

Отчёта по лабораторной работе

Основы работы с Midnight Commander(mc). Структура программы на языке ассемблера NASM.

Саакян Нерсес Варданович

Содержание

| | | |
|----------|--|-----------|
| 1 | Цель работы | 5 |
| 2 | Задание | 6 |
| 3 | Выполнение лабораторной работы | 7 |
| 4 | Выполнение самостоятельной работы | 14 |
| 5 | Выводы | 18 |
| | Список литературы | 19 |

Список иллюстраций

| | | |
|------|--|----|
| 3.1 | Создание каталога | 7 |
| 3.2 | Создание папки lab5 | 8 |
| 3.3 | Проверка | 8 |
| 3.4 | Создание файла lab5-1.asm | 9 |
| 3.5 | Ввод текста | 9 |
| 3.6 | Проверка текста | 10 |
| 3.7 | Трансляция текста | 10 |
| 3.8 | Копирование | 11 |
| 3.9 | Создание копии файла lab5- 1.asm с именем lab5-2.asm | 11 |
| 3.10 | Исправление текста | 12 |
| 3.11 | Проверка текста | 12 |
| 3.12 | Транслировка текста | 12 |
| 3.13 | Исправление текста | 13 |
| 3.14 | Транслировка текста | 13 |
| 4.1 | Копирование файла | 14 |
| 4.2 | Изменение программы | 14 |
| 4.3 | Запуск программы | 15 |
| 4.4 | Копирование файла | 16 |
| 4.5 | Изменение программы | 16 |
| 4.6 | Запуск программы | 17 |

Список таблиц

1 Цель работы

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

2 Задание

1 Создайте копию файла lab6-1.asm. Внесите изменения в программу (без использования внешнего файла in_out.asm), так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введённую строку на экран. 2 Получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите свою фамилию. 3 Создайте копию файла lab6-2.asm. Исправьте текст программы с использование подпрограмм из внешнего файла in_out.asm, так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введённую строку на экран.

3 Выполнение лабораторной работы

Создайте каталог для работы с программами на языке ассемблера NASM. (рис. 4.1).

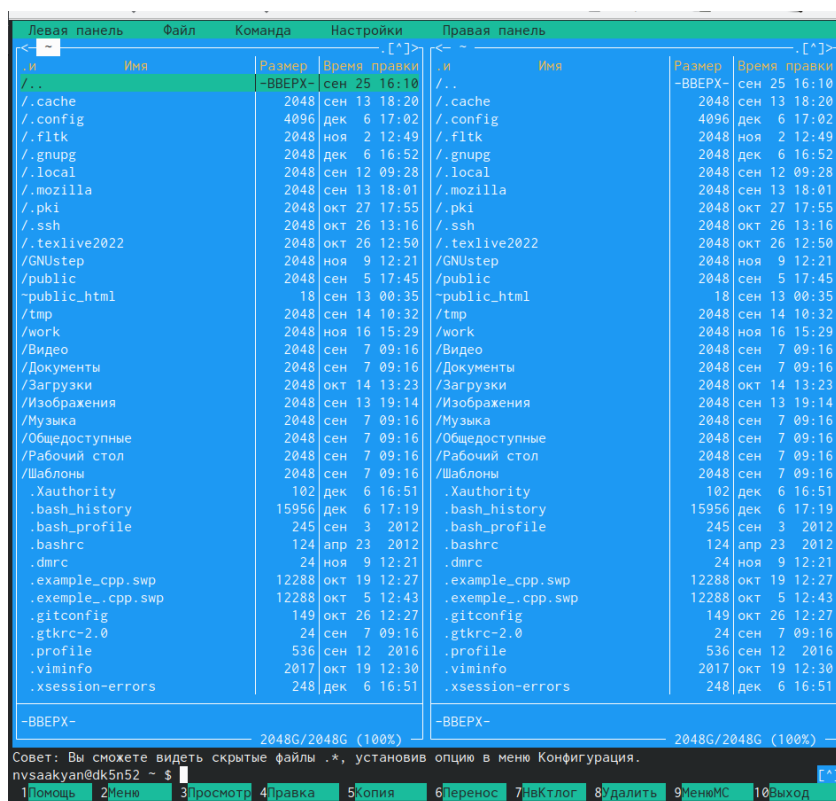


Рис. 3.1: Создание каталога

С помощью функциональной клавиши F7 создаём папку lab05. (рис. 4.2).

