

Отчет по лабораторной работе №5

Дисциплина: Архитектура компьютера

Желобицкая П.А.

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	16
	Список литературы	17

Список иллюстраций

2.1	Перемещение между директориями	6
2.2	Создание папки	7
2.3	Создание файла	8
2.4	Открытие файла	9
2.5	Ввод программы	10
2.6	Запуск программы	11
2.7	Перемещение и копирование	11
2.8	Изменение текста программы	12
2.9	Запуск файла	12
2.10	Создание копии	13
2.11	Внесение изменений	13
2.12	Запуск файла	14
2.13	Создание копии	14
2.14	Изменение программы	15
2.15	Запуск файла	15

Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

Открываю Midnight Commander, перехожу в каталог ~/work/arch-pc (рис. 2.1).

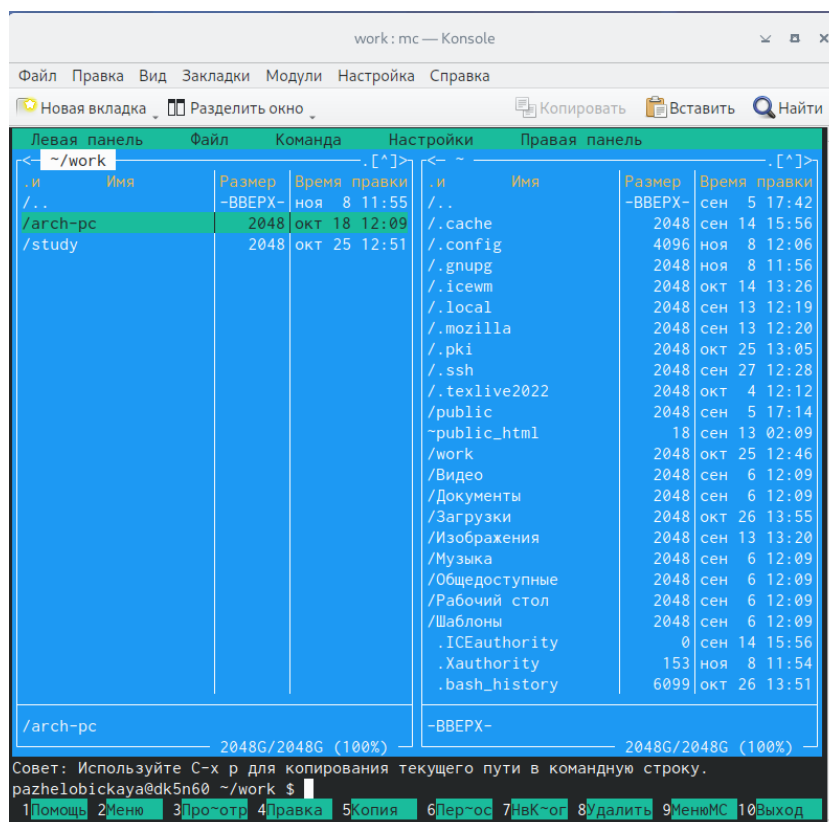


Рис. 2.1: Перемещение между директориями

Создаю папку lab05 (рис. 2.2).

