

# **Лабораторная работа №5**

**Язык разметки Markdown**

Солдатов Алексей

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>5</b>	<b>Выводы</b>	<b>20</b>
	<b>Список литературы</b>	<b>21</b>

## Список иллюстраций

4.1	Открытие mc из терминала . . . . .	8
4.2	Переход в каталог и создание там папки . . . . .	8
4.3	Создание файла . . . . .	9
4.4	Работа с файлом . . . . .	9
4.5	Проверка . . . . .	10
4.6	Создание файла . . . . .	10
4.7	Скачивание . . . . .	10
4.8	Перемещение . . . . .	11
4.9	Копирование . . . . .	11
4.10	Редактирование . . . . .	12
4.11	Создание файла . . . . .	12
4.12	Замена . . . . .	13
4.13	Создание и проверка . . . . .	13
4.14	Создание копии . . . . .	14
4.15	Редактирование . . . . .	15
4.16	Создание и проверка файла . . . . .	17
4.17	Создание копии . . . . .	17
4.18	Изменение текста программы . . . . .	18
4.19	Создание и проверка файла . . . . .	19

# Список таблиц

3.1	Описание некоторых каталогов файловой системы GNU Linux . .	7
-----	---	---

# 1 Цель работы

Приобрести практические навыки работы в Midnight Commander. Освоить инструкции языка ассемблера mov и int.

## 2 Задание

1. Основы работы с тс
2. Структура программы на языке ассемблера NASM
3. Подключение внешнего файла
4. Выполнение заданий для самостоятельной работы

### 3 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы. Например, в табл. 3.1 приведено краткое описание стандартных каталогов Unix.

Таблица 3.1: Описание некоторых каталогов файловой системы GNU Linux

Имя каталога	Описание каталога
/	Корневая директория, содержащая всю файловую
/bin	Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям
/etc	Общесистемные конфигурационные файлы и файлы конфигурации установленных программ
/home	Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя
/media	Точки монтирования для сменных носителей
/root	Домашняя директория пользователя root
/tmp	Временные файлы
/usr	Вторичная иерархия для данных пользователя

Более подробно об Unix см. в [1–6].

## 4 Выполнение лабораторной работы

### 1. Основы работы mc

Открыл Midnight Commander (рис. [fig?];001).

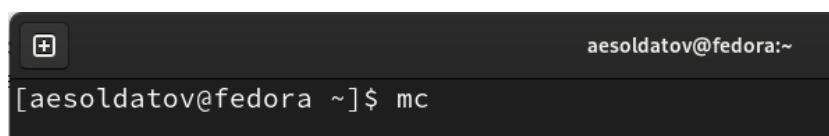


Рис. 4.1: Открытие mc из терминала

Пользуясь стрелочками перешел в каталог “~/work/arch-pc” и создал там папку “lab05” (рис. 4.2).

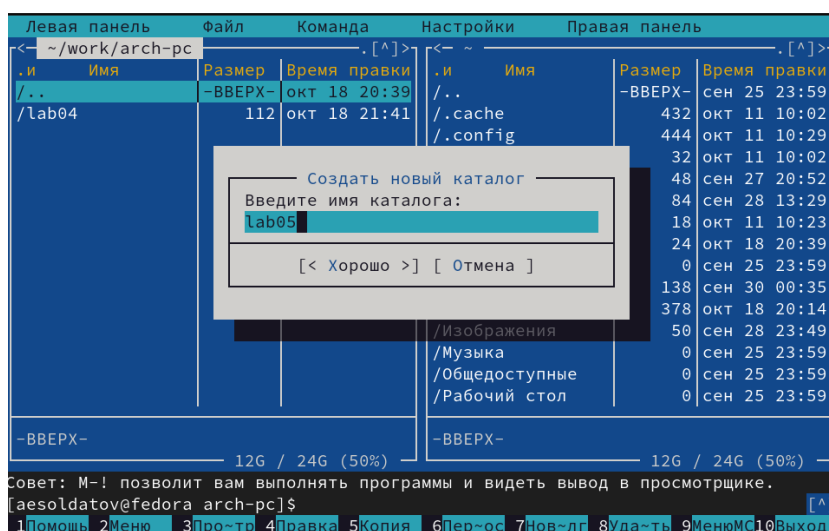


Рис. 4.2: Переход в каталог и создание там папки



Перешел в созданный каталог и пользуясь строкой ввода создал в нем файл “lab5-1.asm” (рис. 4.3).

