

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5

дисциплина: Архитектура компьютера

Студент: Тимаков Макарий Владимирович

Группа: НПИбд 02-25

МОСКВА

2025г.

1. Цель работы

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

2. Выполнение лабораторной работы

Открываем Midnight Commander (Рисунок 2.1).

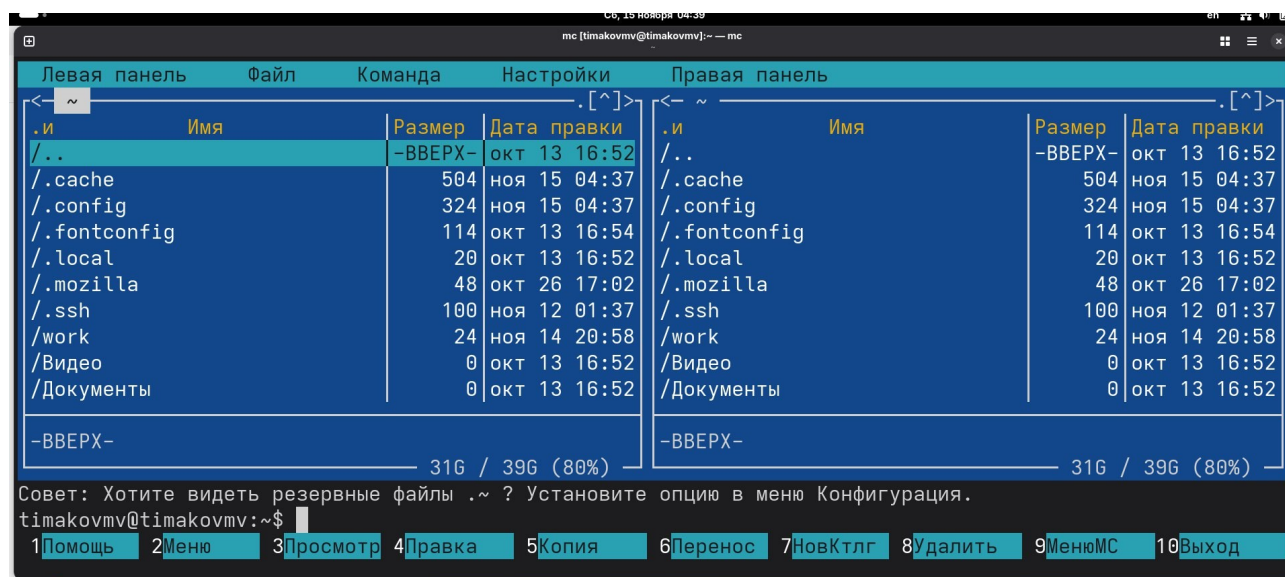


Рисунок 2.1: Открываем с помощью команды `mc`

Переходим в каталог, созданный при выполнении 4 ЛБ (Рисунок 2.2).

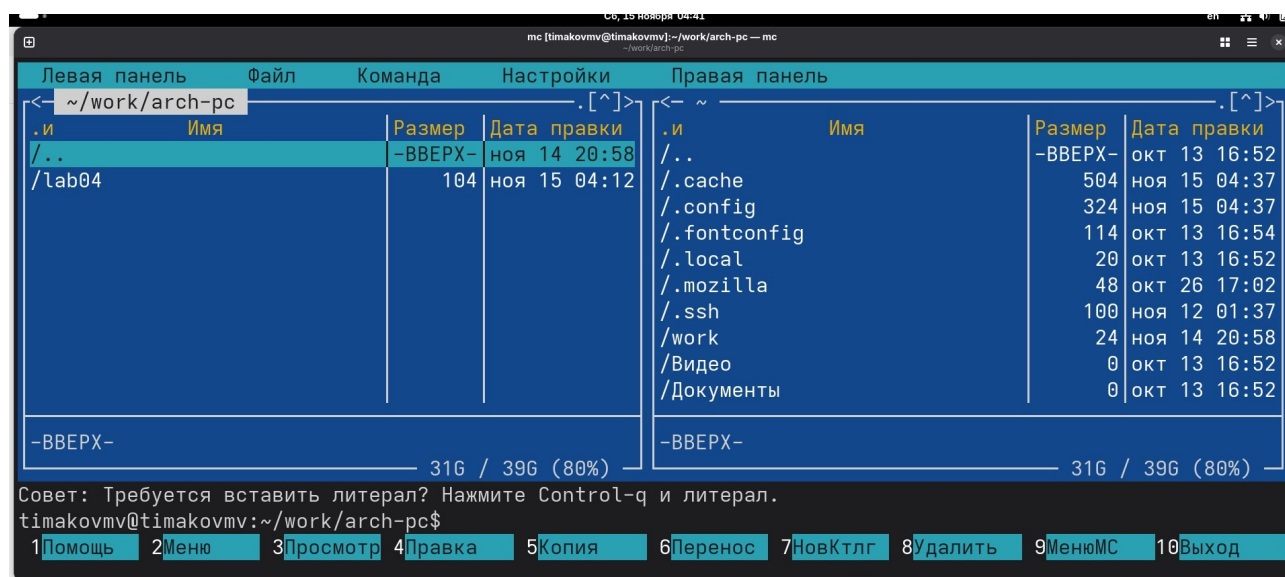


Рисунок 2.2: Переходим в каталог

Создаем каталог lab05 (Рисунок 2.3).

