

Российский университет дружбы народов
Факультет физико-математически и естественных наук

Отчёт
по лабораторной работе №5

по дисциплине:
архитектура компьютеров и операционные
системы

Студент: Аманов Рустам Марксович: НКАбд 01-23

№ ст. Билета: 1032234130

МОСКВА
2023 Г

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	8
4	Выполнение лабораторной работы	9
5	Выполнение заданий для самостоятельной работы	24
6	Выводы	28
	Список литературы	29

Список иллюстраций

4.1	1. Midnight Commander.....	Ошибка!
	Закладка не определена.	
4.2	2. Переход в work.....	9
4.3	3. Переход в arch-rc.....	10
4.4	4. Создание папки lab05	11
4.5	5. lab05.....	12
4.6	6. Созданный файл	13
4.7	7. Открытие файла в папо	14
4.8	7.1 Просмотр файла.....	14
4.9	8. Оттранслирую файл	15
4.10	9. Компоную файл	15
4.11	10. Работа файла	16
4.12	11.два окна.....	16
4.13	12. Копирование файла	17
4.14	13. Копирование файла	18
4.15	14. Скопированный файл.....	18
4.16	15. Редактирование файла	19
4.17	16. Созданный объектный файл.....	20

4.18 17. Создание и работы исполняемого файла.....	20
4.19 18. Редактирование файла	21
4.20 19. Работа файла	21
5.1 20. Копирование файла	22
5.2 21. Открытый файл	23
5.3 22. Код исправленной программы	25
5.4 23. Работа файла	26

Список таблиц

1.Цель работы

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

2.Задание

1. Откройте Midnight Commander user@dk4n31:~\$ mc
2. Пользуясь клавишами , и Enter перейдите в каталог ~/work/arch-рс созданный при выполнении лабораторной работы №4.
3. С помощью функциональной клавиши F7 создайте папку lab05 и перейдите в созданный каталог.
4. Пользуясь строкой ввода и командой touch создайте файл lab5-1.asm.
5. С помощью функциональной клавиши F4 откройте файл lab5-1.asm для редактирования во встроенном редакторе. Как правило в качестве встроенного редактора Midnight Commander используется редакторы nano или mcedit.
6. Введите текст программы из листинга 5.1 (можно без комментариев), сохраните изменения и закройте файл.
7. С помощью функциональной клавиши F3 откройте файл lab5-1.asm для просмотра.Убедитесь, что файл содержит текст программы.
8. Оттранслируйте текст программы lab5-1.asm в объектный файл.Выполните компоновку объектного файла и запустите получившийся исполняемый файл. выводит строку 'Введите строку:' и ожидает ввода с клавиатуры. На запрос введите Ваши ФИО.
9. Скачайте файл in_out.asm со страницы курса в ТУИС.
10. Подключаемый файл in_out.asm должен лежать в том же каталоге, что и файл с программой, в которой он используется.
11. С помощью функциональной клавиши F6 создайте копию файла lab5-1.asm с именем lab5-2.asm. Выделите файл lab5-1.asm, нажмите клавишу F6 , введите имя файла lab5-2.asm и нажмите клавишу Enter
12. Исправьте текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in_out.asm (используйте подпрограммы sprintLF, sread и quit) в

соответствии с листингом 5.2. Создайте исполняемый файл и проверьте его работу.

3. Теоретическое введение

Midnight Commander (или просто mc) — это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управлению файловой системой, т.е. mc является файловым менеджером. Midnight Commander позволяет сделать работу с файлами более удобной и наглядной. Для активации оболочки Midnight Commander достаточно ввести в командной строке mc и нажать клавишу Enter (рис. 5.1). В Midnight Commander используются функциональные клавиши F1—F10, к которым привязаны часто выполняемые операции (табл. 5.1).