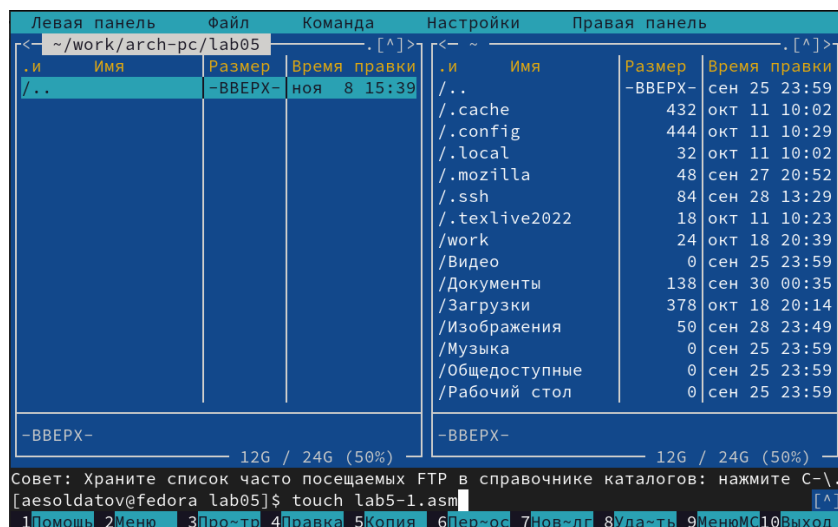


Рис. 4.3: Создание файла

## 2. Структура программы на языке ассемблера NASM

С помощью клавиши F4 открыл созданный файл и ввел текст программы, сохранил и закрыл его (рис. 4.4).

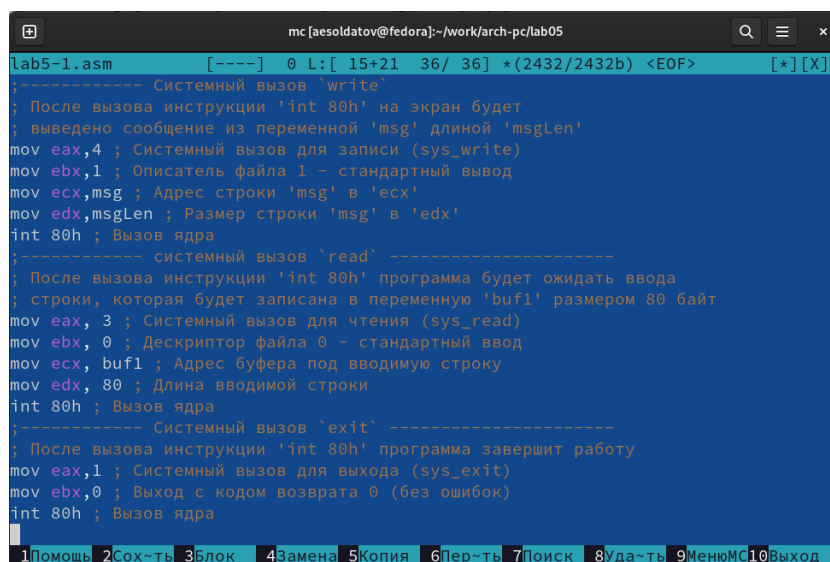
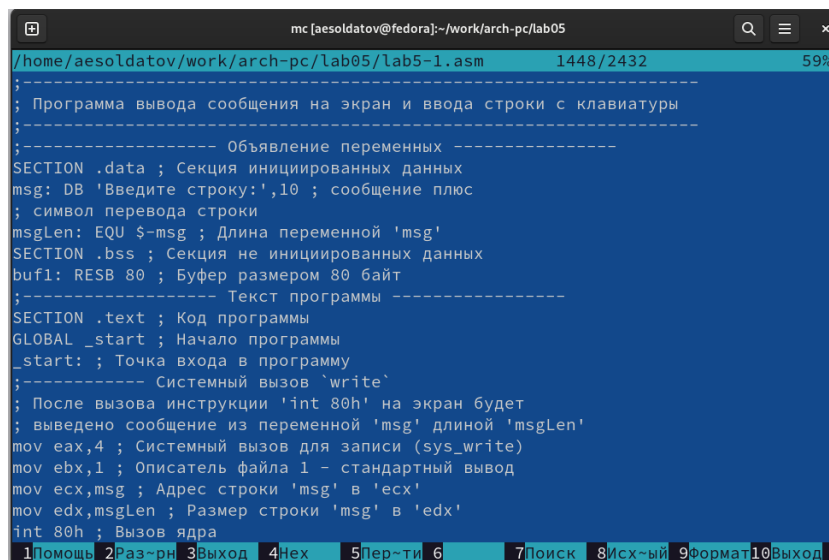