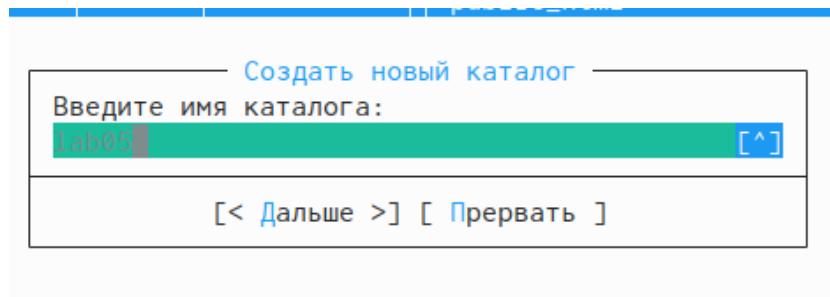


Рис. 3.2: Создание папки lab5

Убедимся в правильном создании папки. (рис. 4.3).

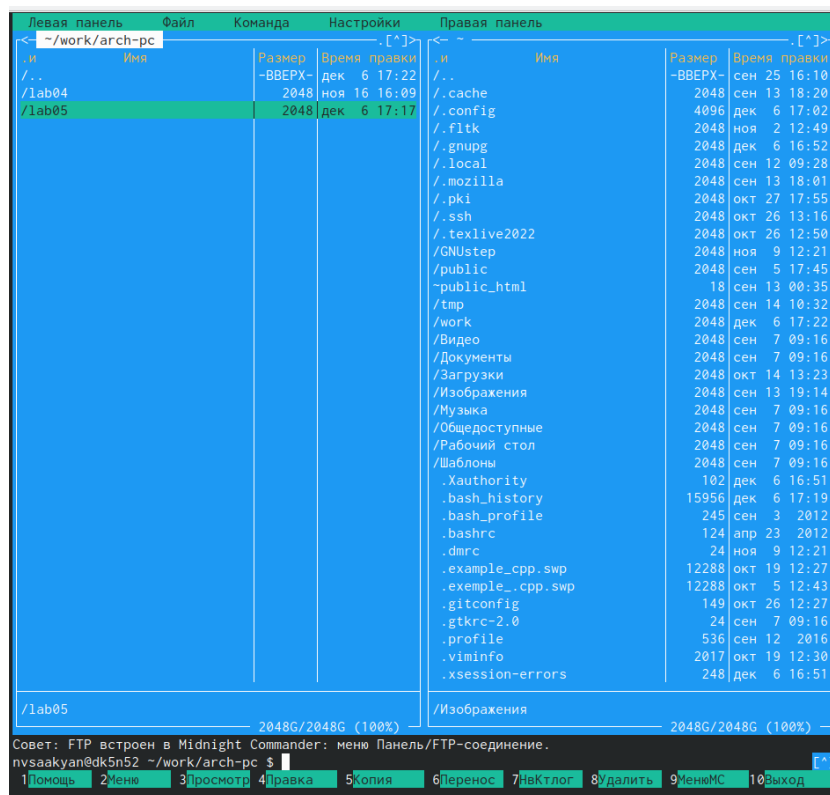


Рис. 3.3: Проверка

Пользуясь строкой ввода и командой touch создадим файл lab5-1.asm. (рис. 4.4).

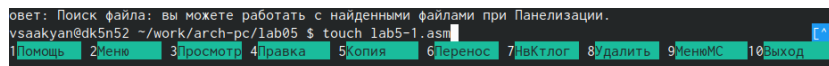


Рис. 3.4: Создание файла lab5-1.asm

С помощью функциональной клавиши F4 откроем файл lab5-1.asm и введём текст из листинга 5.1. (рис. 4.5).