Функциональные клавиши	Выполняемое действие
F1	вызов контекстно-зависимой подсказки
F2	вызов меню, созданного пользователем
F3	просмотр файла, на который указывает подсветка в активной панели
F4	вызов встроенного редактора для файла, на который указывает подсветка в активной панели
F5	копирование файла или группы отмеченных файлов из каталога, отображаемого в активной панели, в каталог, отображаемый на второй панели
F6	перенос файла или группы отмеченных файлов из каталога, отображаемого в активной панели, в каталог, отображаемый на второй панели
F7	создание подкаталога в каталоге, отображаемом в активной панели
F8	удаление файла (подкаталога) или группы отмеченных файлов
F9	вызов основного меню программы
F10	выход из программы

4. Выполнение лабораторной работы

13. Открываю Midnight Commander с помощью команды mc

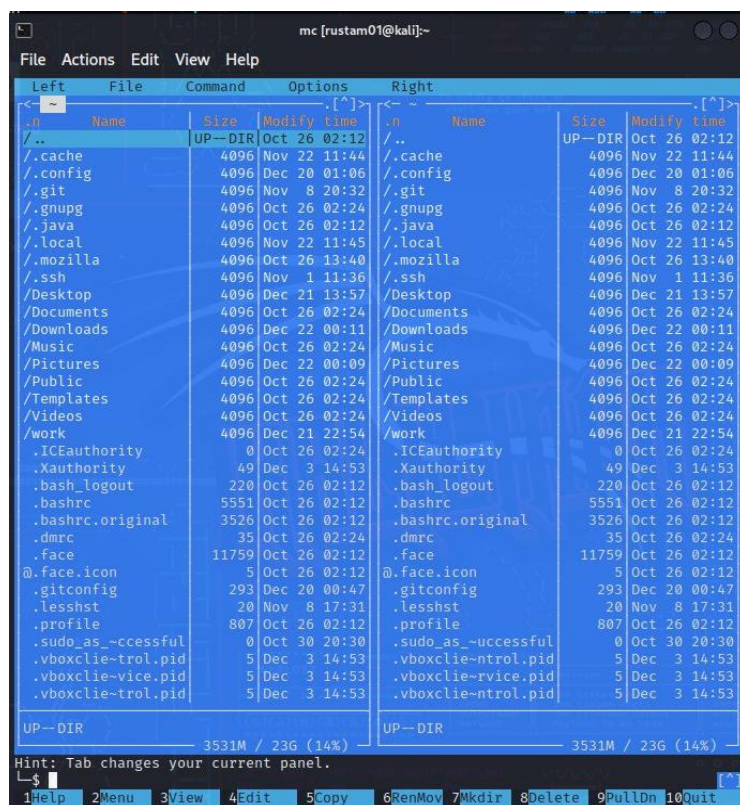


Рис. 4.1: 1. Midnight Commander

14. с помощью клавиш `h,h` и `Enter` перехожу в каталог `~/work/arch-rc` созданный при выполнении лабораторной работы №4 (рисунки 2 и 3)