Рис. 4.4: Работа с файлом

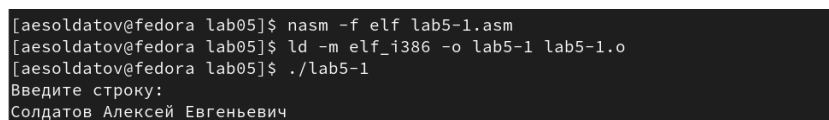
С помощью клавиши F3 откройте файл “lab5-1.asm” для просмотра и убедился, что файл содержит текст программы (рис. 4.5).



```
mc [aesoldatov@fedora]:~/work/arch-pc/lab05
/home/aesoldatov/work/arch-pc/lab05/lab5-1.asm 1448/2432 59%
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
;----- 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
1Помощь 2Раз-рн 3Выход 4Нех 5Пер-ти 6 7Поиск 8Исх-ый 9Формат10Выход
```

Рис. 4.5: Проверка

Создал исполняемый файл и запустил его, ввел свои ФИО (рис. 4.6).



```
[aesoldatov@fedora lab05]$ nasm -f elf lab5-1.asm
[aesoldatov@fedora lab05]$ ld -m elf_i386 -o lab5-1 lab5-1.o
[aesoldatov@fedora lab05]$ ./lab5-1
Введите строку:
Солдатов Алексей Евгеньевич
```

Рис. 4.6: Создание файла

### 3. Подключение внешнего файла

Скачал файл “in\_out.asm” со страницы курса в ТУИС (рис. 4.7).

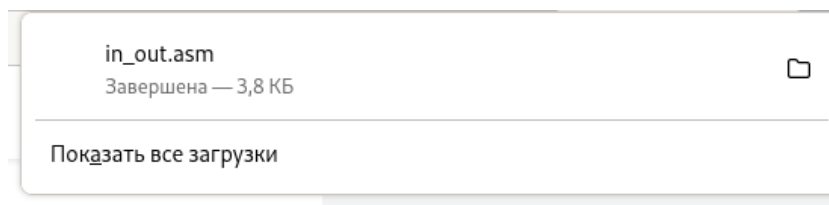


Рис. 4.7: Скачивание

Перенес скачанный файл в каталог с файлом “lab5-1” (рис. 4.8).

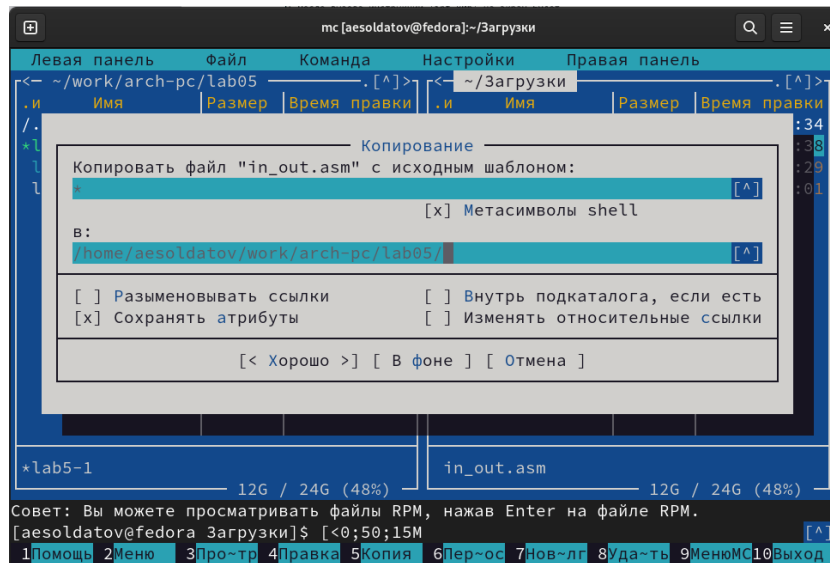


Рис. 4.8: Перемещение

С помощью клавиши F6 создал копию файла “lab5-1.asm” с именем “lab5-2.asm” (рис. 4.9).