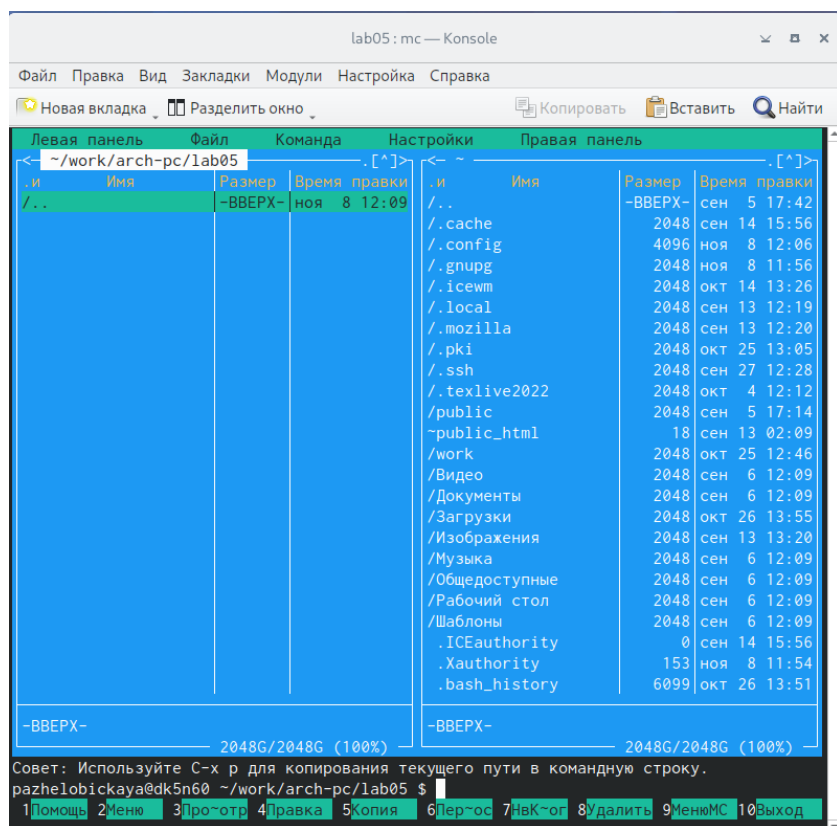


Рис. 2.2: Создание папки

В созданной папке, создаю файл lab5-1.asm (рис. 2.3).

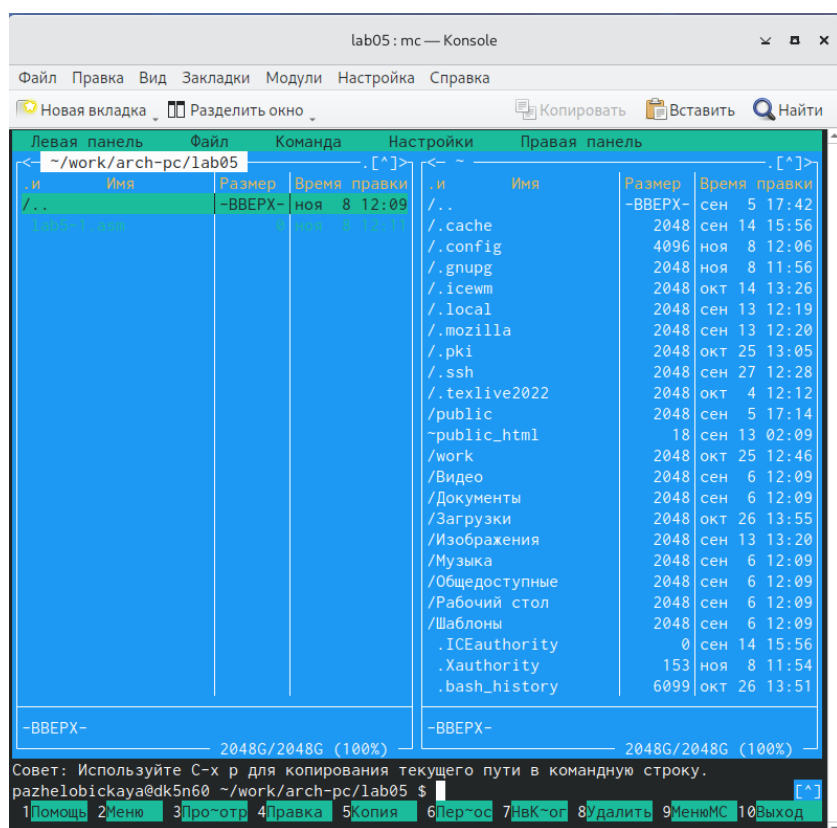


Рис. 2.3: Создание файла

С помощью функциональной клавиши F4 открываю файл `lab5-1.asm` для редактирования во встроенном редакторе (рис. 2.4).