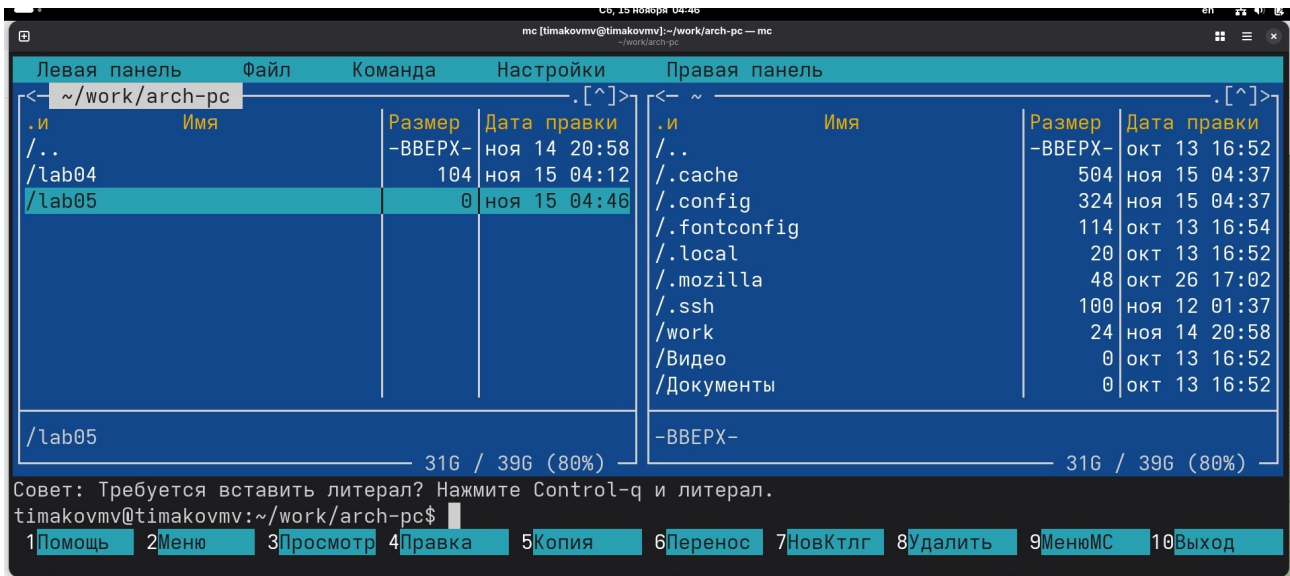


Рисунок 2.3: Создаем каталог функциональной клавишей

Создаем файл lab5-1.asm (Рисунок 2.4)