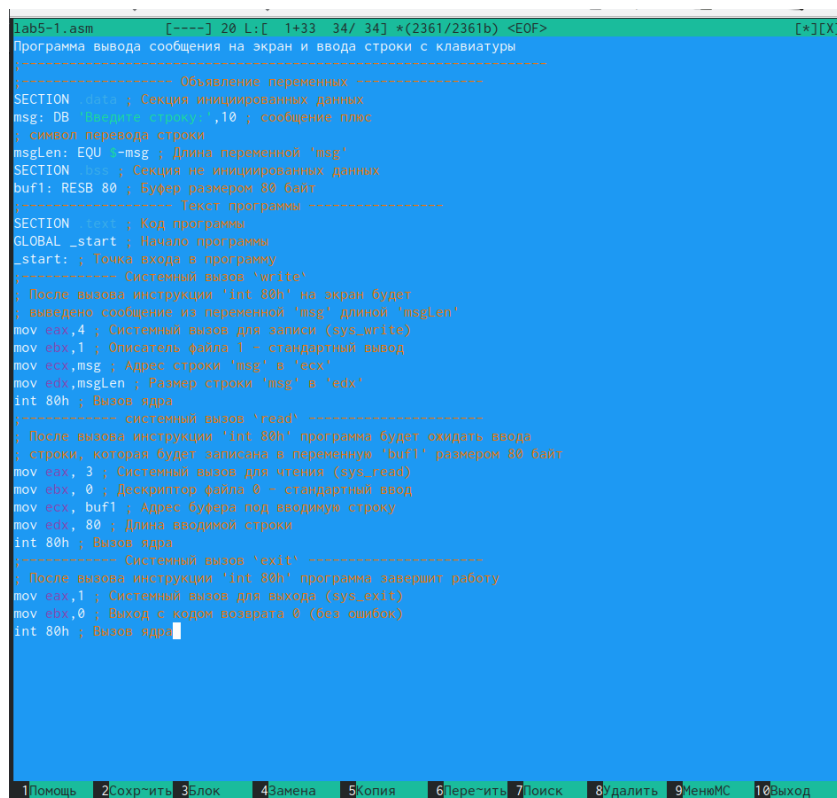


Рис. 3.5: Ввод текста

С помощью функциональной клавиши F3 откроем файл lab5-1.asm для проверки наличия текста. (рис. 4.6).

```

/afs/dk.sc1.pfu.edu.ru/home/n/v/nv-kyan/work/arch-pc/lab05/lab5-1.asm 2361/2361 100%
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
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

```

Рис. 3.6: Проверка текста

Оттранслируем текст программы lab5-1.asm в объектный файл. (рис. 3.7).

```

nvsaakyan@dk5n52 ~/work/arch-pc/lab05 $
nvsaakyan@dk5n52 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.asm
nvsaakyan@dk5n52 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o
nvsaakyan@dk5n52 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку:
Saakyan Nerses
nvsaakyan@dk5n52 ~/work/arch-pc/lab05 $

```

Рис. 3.7: Трансляция текста

Скопируем файл in_out.asm в каталог с файлом lab5-1.asm с помощью функциональной клавиши F5. (рис. 3.8).

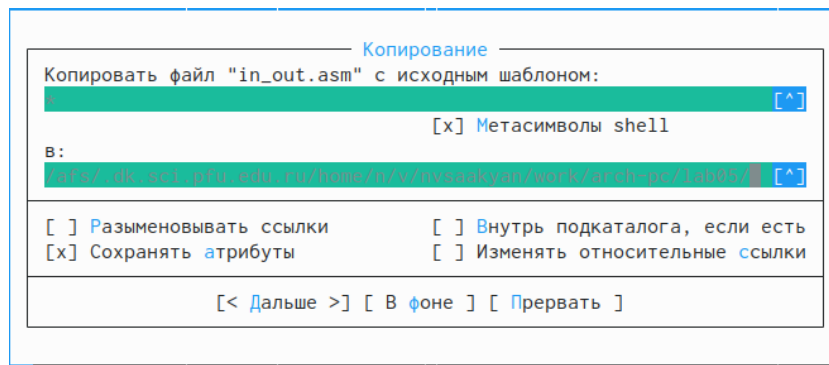


Рис. 3.8: Копирование

С помощью функциональной клавиши F6 создадим копию файла lab5- 1.asm с именем lab5-2.asm. (рис. 3.9).

