Лабораторная работа №5

Язык разметки Markdown

Солдатов Алексей

Содержание

# 1 Цель работы

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

# 2 Задание

1. Основы работы с mc
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. [1](#tbl:std-dir) приведено краткое описание стандартных каталогов Unix.

Table 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).

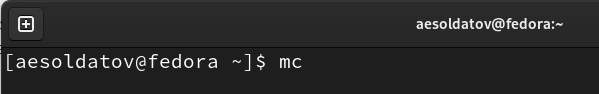


Figure 1: Открытие mc из терминала

Пользуясь стрелочками перешел в каталог “~/work/arch-pc” и создал там папку “lab05” (рис. [2](#fig:002)).

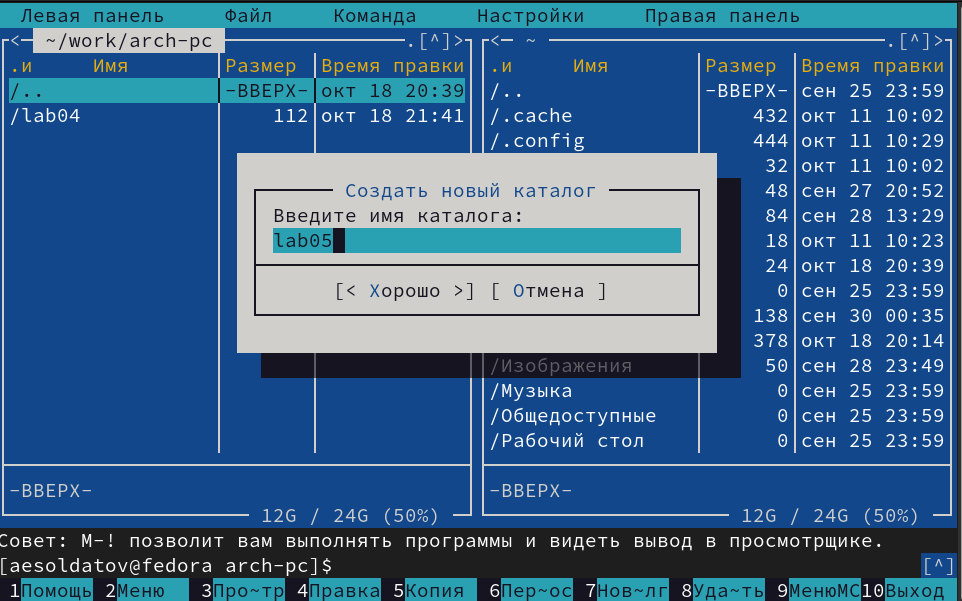


Figure 2: Переход в каталог и создание там папки

Перешел в созданный каталог и пользуясь строкой ввода создал в нем файл “lab5-1.asm” (рис. [3](#fig:003)).