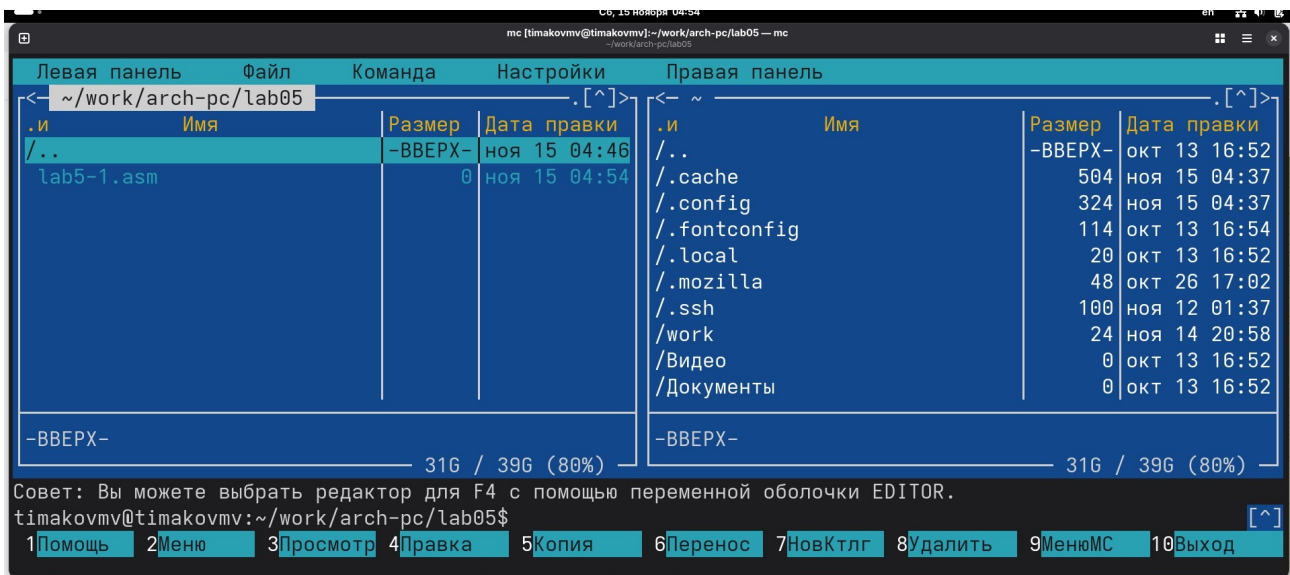


Рисунок 2.4: Создаем файл с помощью команды touch

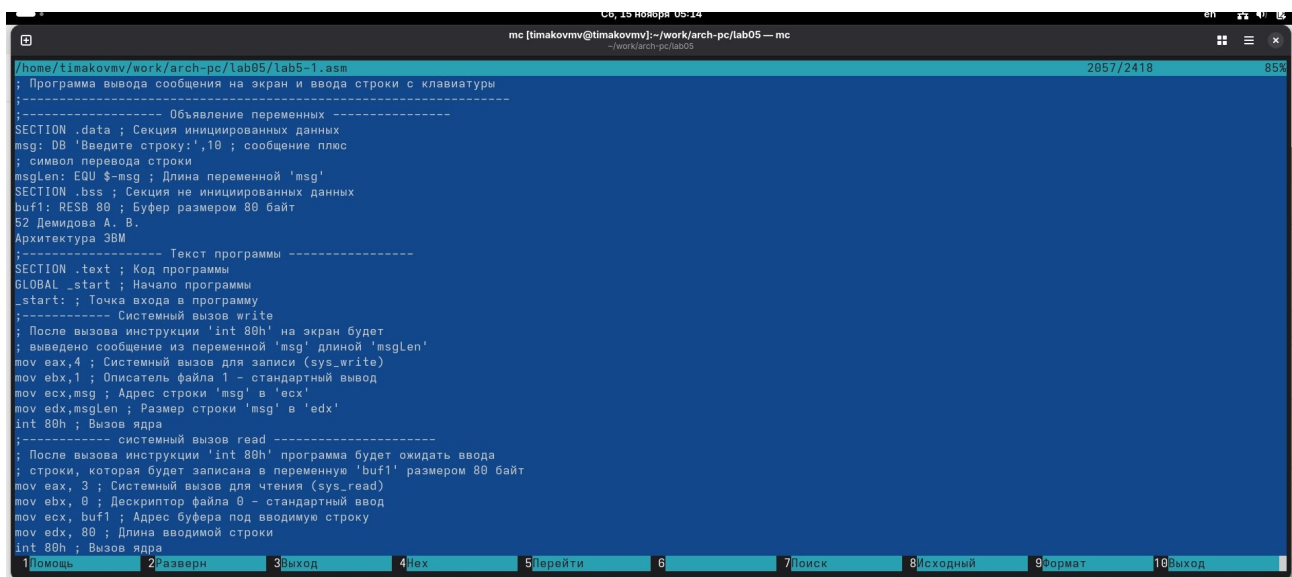
Открываем файл для редактирования и заполняем его по листингу (Рисунок 2.5)



```
GNU nano 8.3 /home/timakovmv/work/arch-pc/lab05/lab5-1.asm
; 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
52 Демидова А. В.
Архитектура 386
;----- 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)
```

Рисунок 2.5: Открываем файл функциональной клавишей, заполняем и сохраняем

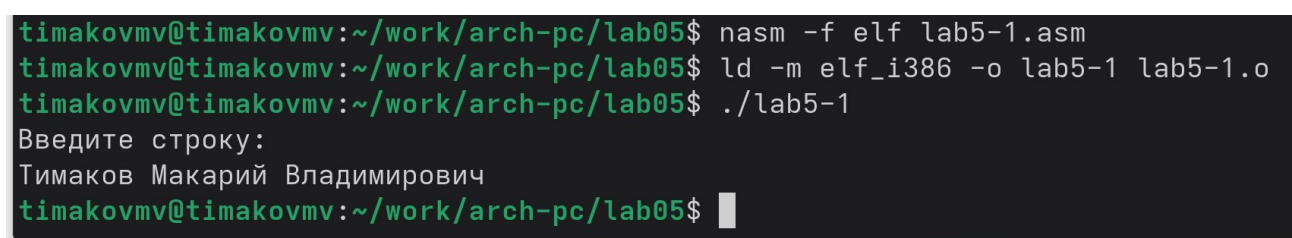
Открываем файл для просмотра (Рисунок 2.6).



```
/home/timakovmv/work/arch-pc/lab05/lab5-1.asm 2057/2418 85%
; 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
52 Демидова А. В.
Архитектура 386
;----- 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
1Помощь 2Разверн 3Выход 4Лекс 5Перейти 6 7Поиск 8Исходный 9Формат 10Выход
```