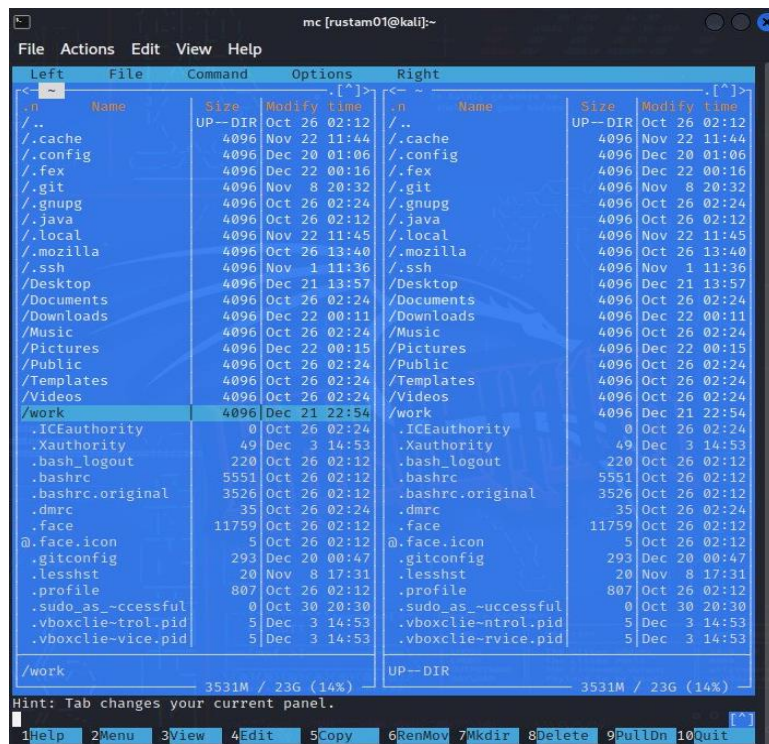


Рис. 4.2: 2. Переход в work

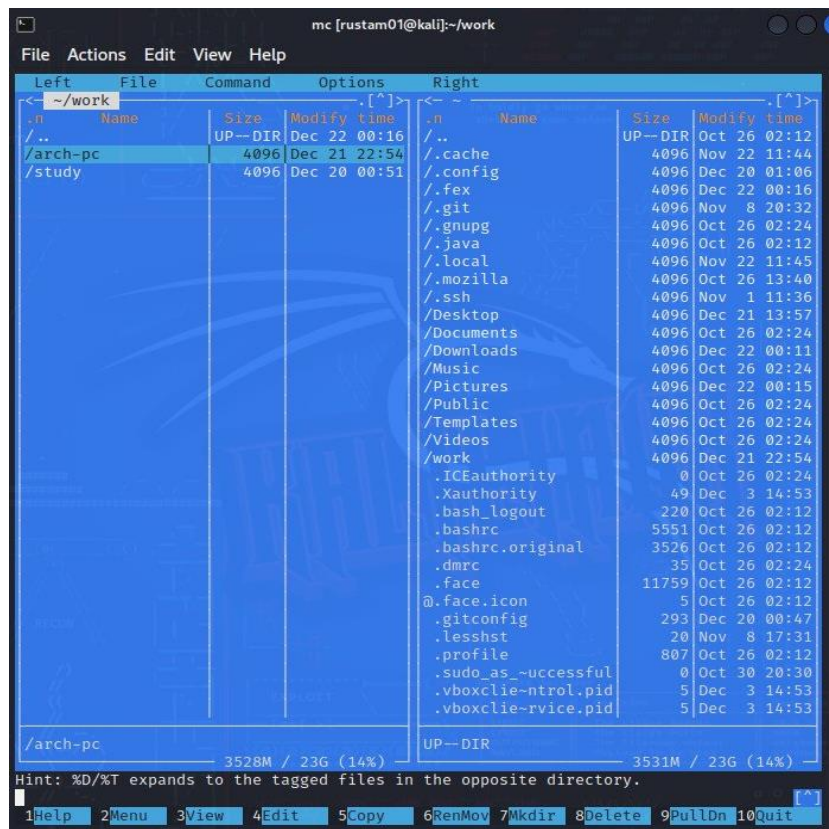


Рис. 4.3: 3. Переход в arch-pc

3) С помощью F7 создаю папку lab05

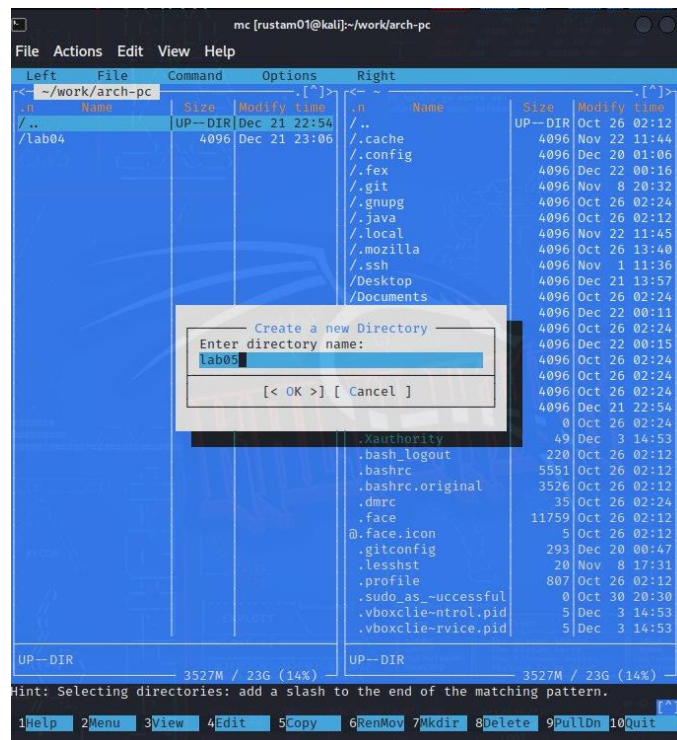


Рис. 4.4: 4. Создание папки lab05

4) Перехожу в созданный каталог и с помощью touch создаю lab5-1.asm