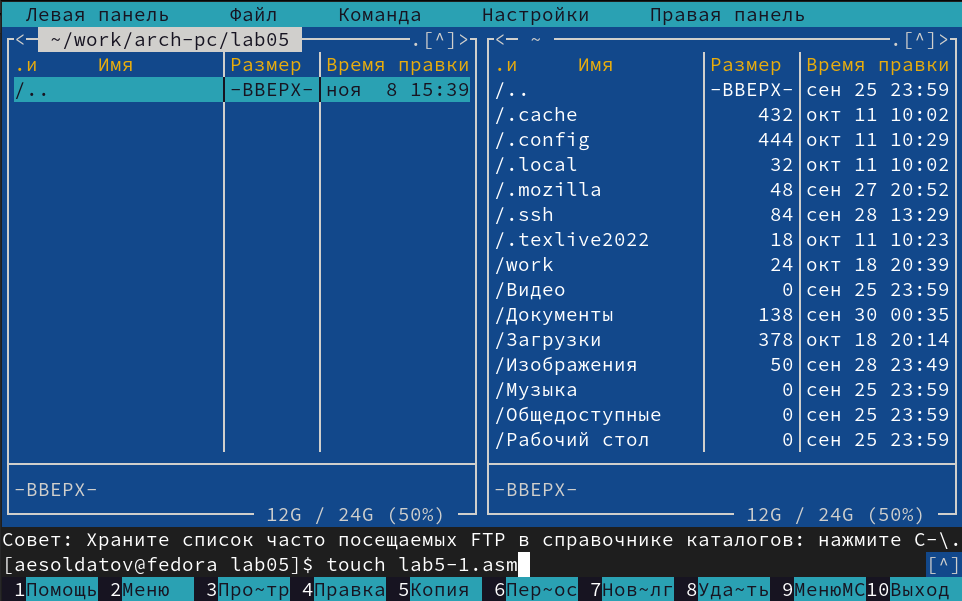


Figure 3: Создание файла

1. Структура программы на языке ассемблера NASM

С помощью клавиши F4 открыл созданный файл и ввел текст программы, сохранил и закрыл его (рис. [4](#fig:004)).

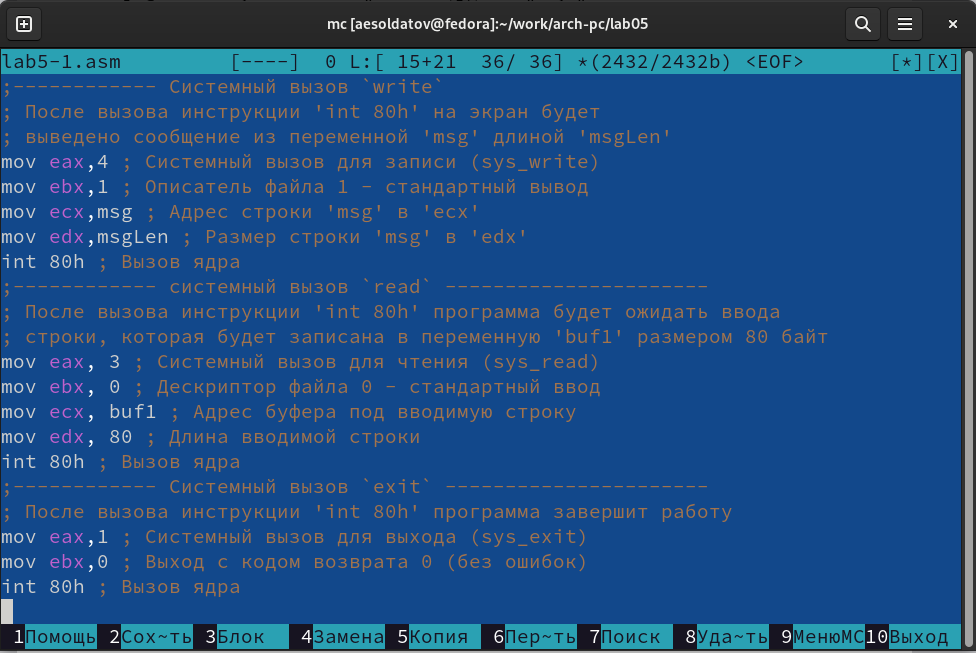


Figure 4: Работа с файлом

С помощью клавиши F3 откройте файл “lab5-1.asm” для просмотра и убедился, что файл содержит текст программы (рис. [5](#fig:005)).