Рисунок 2.6: Открываем файл и убеждаемся, что файл содержит текст программы

Транслируем текст программы и запускаем исполняемый файл (Рисунок 2.7).



```
timakovmv@timakovmv:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
timakovmv@timakovmv:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
timakovmv@timakovmv:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Тимаков Макарий Владимирович
timakovmv@timakovmv:~/work/arch-pc/lab05$
```

Рисунок 2.7: Проверяем, как работает данная программа

Скачиваем файл со страницы курса (Рисунок 2.8).

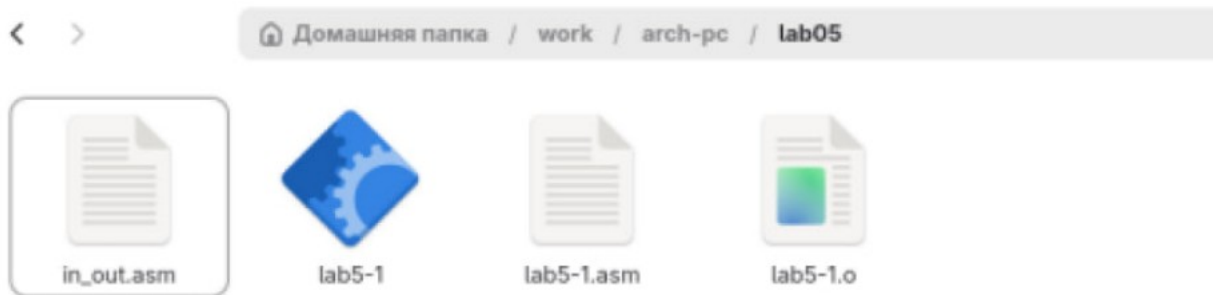


Рисунок 2.8: Скачиваем файл

Копируем файл в нужную директорию (Рисунок 2.9).

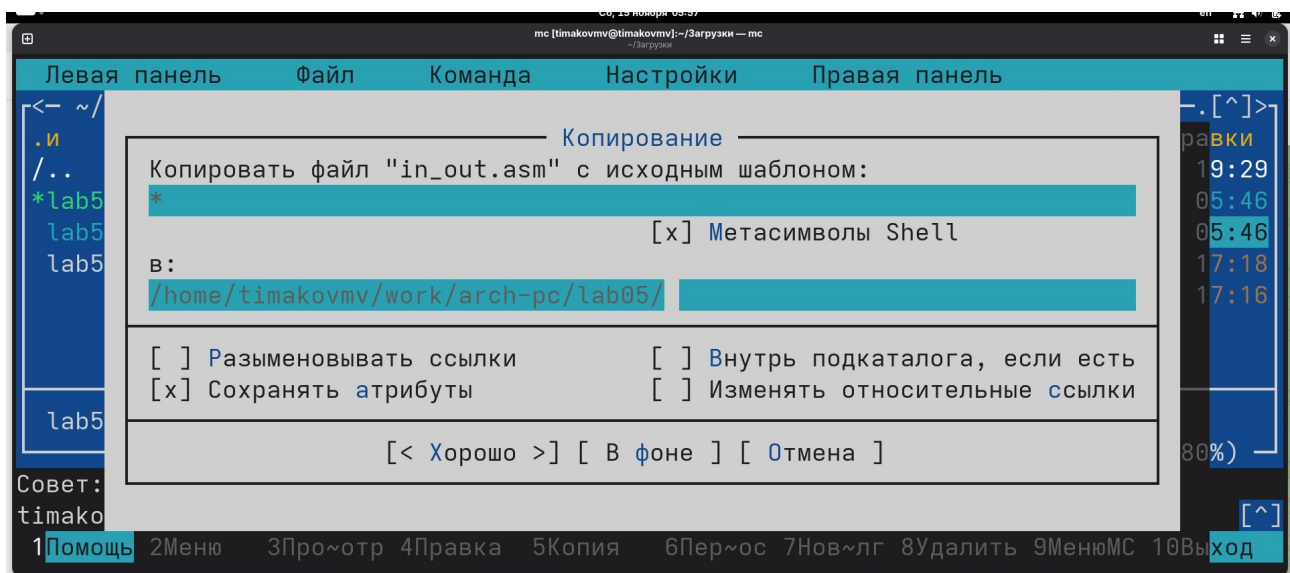


Рисунок 2.9: Копируем скаченный файл

Создаем копию файла lab5-1.asm и называем lab5-2.asm (Рисунок 2.10).