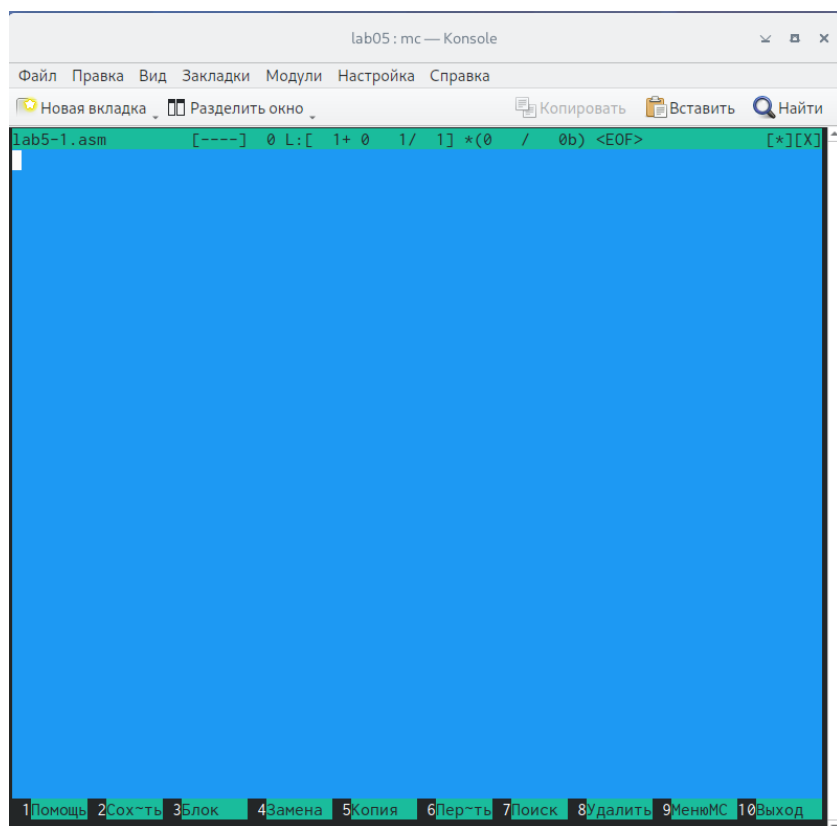


Рис. 2.4: Открытие файла

Вставляю программу в файл (рис. 2.5).

```

lab05: mc — Konsole
Файл  Правка  Вид  Закладки  Модули  Настройка  Справка
Новая вкладка  Разделить окно  Копировать  Вставить  Найти
lab5-1.asm  [----] 35 L: 12+30 42/ 42] *(2685/2685b) <EOF>  [*][X]
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 '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
1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перейти 7Поиск 8Удалить 9МенюМС 10Выход

```

Рис. 2.5: Ввод программы

(рис. ??).

```

lab05: mc — Konsole
Файл  Правка  Вид  Закладки  Модули  Настройка  Справка
Новая вкладка  Разделить окно  Копировать  Вставить  Найти
/afs/.dk.sci.pfu.edu.ru/home/p-/work/arch-pc/lab05/lab5-1.asm  1907/2685  71%
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----

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

SECTION .bss ; Секция не инициализированных 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
1Помощь 2Развернуть 3Выход 4Неч 5Перейти 6 7Поиск 8Исходный 9Формат 10Выход

```

Создаю объектный файл lab5-1.o, выполняю компоновку объектного файла и запускаю исполняемый файл (рис. 2.6).

```
pazhelobickaya@dk5n60 ~ $ cd ~/work/arch-pc/lab05
pazhelobickaya@dk5n60 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.asm
pazhelobickaya@dk5n60 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o
pazhelobickaya@dk5n60 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку:
Zhelobitskaya Polina
pazhelobickaya@dk5n60 ~/work/arch-pc/lab05 $
```

Рис. 2.6: Запуск программы

Перемещаю скачанный файл in_out.asm в тот же каталог, создаю копию файла lab5-1.asm (рис. 2.7).

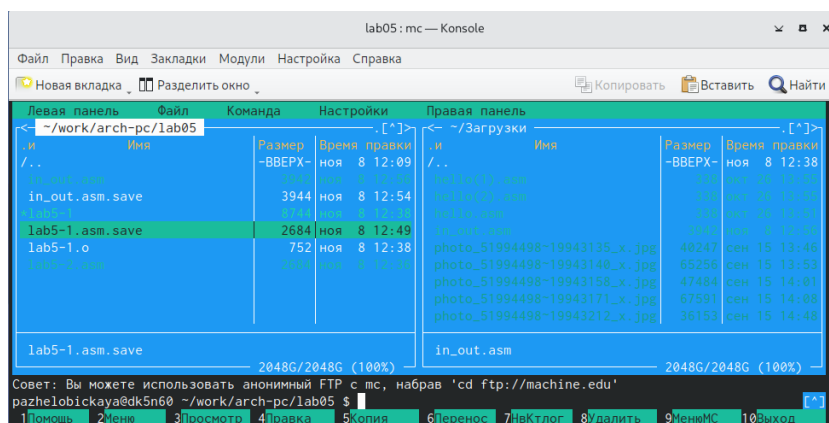


Рис. 2.7: Перемещение и копирование

Исправляю текст программы в файле lab5-2.asm с использованием подпрограмм из файла in_out.asm, заменяя sprintf на printf (рис. 2.8).

```

;-----
; 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. 2.8: Изменение текста программы

Создаю объектный файл lab5-2.o, выполняю компоновку объектного файла и запускаю исполняемый файл (рис. 2.9).