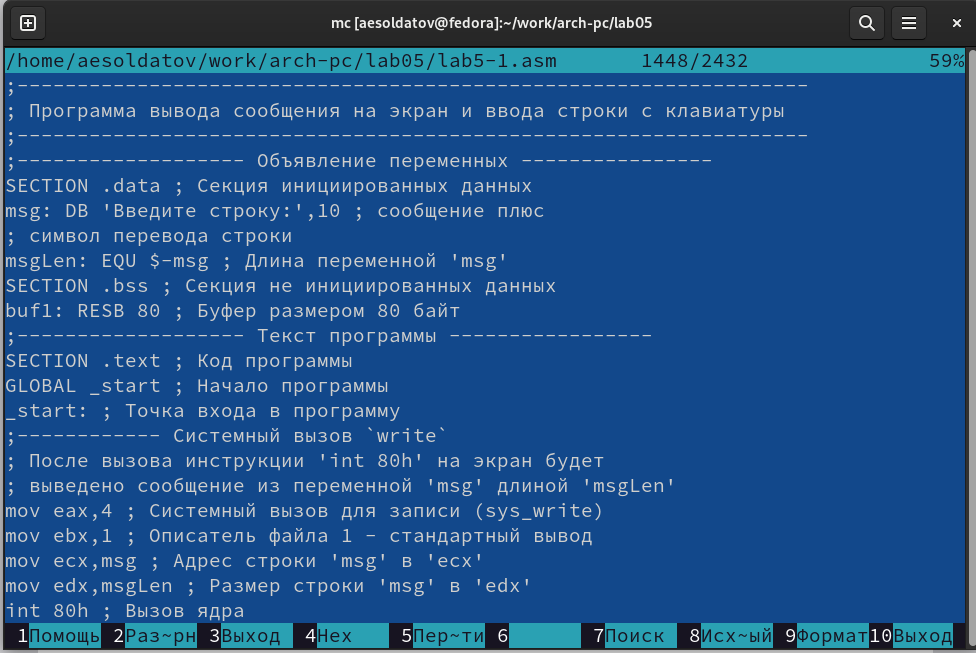


Figure 5: Проверка

Создал исполняемый файл и запустил его, ввел свои ФИО (рис. [6](#fig:006)).

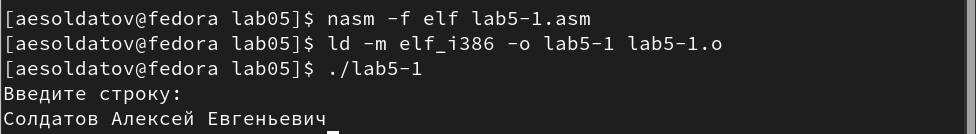


Figure 6: Создание файла

1. Подключение внешнего файла

Скачал файл “in\_out.asm” со страницы курса в ТУИС (рис. [7](#fig:007)).

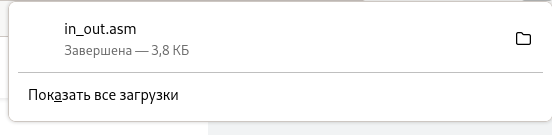


Figure 7: Скачивание

Перенес скачанный файл в каталог с файлом “lab5-1” (рис. [8](#fig:008)).

