10: Копирование файла

С помощью функциональной клавиши F6 создаю копию файла lab5-1.asm с именем lab5-2.asm. Выделяю файл lab5-1.asm, нажимаю клавишу F6, ввожу имя файла lab5-2.asm и нажимаю клавишу Enter (рис. 4.11).

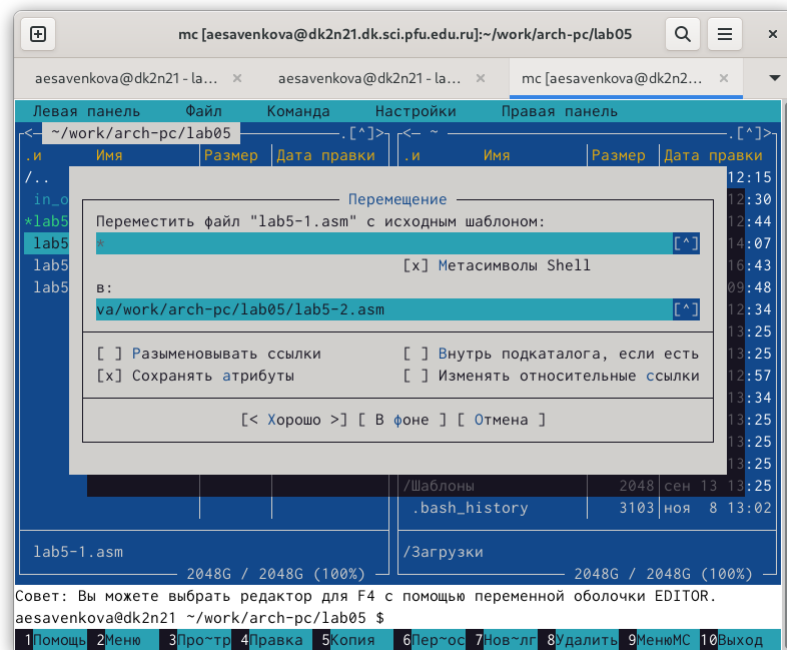


Рис. 4.11: Копирование файла

Исправляю текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in_out.asm в соответствии с листингом 5.2. Создаю исполняемый файл и проверяю его работу (рис. 4.12 - 4.13).