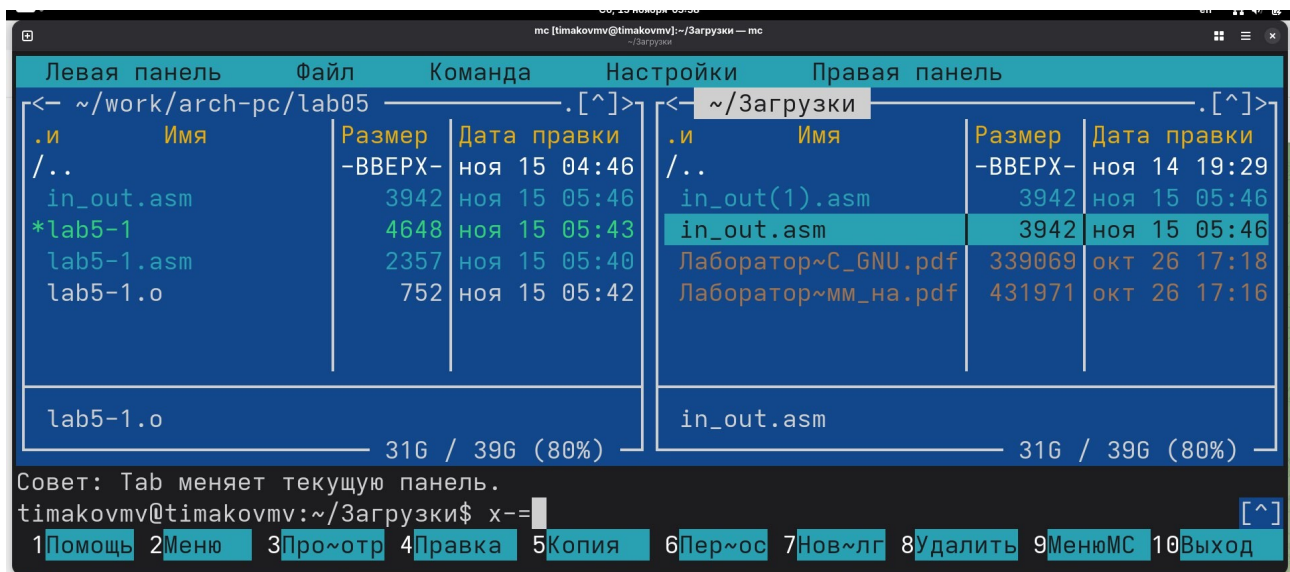


Рисунок 2.10: Создаем копию файла клавишей с помощью функциональной клавиши и проверяем

Открываем новый файл и заполняем его в соответствии с листингом (Рисунок 2.11).