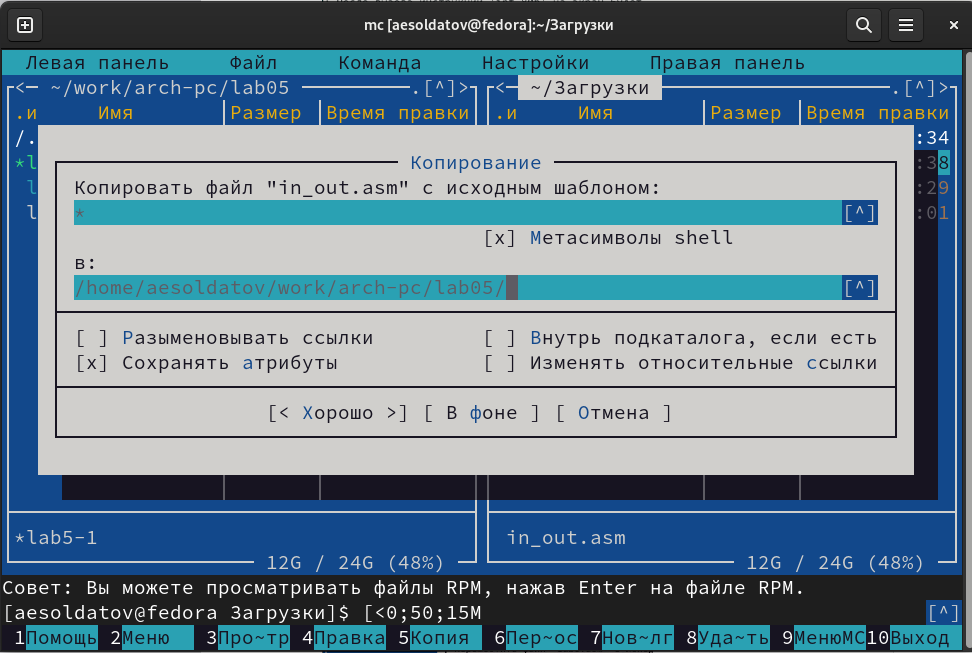


Figure 8: Перемещение

С помощью клавиши F6 создал копию файла “lab5-1.asm” с именем “lab5-2.asm” (рис. [9](#fig:009)).

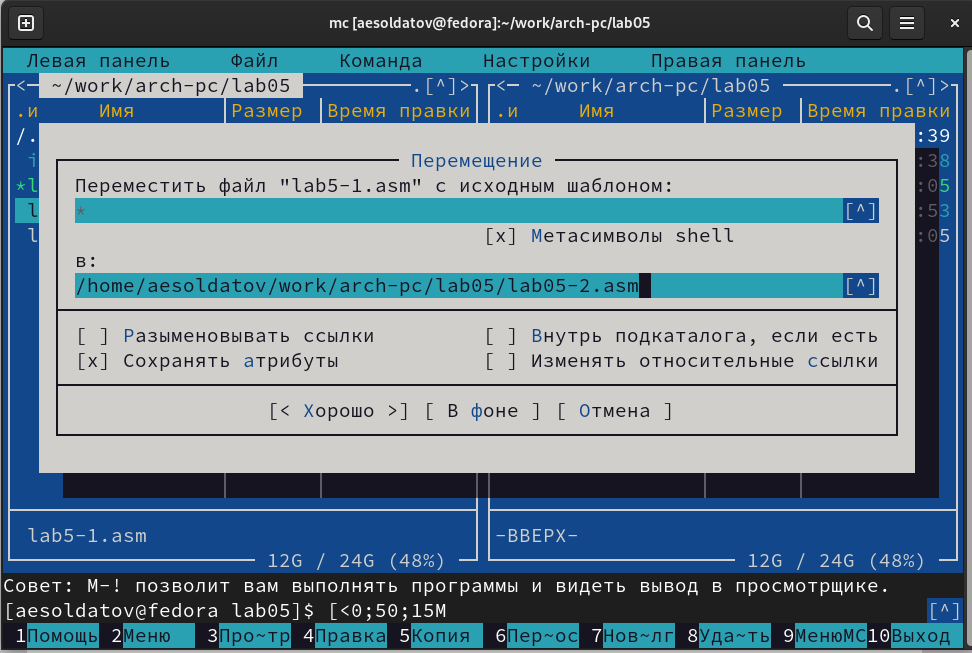


Figure 9: Копирование

Исправил текст программы в файле “lab5-2.asm”, добавив использование подпрограмм из внешнего файла “in\_out.asm” (рис. [10](#fig:010)).