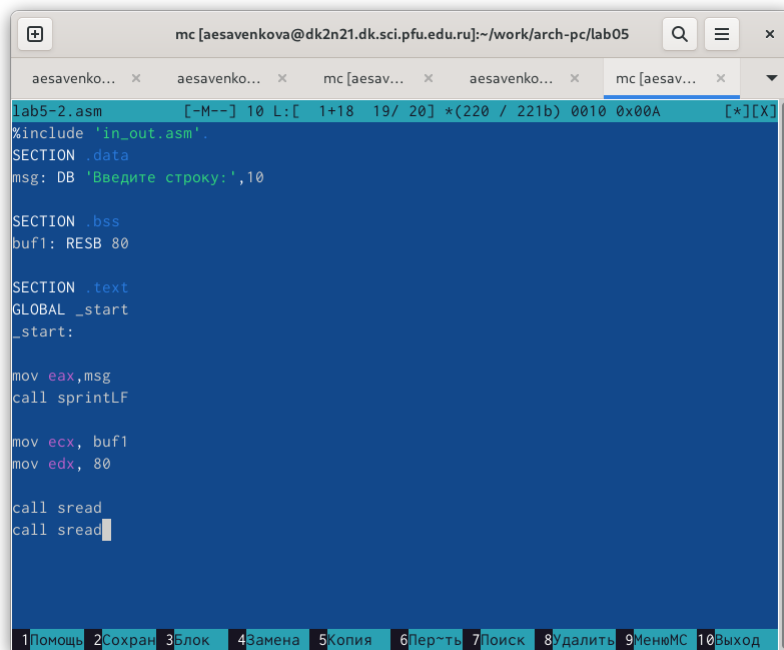


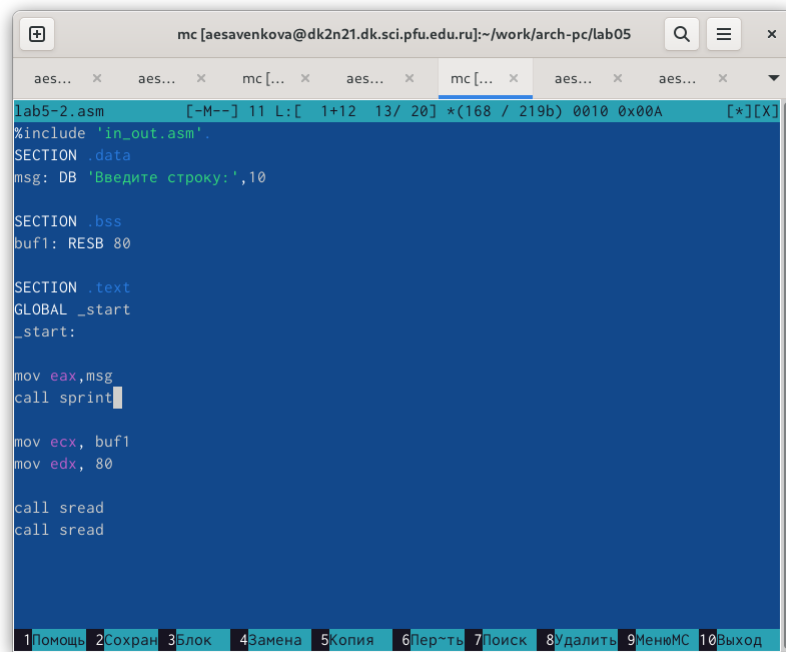
Рис. 4.12: Изменение файла

```

aesavenkova@dk2n21 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
aesavenkova@dk2n21 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
aesavenkova@dk2n21 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку:
Савенкова Алиса Евгеньевна
  
```

Рис. 4.13: Запуск файла

В файле lab5-2.asm заменяю подпрограмму sprintf на printf. Создаю исполняемый файл и проверяю его работу (рис. 4.14 - 4.15).



```
lab5-2.asm [-M--] 11 L: [ 1+12 13/ 20] *(168 / 219b) 0010 0x00A [*][X]
#include 'in_out.asm'.
SECTION .data
msg: DB 'Введите строку:',10

SECTION .bss
buf1: RESB 80

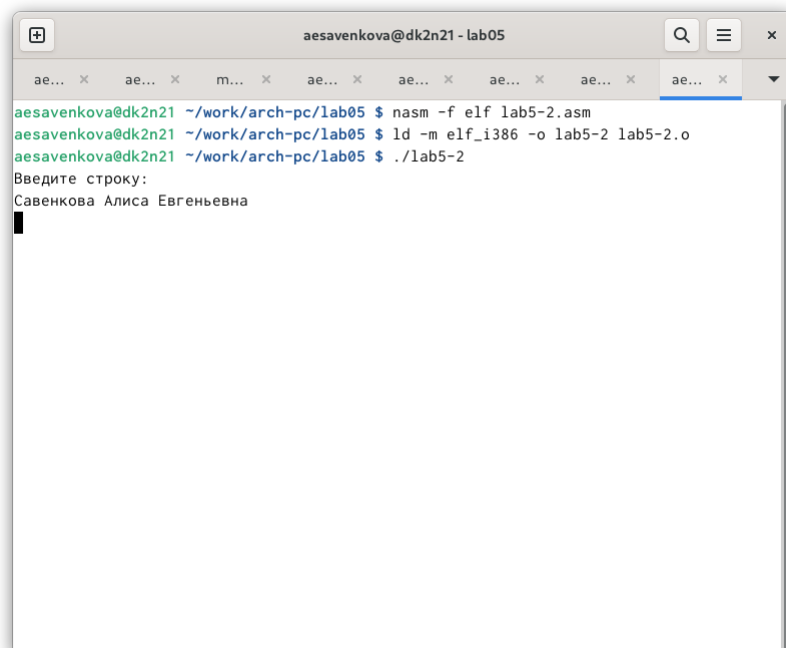
SECTION .text
GLOBAL _start
_start:

mov eax,msg
call sprint

mov ecx, buf1
mov edx, 80

call sread
call sread
```

Рис. 4.14: Изменение файла



```
aesavenkova@dk2n21 ~/work/arch-pc/lab05
aesavenkova@dk2n21 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
aesavenkova@dk2n21 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
aesavenkova@dk2n21 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку:
Савенкова Алиса Евгеньевна
```