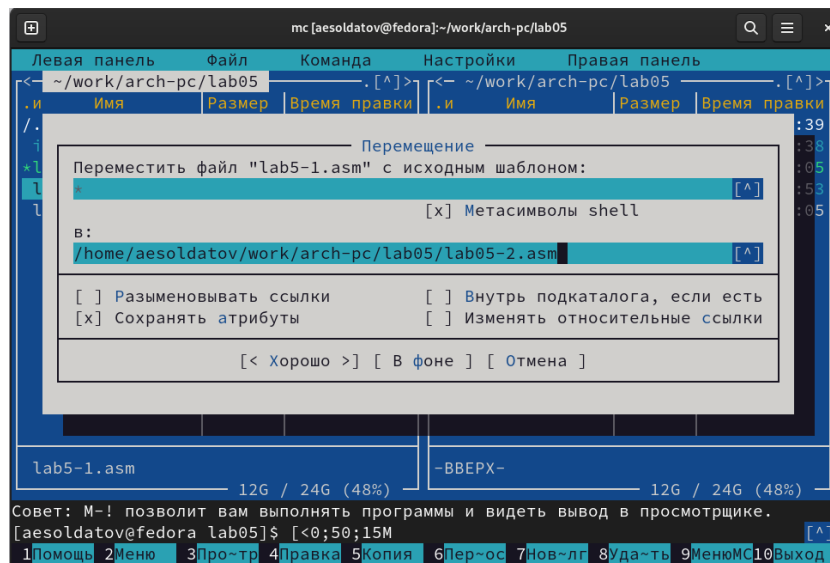
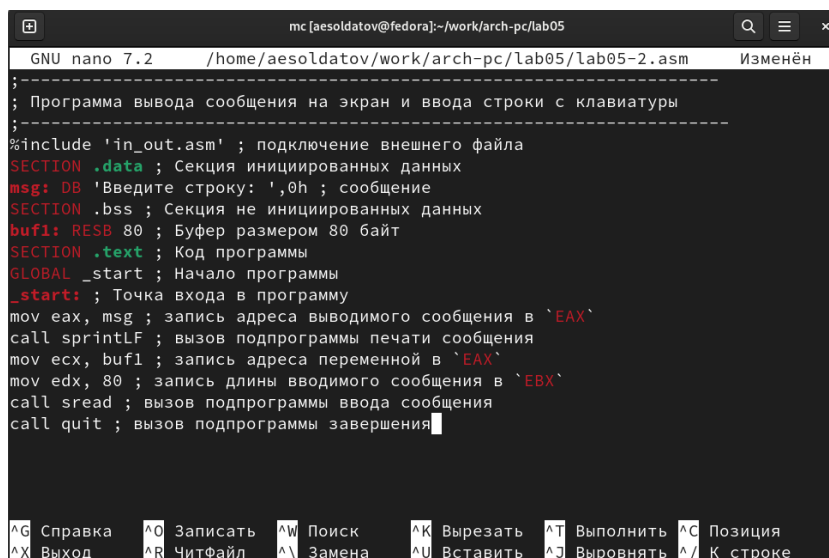


Рис. 4.9: Копирование

Исправил текст программы в файле “lab5-2.asm”, добавив использование под-

программ из внешнего файла “in\_out.asm” (рис. 4.10).