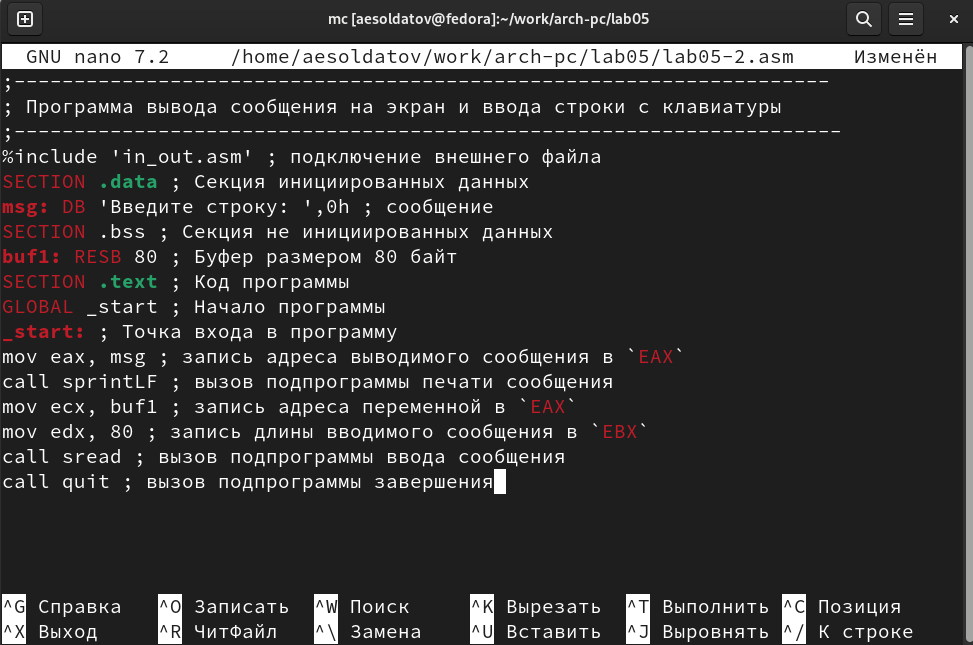


Figure 10: Редактирование

Создал исполняемый файл и проверил его работу (рис. [11](#fig:011)).

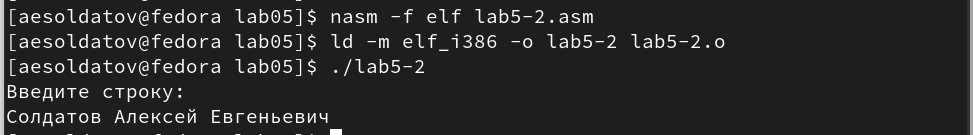


Figure 11: Создание файла

В файле “lab5-2.asm” замените подпрограмму “sprintLF” на “sprint” (рис. [12](#fig:012)).

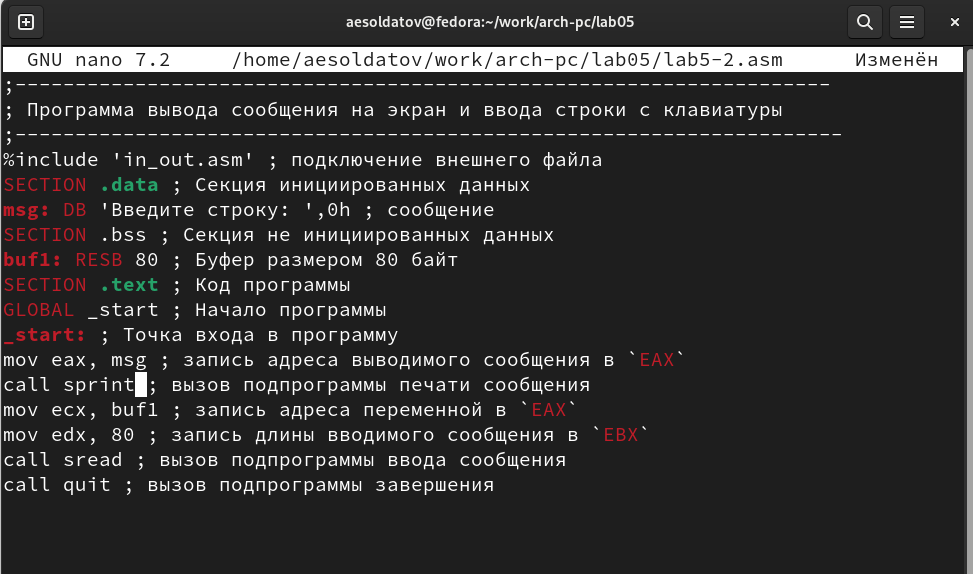


Figure 12: Замена

Создал исполняемый файл и проверил его работу (рис. [13](#fig:013)).