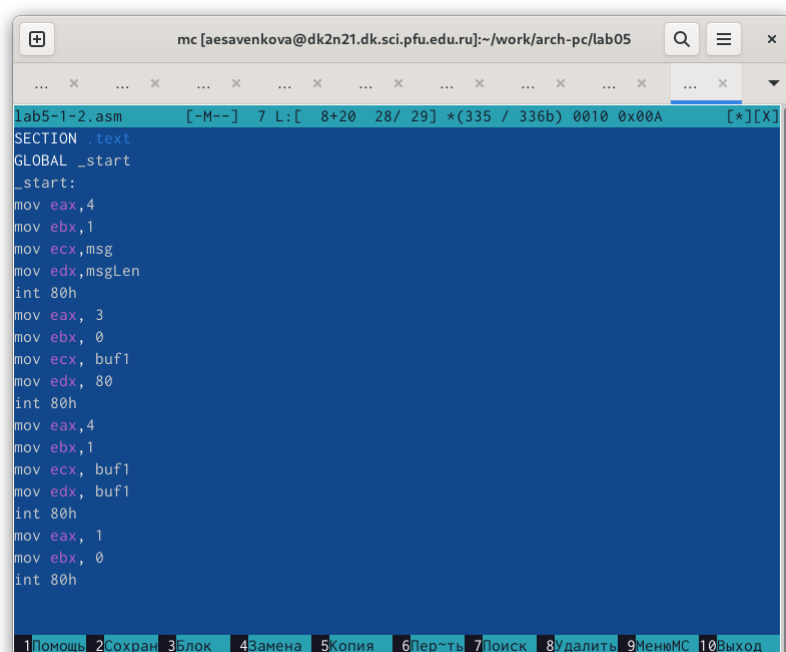
Рис. 4.15: Запуск файла

Разница в том, что во втором файле ввод своего ФИО происходит не через

строку, а сразу после “Введите строку:”.

5 Задания для самостоятельной работы

Создаю копию файла lab5-1.asm. Вношу изменения в программу (рис. 5.1).

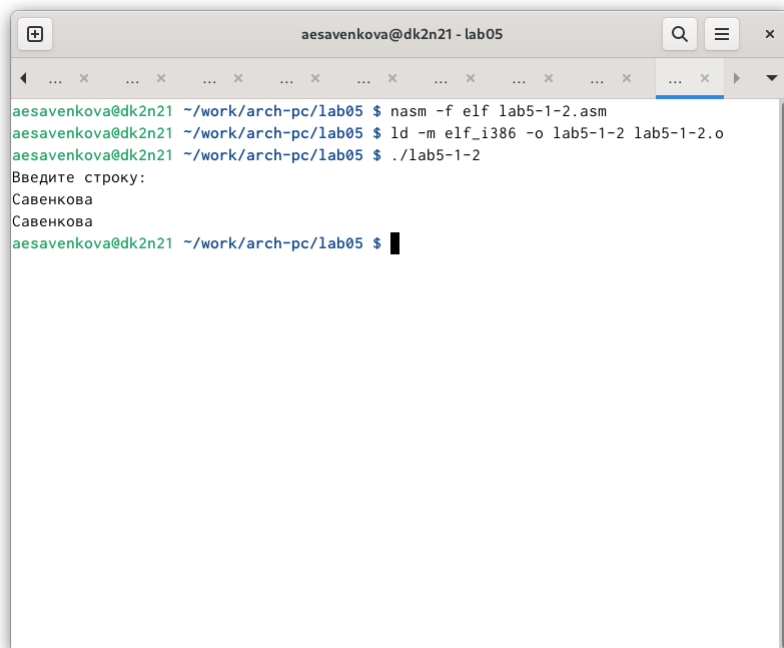


```
lab5-1-2.asm [-M--] 7 L: [ 8+20 28/ 29] *(335 / 336b) 0010 0x00A [*][X]
SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h
mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h
mov eax,4
mov ebx,1
mov ecx, buf1
mov edx, buf1
int 80h
mov eax, 1
mov ebx, 0
int 80h
```

1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Удалить 9МенюМС 10Выход

Рис. 5.1: Изменение файла

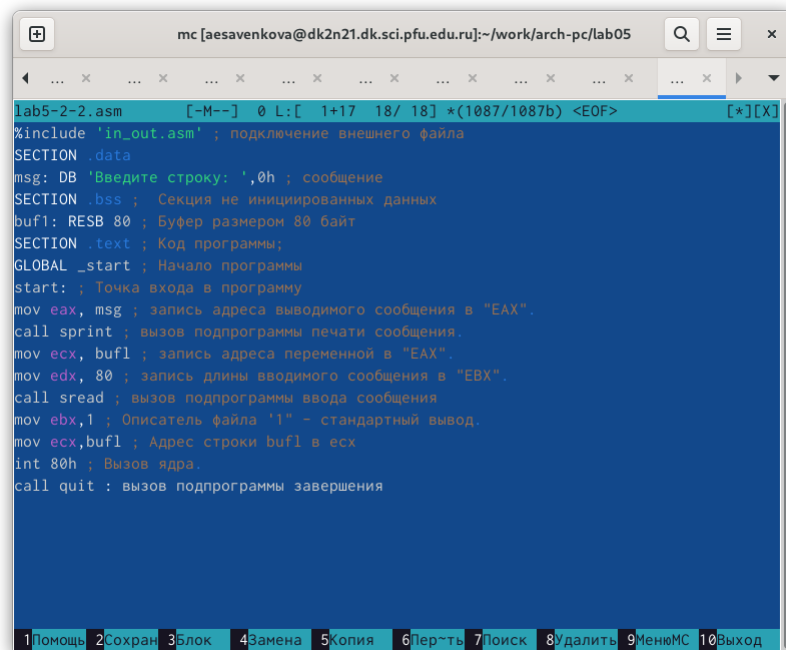
Получаю исполняемый файл и проверяю его работу. На приглашение ввести строку ввожу свою фамилию (рис. 5.2).