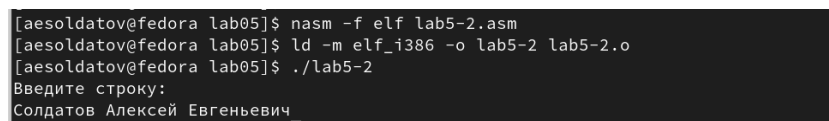


```
mc [aesoldatov@fedora]:~/work/arch-pc/lab05
GNU nano 7.2 /home/aesoldatov/work/arch-pc/lab05/lab05-2.asm  Изменён
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция инициированных данных
msg: DB '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 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 Справка  ^O Записать  ^W Поиск    ^K Вырезать  ^T Выполнить ^C Позиция
^X Выход    ^R Читфайл  ^\ Замена   ^U Вставить  ^J Выводить  ^_ К строке
```

Рис. 4.10: Редактирование

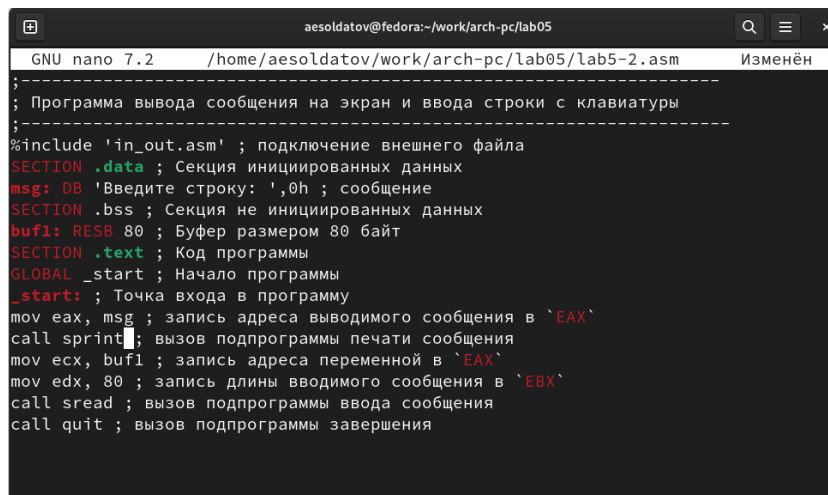
Создал исполняемый файл и проверил его работу (рис. 4.11).



```
[aesoldatov@fedora lab05]$ nasm -f elf lab5-2.asm
[aesoldatov@fedora lab05]$ ld -m elf_i386 -o lab5-2 lab5-2.o
[aesoldatov@fedora lab05]$ ./lab5-2
Введите строку:
Солдатов Алексей Евгеньевич
```

Рис. 4.11: Создание файла

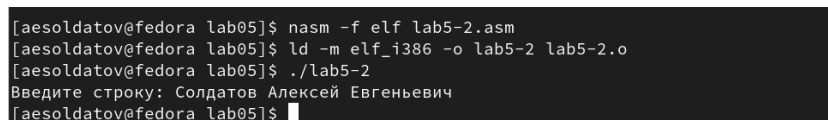
В файле “lab5-2.asm” замените подпрограмму “sprintf” на “print” (рис. 4.12).



```
GNU nano 7.2 /home/aesoldatov/work/arch-pc/lab05/lab5-2.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
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. 4.12: Замена

Создал исполняемый файл и проверил его работу (рис. 4.13).



```
[aesoldatov@fedora lab05]$ nasm -f elf lab5-2.asm
[aesoldatov@fedora lab05]$ ld -m elf_i386 -o lab5-2 lab5-2.o
[aesoldatov@fedora lab05]$ ./lab5-2
Введите строку: Солдатов Алексей Евгеньевич
[aesoldatov@fedora lab05]$
```

Рис. 4.13: Создание и проверка

Подпрограмма “sprintLF” запрашивает ввод с новой строки, а “sprint” запрашивает ввод, сохраняя строку

#### 4. Выполнение заданий для самостоятельной работы

Создал копию файла “lab5-1.asm”, назвал ее “lab5-1-1.asm” (рис. 4.14).

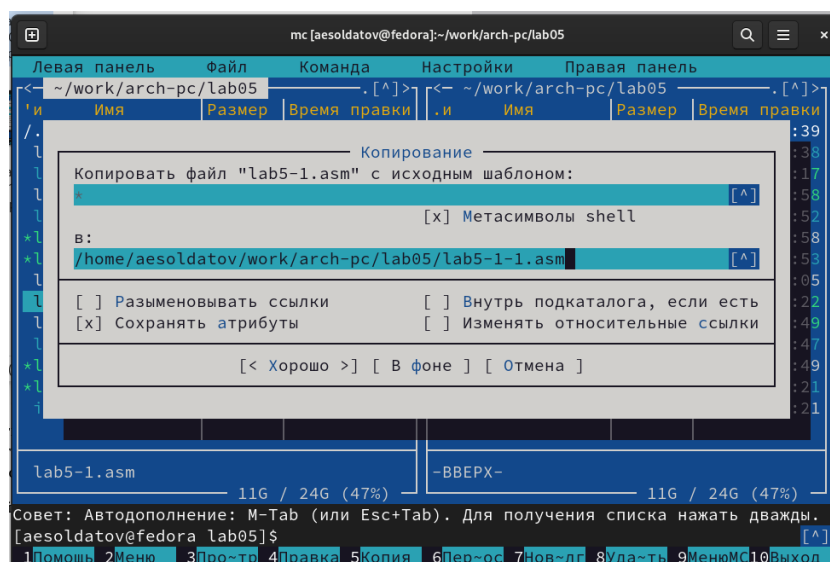


Рис. 4.14: Создание копии

Внес изменения в программу (без использования внешнего файла `in_out.asm`), так чтобы она работала по алгоритму, заданному в ТУИС (рис. 4.15).