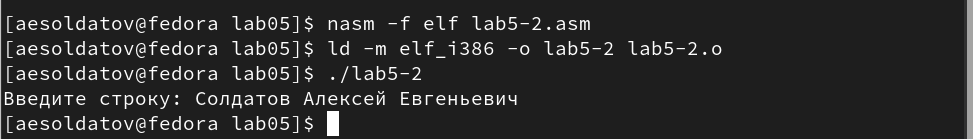


Figure 13: Создание и проверка

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

1. Выполнение заданий для самостоятельной работы

Создал копию файла “lab5-1.asm”, назвал ее “lab5-1-1.asm” (рис. [14](#fig:014)).

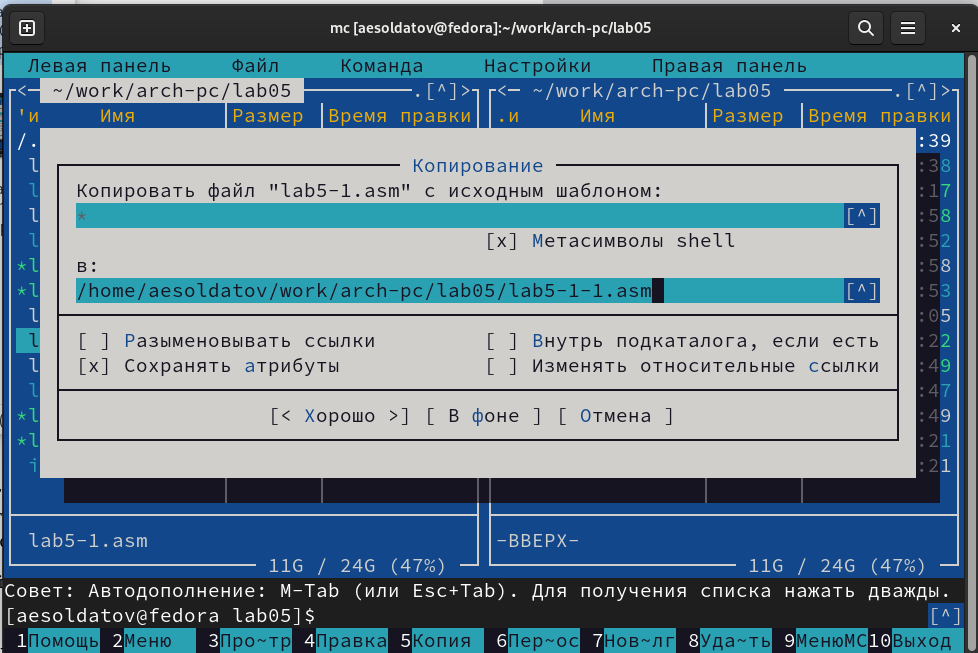


Figure 14: Создание копии

Внес изменения в программу (без использования внешнего файла in\_out.asm), так чтобы она работала по алгоритму, заданному в ТУИС (рис. [15](#fig:015)).