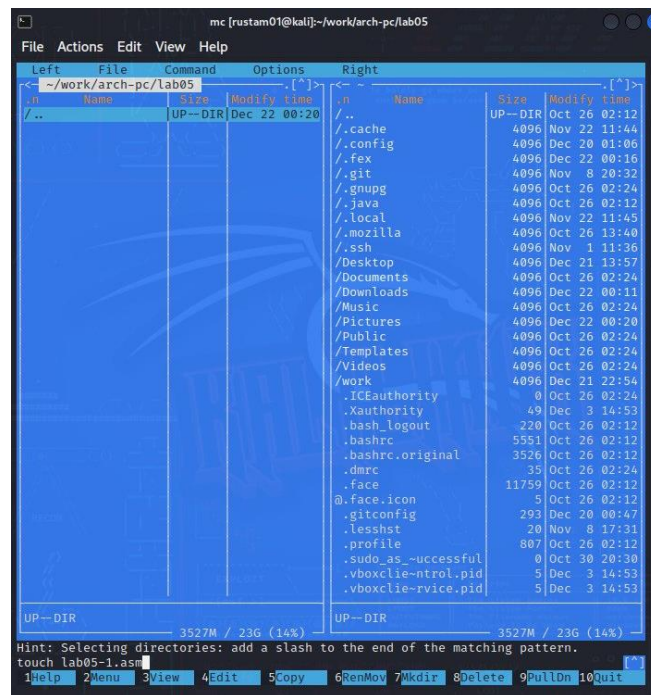


Рис. 4.5: 5. lab05

Созданный файл

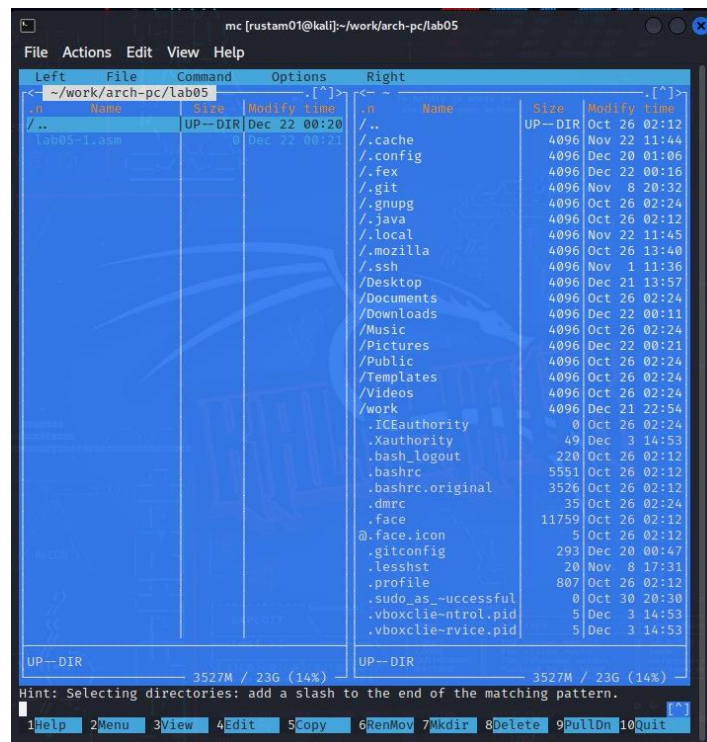


Рис. 4.6: 6. Созданный файл

- 5) С помощью F4 открываю файл lab5-1.asm во встроенном редакторе, в моём случае - папо, и копирую туда код из задания лабораторной работы, сохраняю изменения