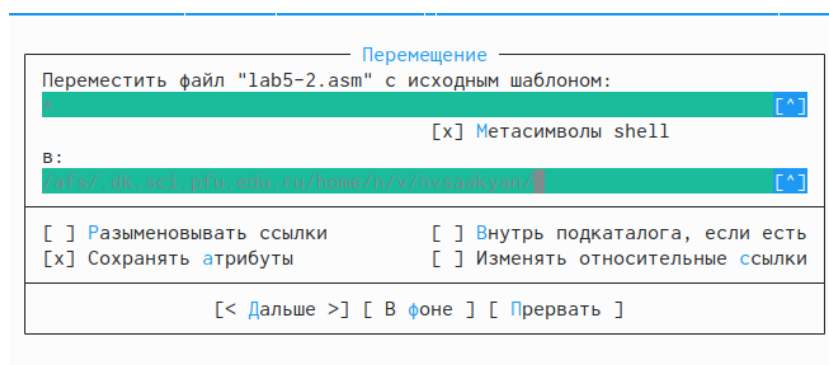


Рис. 3.9: Создание копии файла lab5- 1.asm с именем lab5-2.asm

Исправим текст программы в соответствии с листингом 5.2 (рис. 3.10).

```

lab5-2.asm [----] 6 L: [ 1+20 21/ 23] *(1236/1396b) 0120 0x070
; Программа вывода сообщения на экран и ввода строки с клавиатуры
; Подключение внешнего файла
%include 'in_out.asm'
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 sprintf ; вызов подпрограммы печати сообщения

mov ecx, buf1 ; запись адреса переменной в 'ECX'
mov edx, 80 ; запись длины вводимого сообщения в 'EDX'
call sread ; вызов подпрограммы ввода сообщения

mov eax, buf1 ; запись адреса строки в 'EAX'
mov edx, 80 ; запись длины выводимого сообщения в 'EDX'
call sprintf ; вызов подпрограммы печати сообщения
call quit ; вызов подпрограммы завершения

```

Рис. 3.10: Исправление текста

Проверим, что текст был изменен (рис. 3.11).

```

GNU nano 6.4 /afs/.dk.sci.pfu.edu.ru/home/n/v/nvsaakyan/work/arch-pc/lab05/lab5-2.asm Изменён
; Программа вывода сообщения на экран и ввода строки с клавиатуры
; Подключение внешнего файла
%include 'in_out.asm'
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 sprintf ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса переменной в 'ECX'
mov edx, 80 ; запись длины вводимого сообщения в 'EDX'
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения

```

Рис. 3.11: Проверка текста

Оттранслируем текст программы lab5-2.asm в объектный файл и проверим его работоспособность (рис. 3.12).

```

nvsaakyan@dk5n52 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
nvsaakyan@dk5n52 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
nvsaakyan@dk5n52 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку:
123
nvsaakyan@dk5n52 ~/work/arch-pc/lab05 $

```

Рис. 3.12: Транслировка текста

- вывести приглашение типа “Введите строку:”
- ввести строку с клавиатуры;
- вывести введенную строку на экран. (рис. 3.13).

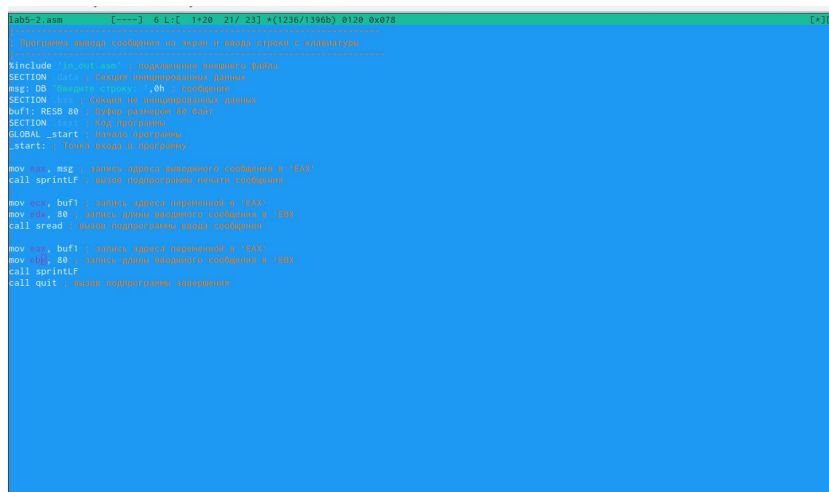


Рис. 3.13: Исправление теста

Оттранслируем текст программы в объектный файл и внесём изменения в текст программы в файле lab5.asm (рис. 3.14).