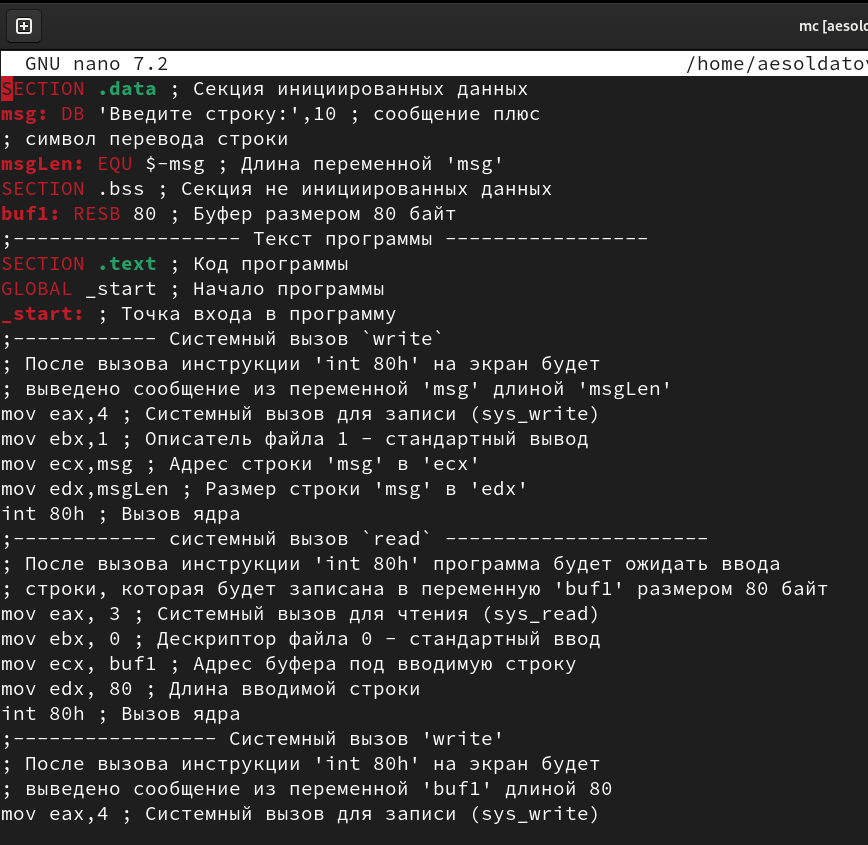


Figure 15: Редактирование

;------------------------------------------------------------------  
; 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  
;------------ Cистемный 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. [16](#fig:016)).

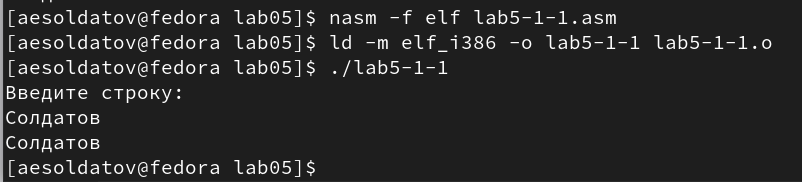


Figure 16: Создание и проверка файла

Создал копию файла “lab5-2.asm”, назвал ее “lab5-2-1.asm” (рис. [17](#fig:017)).

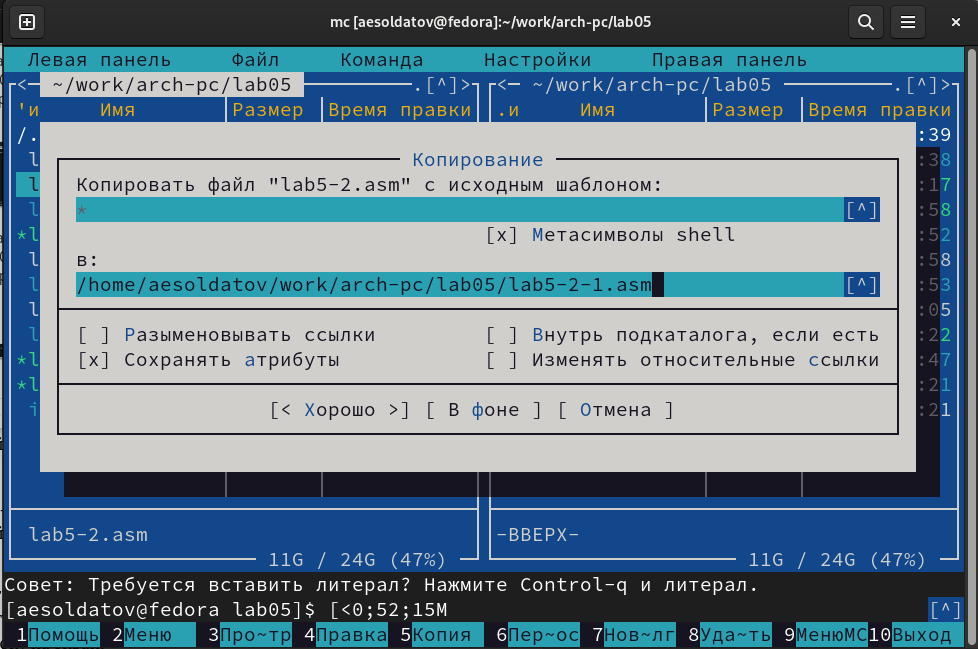


Figure 17: Создание копии

Исправил текст программы с использованием подпрограмм из внешнего файла “in\_out.asm”, так чтобы она работала по алгоритму, заданному в ТУИС (рис. [18](#fig:018)).