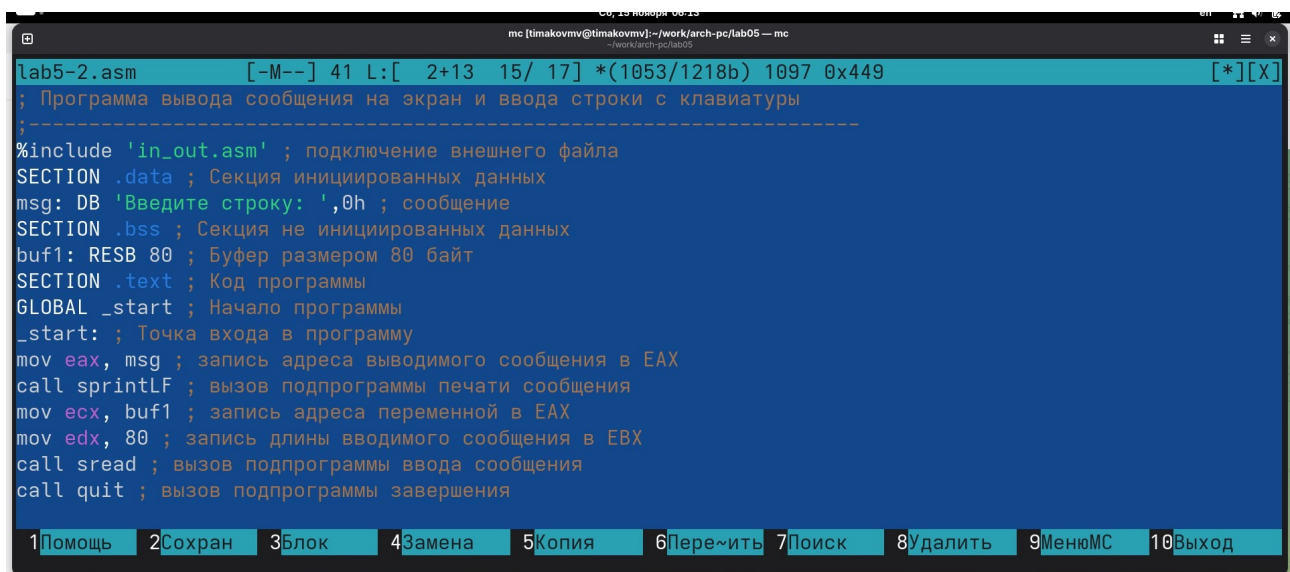


Рисунок 2.11: Открывем файл функциональной клавишей, заполняем и сохраняем

Транслируем и запускаем новый файл (Рисунок 2.12).

```
timakovmv@timakovmv:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
timakovmv@timakovmv:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
timakovmv@timakovmv:~/work/arch-pc/lab05$ ./lab6-2
bash: ./lab6-2: Нет такого файла или каталога
timakovmv@timakovmv:~/work/arch-pc/lab05$ ls
in_out.asm  lab5-1  lab5-1.asm  lab5-1.o  lab5-2  lab5-2.asm  lab5-2.o
timakovmv@timakovmv:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Тимаков Макарий Владимирович
timakovmv@timakovmv:~/work/arch-pc/lab05$ ~
```

Рисунок 2.12: Смотрим, как сработала программа

Снова открываем файл для редактирования и меняем `sprintLF` на `sprint` (Рисунок 2.13).

```
mov eax, msg ; запись адреса выводимого сообщения в EAX
call sprint  ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса переменной в EAX
```

Рисунок 2.13: Редактируем файл и сохраняем

Транслируем и запускаем файл (Рисунок 2.14).

```
timakovmv@timakovmv:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
timakovmv@timakovmv:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
timakovmv@timakovmv:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Тимаков Макарий Владимирович
timakovmv@timakovmv:~/work/arch-pc/lab05$ ~
```

Рисунок 2.14: Смотрим, как сработала программа и сравниваем с прошлой

Таким образом, команда `sprint` выводит текст в той же строке, а `sprintLF` выполняет переход на новую строку.

3. Задание для самостоятельной работы

Создаем копию файла lab5-1.asm (Рисунок 3.1).

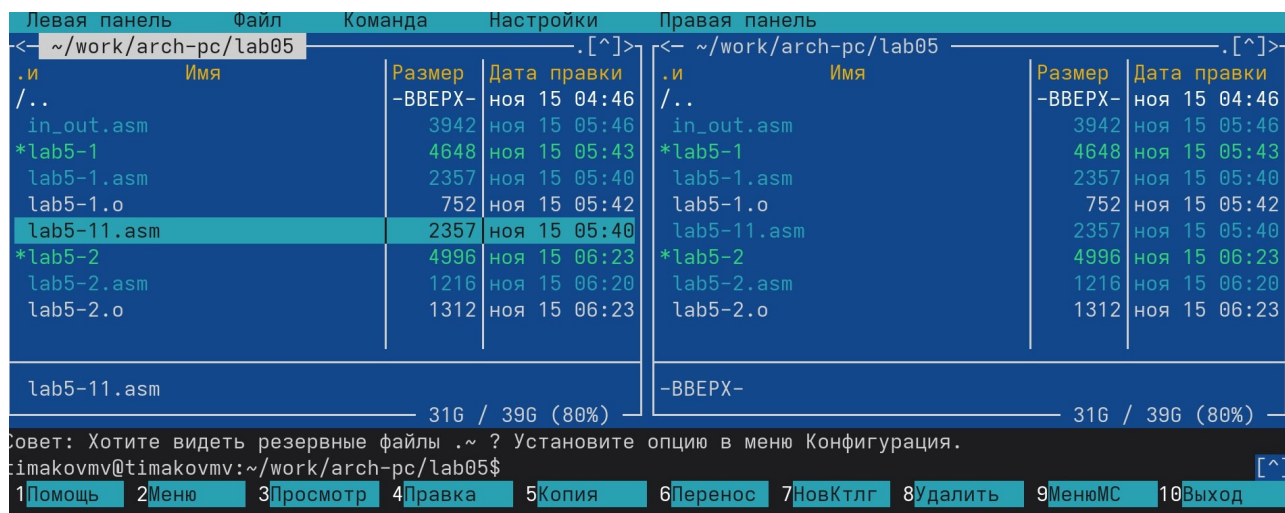


Рисунок 3.1: Создаем копию файла с помощью функциональной клавиши

Редактируем файл, чтобы введенный текст с клавиатуры выводился в консоль (Рисунок 3.2).