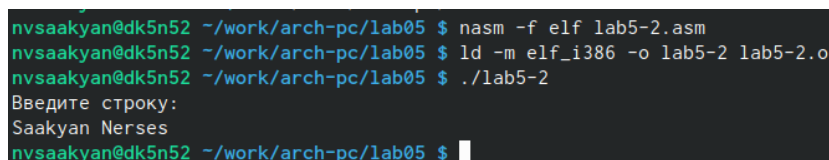


Рис. 3.14: Транслировка текста

4 Выполнение самостоятельной работы

Коопирую файл lab5-1.asm с именем lab5-1.1asm (рис. 4.1).

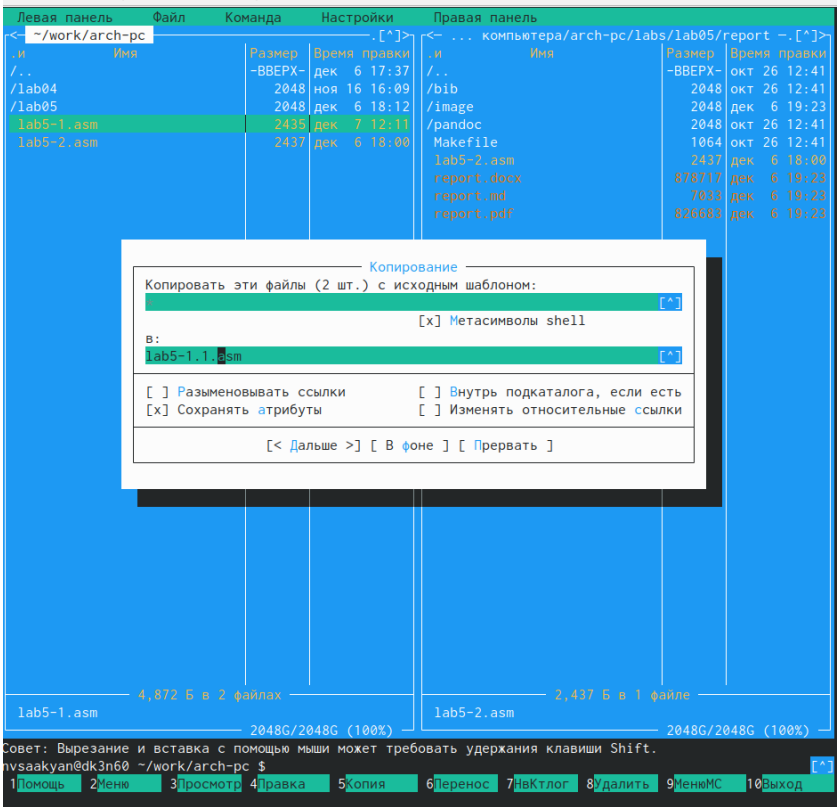


Рис. 4.1: Коопирование файла

Изменяю код программы, добавляя вывод введенной строки (рис. 4.2).

Изменение программы

Рис. 4.2: Изменение программы

Создаю объектный файл lab5-1.1.o, componую его в исполняемый файл, запускаю исполняемый файл (рис. 4.3).

```
nvsaaayan@dk3n60 ~/work/arch-pc $ nasm -f elf lab5-1.1.asm
nvsaaayan@dk3n60 ~/work/arch-pc $ ld -m elf_i386 -o lab5-1.1.asm lab5-1.o
ld: невозможно найти lab5-1.o: Нет такого файла или каталога
nvsaaayan@dk3n60 ~/work/arch-pc $ ld -m elf_i386 -o lab5-1.1. lab5-1.1.o
nvsaaayan@dk3n60 ~/work/arch-pc $ ./lab5-1.1
bash: ./lab5-1.1: Нет такого файла или каталога
nvsaaayan@dk3n60 ~/work/arch-pc $ ./lab5-1.1.
Введите строку:
Saakyan Nerses
Saakyan Nerses
```

Рис. 4.3: Запуск программы

Программа из пункта 1: ;—————- 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 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 Копирую файл lab5-2.asm с
именем lab5-2.2.asm (рис. 4.4).