```

GNU nano 7.2 /home/aesoldato'
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 ; Дескриптор файла 0 - стандартный ввод
mov ecx, buf1 ; Адрес буфера под вводимую строку
mov edx, 80 ; Длина вводимой строки
int 80h ; Вызов ядра
;----- Системный вызов 'write'
; После вызова инструкции 'int 80h' на экран будет
; выведено сообщение из переменной 'buf1' длиной 80
mov eax,4 ; Системный вызов для записи (sys_write)

```

Рис. 4.15: Редактирование

```

;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
;----- 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 ; Дескриптор файла 0 - стандартный ввод
mov ecx, buf1 ; Адрес буфера под вводимую строку
mov edx, 80 ; Длина вводимой строки
int 80h ; Вызов ядра
;----- Системный вызов 'write'
; После вызова инструкции 'int 80h' на экран будет
; выведено сообщение из переменной 'buf1' длиной 80
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла 1 - стандартный вывод
mov ecx,buf1 ; Адрес введенной строки в 'ecx'
mov edx,80 ; Длина введенной 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

Создал исполняемый файл и проверил его работу (рис. 4.16).

```
[aesoldatov@fedora lab05]$ nasm -f elf lab5-1-1.asm
[aesoldatov@fedora lab05]$ ld -m elf_i386 -o lab5-1-1 lab5-1-1.o
[aesoldatov@fedora lab05]$ ./lab5-1-1
Введите строку:
Солдатов
Солдатов
[aesoldatov@fedora lab05]$
```

Рис. 4.16: Создание и проверка файла

Создал копию файла “lab5-2.asm”, назвал ее “lab5-2-1.asm” (рис. 4.17).

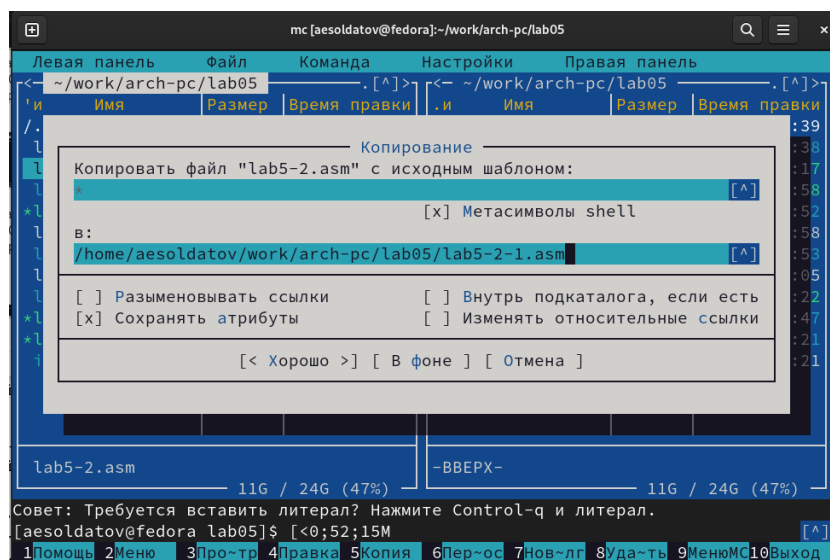
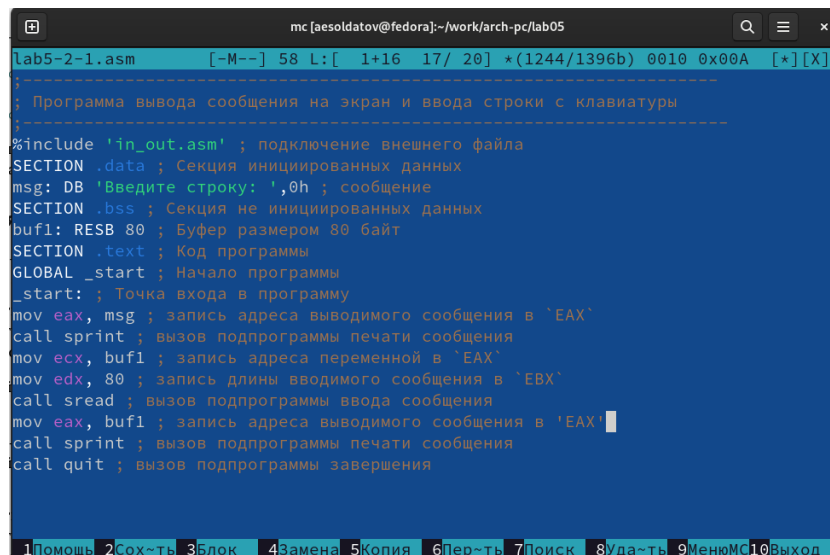