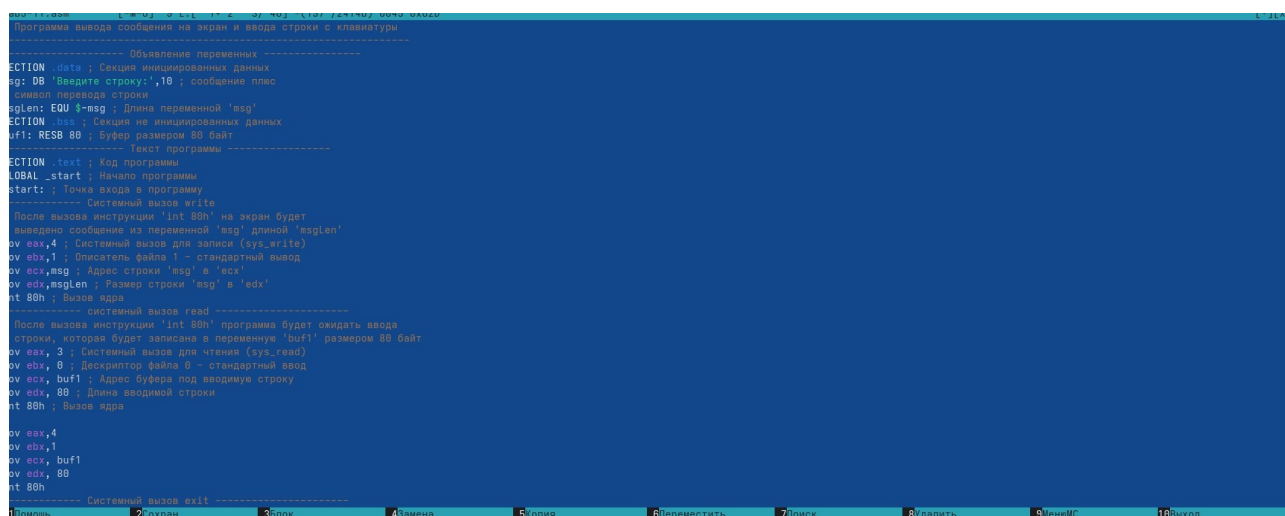


Рисунок 3.2: Редактируем файл

Транслируем файл и запускаем программу (Рисунок 3.3).

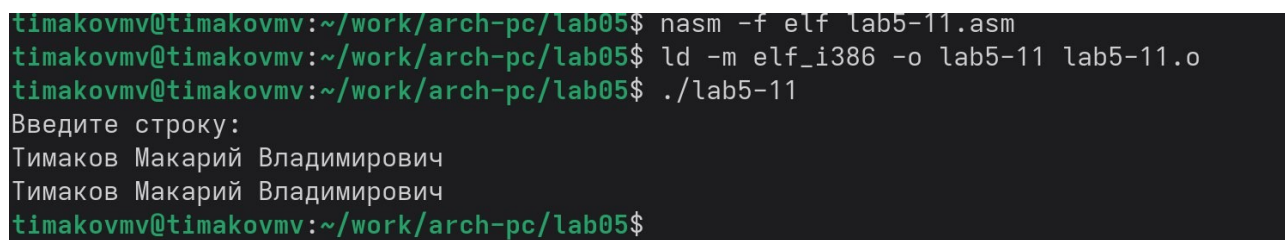


Рисунок 3.3: Смотрим, как сработала программа

Создаем копию файла lab5-2.asm (Рисунок 3.4).

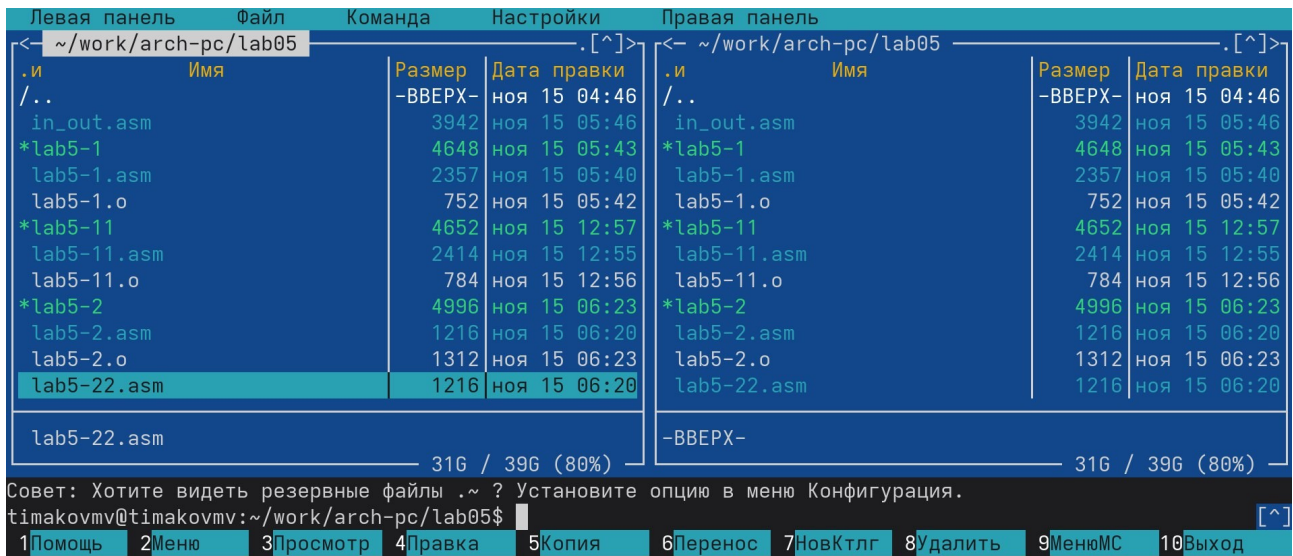


Рисунок 3.4: Создаем копию файла с помощью функциональной клавиши

Редактируем файл, чтобы введенный текст с клавиатуры выводился в консоль (Рисунок 3.5).