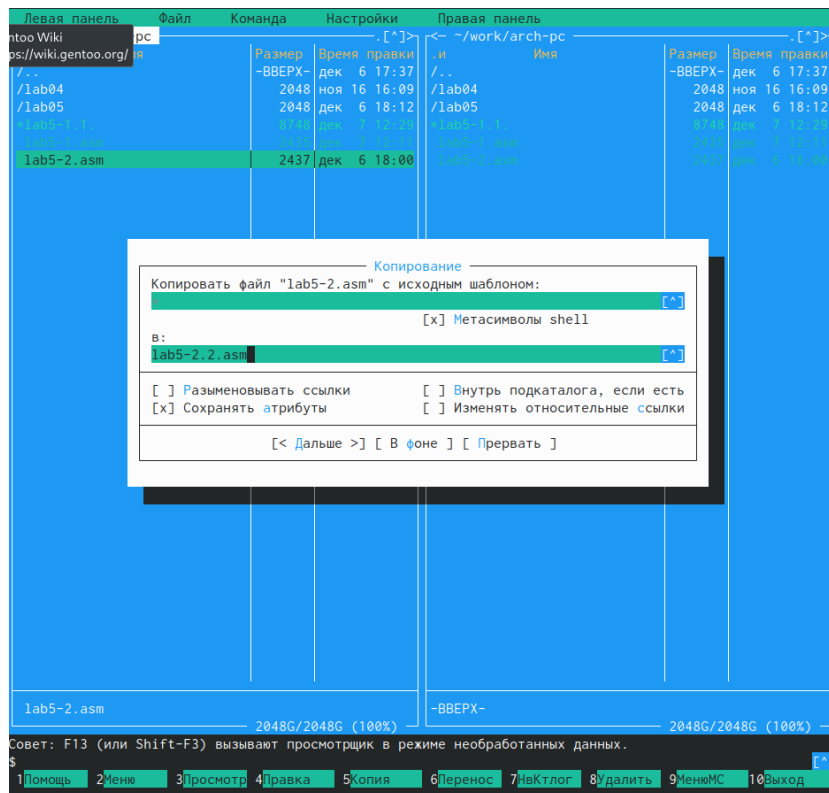
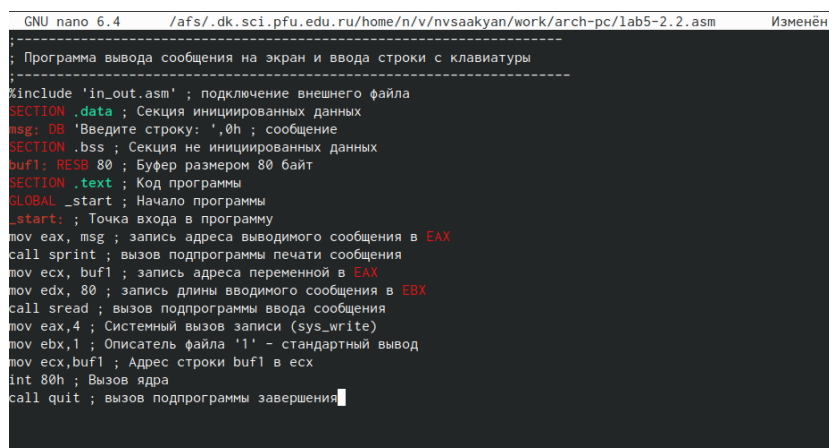


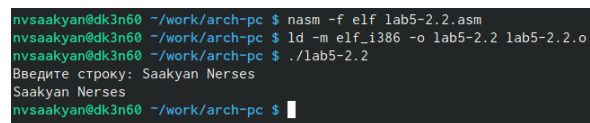
Рис. 4.4: Копирование файла

Изменяю код программы, добавляя вывод введенной строки (рис. 4.5).



Создаю объектный файл lab5-2.2.o, компоную его в исполняемый файл, запус-

каю исполняемый файл (рис. 4.6).



```
nvsaakyan@dk3n60 ~/work/arch-pc $ nasm -f elf lab5-2.2.asm
nvsaakyan@dk3n60 ~/work/arch-pc $ ld -m elf_i386 -o lab5-2.2 lab5-2.2.o
nvsaakyan@dk3n60 ~/work/arch-pc $ ./lab5-2.2
Введите строку: Saakyan Nerses
Saakyan Nerses
nvsaakyan@dk3n60 ~/work/arch-pc $
```

Рис. 4.6: Запуск программы

5 Выводы

В ходе лабораторной работы мною были приобретены практические навыки работы в Midnight Commander, а также освоены инструкции языка ассемблера `mov` и `int`. Я научился работать с МС, и с его помощью работать с файлами (Создание, переименовывание, копирование, перемещение, удаление, и тд.)

Список литературы