```

pazhelobickaya@dk8n63 ~ $ cd ~/work/arch-pc/lab05
pazhelobickaya@dk8n63 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
pazhelobickaya@dk8n63 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
pazhelobickaya@dk8n63 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку: Zhelobitskaya Polina
pazhelobickaya@dk8n63 ~/work/arch-pc/lab05 $

```

Рис. 2.9: Запуск файла

#Выполнение заданий для самостоятельной работы

Создаю копию файла lab5-1.asm (рис. 2.10).

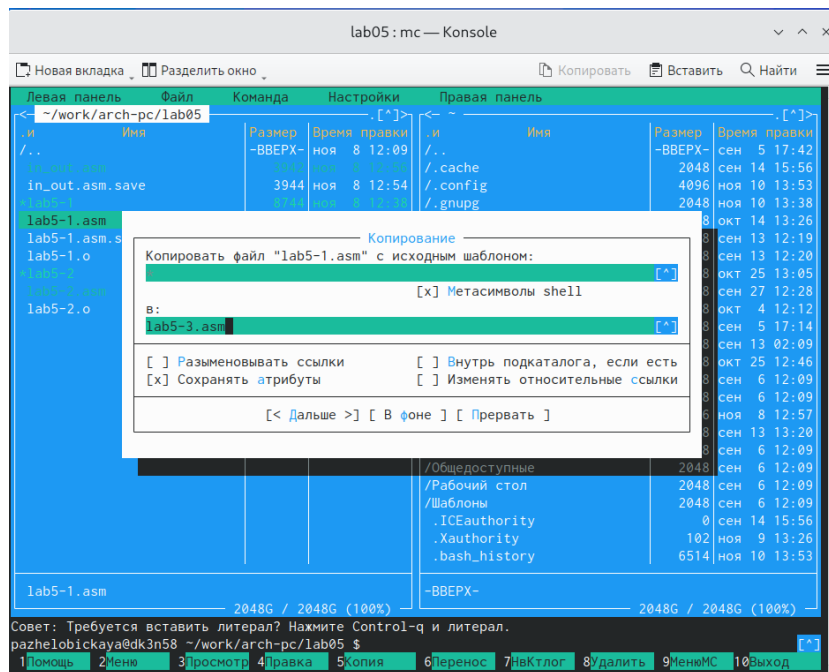


Рис. 2.10: Создание копии

Ввожу изменения в программу, чтобы она работала по следующему алгоритму:

- вывести приглашение типа “Введите строку:”;
- ввести строку с клавиатуры;
- вывести введенную строку на экран (рис. 2.11).

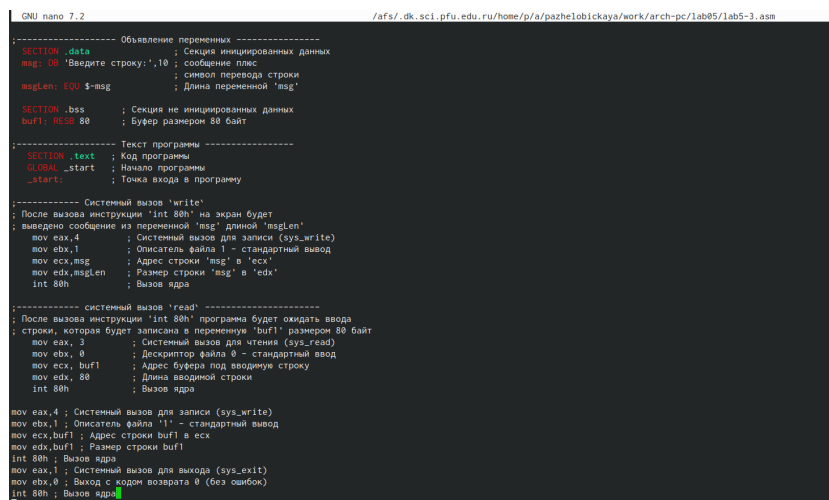


Рис. 2.11: Внесение изменений

Получаю исполняемый файл и проверяю его работу (рис. 2.12).