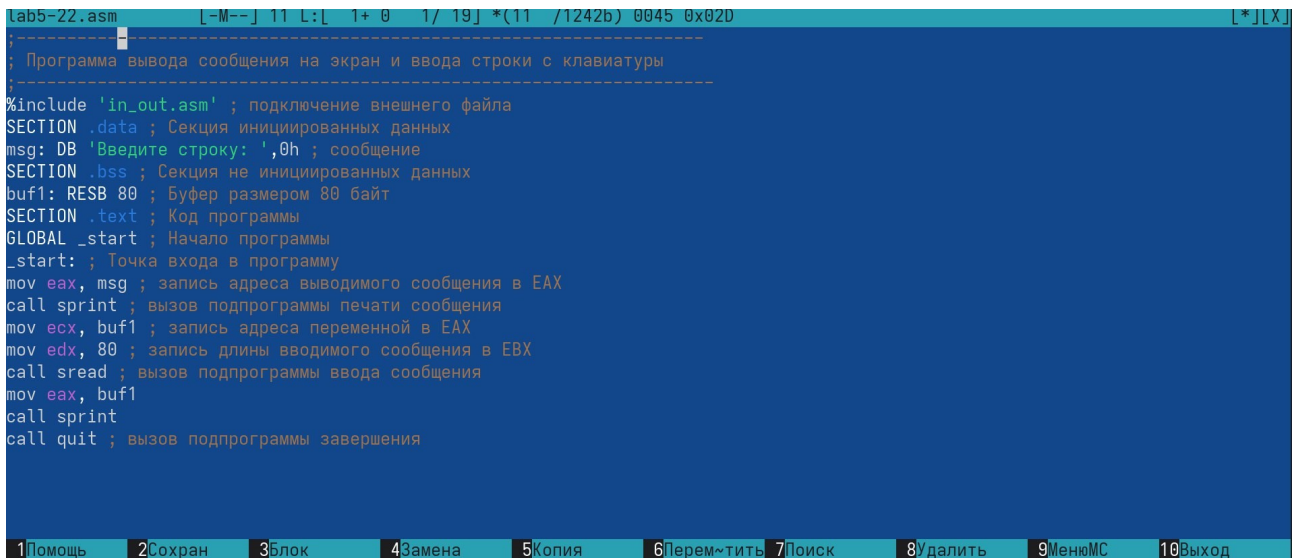
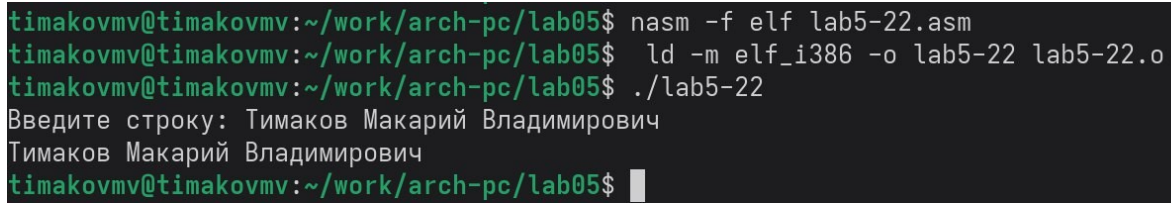


Рисунок 3.5: Редактируем файл

Транслируем файл и запускаем программу (Рисунок 3.6).

A terminal window with a dark background and green text. The prompt is 'timakovmv@timakovmv:~/work/arch-pc/lab05\$'. The user enters 'nasm -f elf lab5-22.asm', then 'ld -m elf_i386 -o lab5-22 lab5-22.o', and finally './lab5-22'. The program outputs 'Введите строку: Тимаков Макарий Владимирович' and then 'Тимаков Макарий Владимирович' before returning to the prompt.

```
timakovmv@timakovmv:~/work/arch-pc/lab05$ nasm -f elf lab5-22.asm
timakovmv@timakovmv:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-22 lab5-22.o
timakovmv@timakovmv:~/work/arch-pc/lab05$ ./lab5-22
Введите строку: Тимаков Макарий Владимирович
Тимаков Макарий Владимирович
timakovmv@timakovmv:~/work/arch-pc/lab05$
```

Рисунок 3.6: Смотрим, как сработала программа

4. Выводы

Мы приобрели практические навыки работы в файловом менеджере Midnight Commander, а также освоили инструкции языка ассемблера `mov` и `int`.