Рис. 4.17: Создание копии

Исправил текст программы с использованием подпрограмм из внешнего файла “in\_out.asm”, так чтобы она работала по алгоритму, заданному в ТУИС (рис. 4.18).



The screenshot shows a text editor window titled 'mc [aesoldatov@fedora]:~/work/arch-pc/lab05'. The file being edited is 'lab5-2-1.asm'. The code is in Russian and includes comments. The code defines a data section with a message 'Введите строку: ', a bss section for a buffer 'buf1' of size 80, and a text section with the program code. The code uses 'sprint' to print the message and 'sread' to read input from the keyboard. The code is as follows:

```
lab5-2-1.asm [-M--] 58 L: [ 1+16 17/ 20] *(1244/1396b) 0010 0x00A [*][X]
;-----
; 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 ; запись адреса выводимого сообщения в `EAX`
call sprint ; вызов подпрограммы печати сообщения
call quit ; вызов подпрограммы завершения
```

At the bottom of the window, there is a menu bar with the following items: 1Помощь, 2Сохранить, 3Блок, 4Замена, 5Копия, 6Перейти, 7Поиск, 8Удалить, 9Меню, 10Выход.

Рис. 4.18: Изменение текста программы

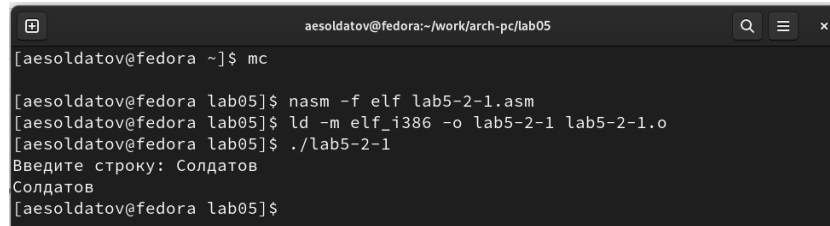
```
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----

#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 ; запись адреса выводимого сообщения в `EAX`
```

**call** sprint ; вызов подпрограммы печати сообщения

**call** quit ; вызов подпрограммы завершения

Создал исполняемый файл и проверил его работу (рис. 4.19).



```
aesoldatov@fedora:~/work/arch-pc/lab05
[aesoldatov@fedora ~]$ mc
[aesoldatov@fedora lab05]$ nasm -f elf lab5-2-1.asm
[aesoldatov@fedora lab05]$ ld -m elf_i386 -o lab5-2-1 lab5-2-1.o
[aesoldatov@fedora lab05]$ ./lab5-2-1
Введите строку: Солдатов
Солдатов
[aesoldatov@fedora lab05]$
```

Рис. 4.19: Создание и проверка файла

## 5 Выводы

Приобрел практические навыки работы в Midnight Commander и освоил инструкции языка ассемблера `mov` и `int`.

## Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016.  
URL: <https://www.gnu.org/software/bash/manual/>.
2. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.