A terminal window titled 'aesavenkova@dk2n21 - lab05' with a search icon, menu icon, and close button in the title bar. The terminal shows a series of commands and their outputs. The first command is 'nasm -f elf lab5-1-2.asm', followed by 'ld -m elf_i386 -o lab5-1-2 lab5-1-2.o', and then './lab5-1-2'. The output of the last command is 'Введите строку:' followed by 'Савенкова' on two separate lines. The prompt 'aesavenkova@dk2n21 ~/work/arch-pc/lab05 \$' is visible at the end of the last line.

```
aesavenkova@dk2n21 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1-2.asm
aesavenkova@dk2n21 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1-2 lab5-1-2.o
aesavenkova@dk2n21 ~/work/arch-pc/lab05 $ ./lab5-1-2
Введите строку:
Савенкова
Савенкова
aesavenkova@dk2n21 ~/work/arch-pc/lab05 $
```

Рис. 5.2: Запуск файла

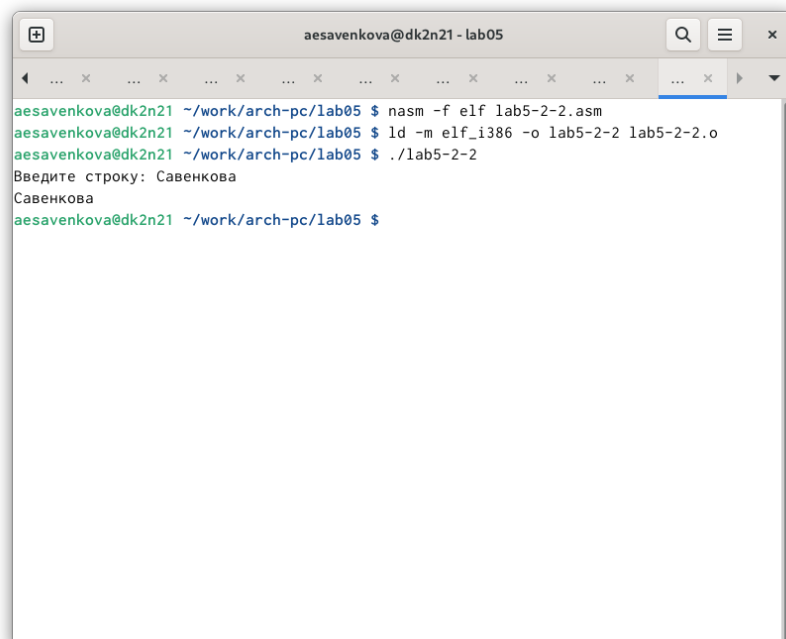
Создаю копию файла lab5-2.asm. Исправляю текст программы с использованием подпрограмм из внешнего файла in_out.asm (рис. 5.3).



```
lab5-2-2.asm [-M--] 0 L: [ 1+17 18/ 18] *(1087/1087b) <EOF> [*][X]
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data
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 ebx, 1 ; Описатель файла '1' - стандартный вывод.
mov ecx, buf1 ; Адрес строки buf1 в ecx
int 80h ; Вызов ядра.
call quit ; вызов подпрограммы завершения
```

Рис. 5.3: Создание копии файла

Проверяю работу файла (рис. 5.4).



```
aesavenkova@dk2n21 - lab05
aesavenkova@dk2n21 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2-2.asm
aesavenkova@dk2n21 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2-2 lab5-2-2.o
aesavenkova@dk2n21 ~/work/arch-pc/lab05 $ ./lab5-2-2
Введите строку: Савенкова
Савенкова
aesavenkova@dk2n21 ~/work/arch-pc/lab05 $
```

Рис. 5.4: Запуск файла

6 Выводы

При выполнении данной лабораторной работы я приобрела практические навыки работы в Midnight Commander и освоила инструкции языка ассемблера `mov` и `int`.

Список литературы