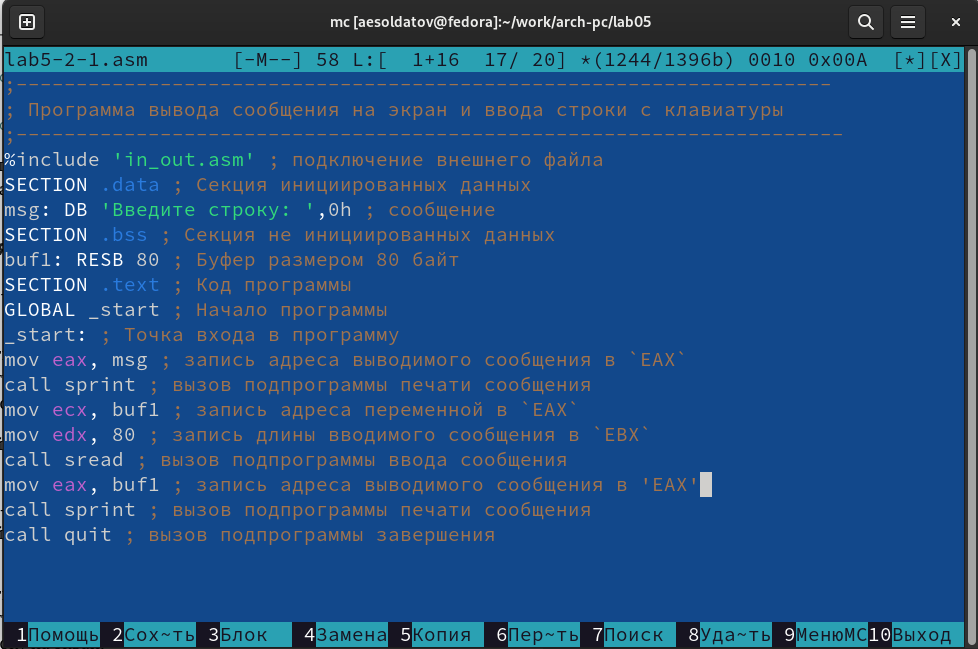


Figure 18: Изменение текста программы

;--------------------------------------------------------------------  
; 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. [19](#fig:019)).

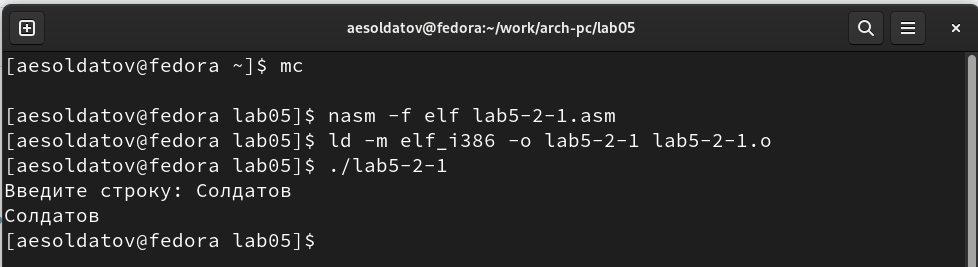


Figure 19: Создание и проверка файла

# 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](http://www.amazon.com/Learning-bash-Shell-Programming-Nutshell/dp/0596009658). O’Reilly Media, 2005. 354 с.

3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.

4. Robbins A. [Bash Pocket Reference](https://www.ncbi.nlm.nih.gov/pubmed/25246403). 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.