```
pazhelobickaya@dk3n58 ~/work/arch-pc/lab05 $ nasm -f elf lab5-3.asm
pazhelobickaya@dk3n58 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-3 lab5-3.o
pazhelobickaya@dk3n58 ~/work/arch-pc/lab05 $ ./lab5-3
Введите строку:
Zhelobitskaya
Zhelobitskaya
pazhelobickaya@dk3n58 ~/work/arch-pc/lab05 $
```

Рис. 2.12: Запуск файла

Создаю копию файла lab5-2.asm (рис. 2.13).

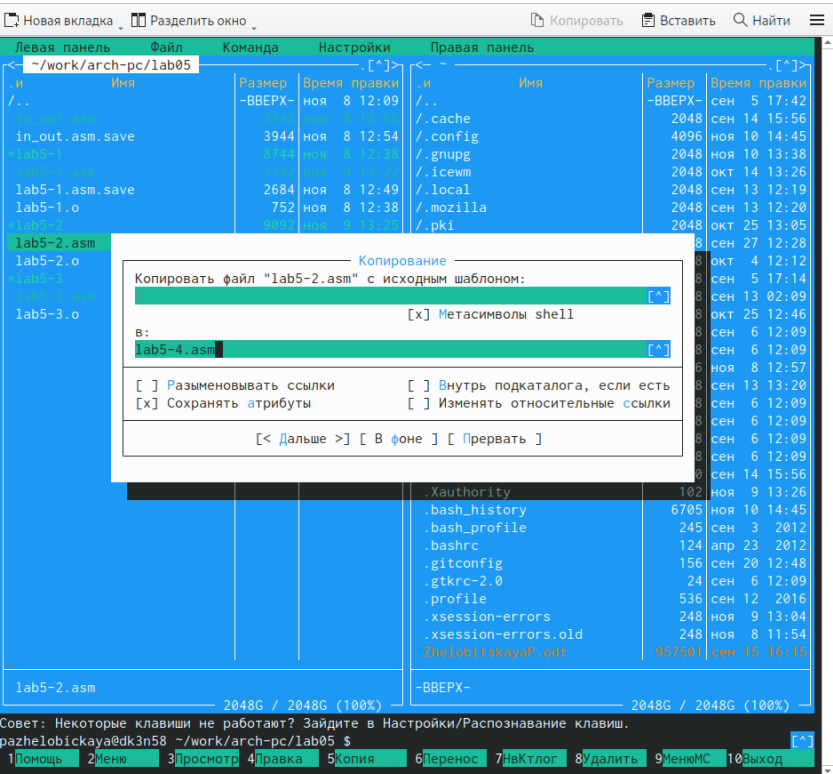


Рис. 2.13: Создание копии

Исправляю текст программы, чтобы она работала следующим образом: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введенную строку на экран (рис. 2.14).

```

GNU nano 7.2 /afs/.dk.sci.pfu.edu.ru/home/p/a/pazhelobickaya/work/arch-pc/lab05/lab5-4.asm
; Программа вывода сообщения на экран и ввода строки с клавиатуры
-----
%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
    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

```

Прочитано 30 строк

^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^U Отмена ^A Установить мет
 ^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^C Позиция ^E Повтор ^M-6 Копировать

Рис. 2.14: Изменение программы

Получаю исполняемый файл и проверяю его работу (рис. 2.15).

```

pazhelobickaya@dk3n58 ~ $ cd ~/work/arch-pc/lab05
pazhelobickaya@dk3n58 ~/work/arch-pc/lab05 $ nasm -f elf lab5-4.asm
pazhelobickaya@dk3n58 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-4 lab5-4.o
pazhelobickaya@dk3n58 ~/work/arch-pc/lab05 $ ./lab5-4
Введите строку: Zhelobitskaya
Zhelobitskaya
pazhelobickaya@dk3n58 ~/work/arch-pc/lab05 $

```

Рис. 2.15: Запуск файла

3 Выводы

Я приобрела практические навыки работы в Midnight Commander и освоила инструкции языка ассамблера `mov` и `int`.

Список литературы