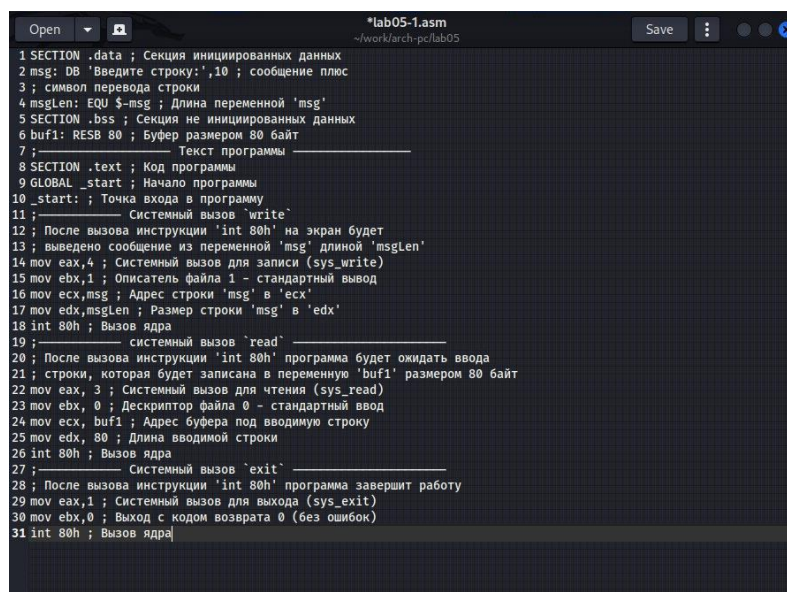


Рис. 4.9: 8. Оттранслирую файл

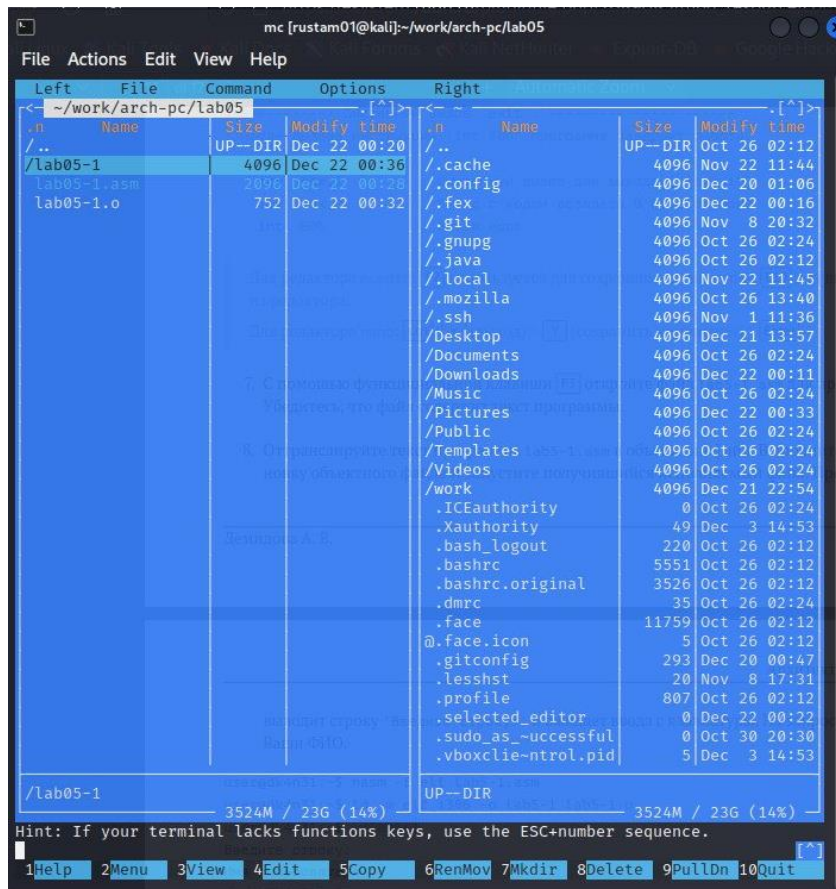


Рис. 4.10: 9. Компоную файл

8) Работа файла

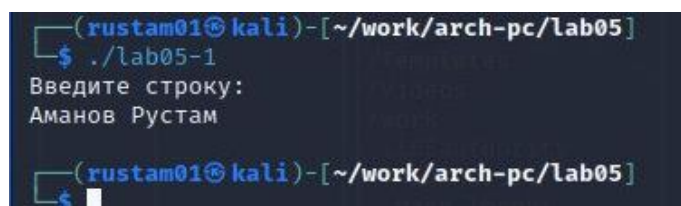


Рис. 4.11: 10. Работа файла

9) Скачиваю in_out.asm и открываю во втором окне

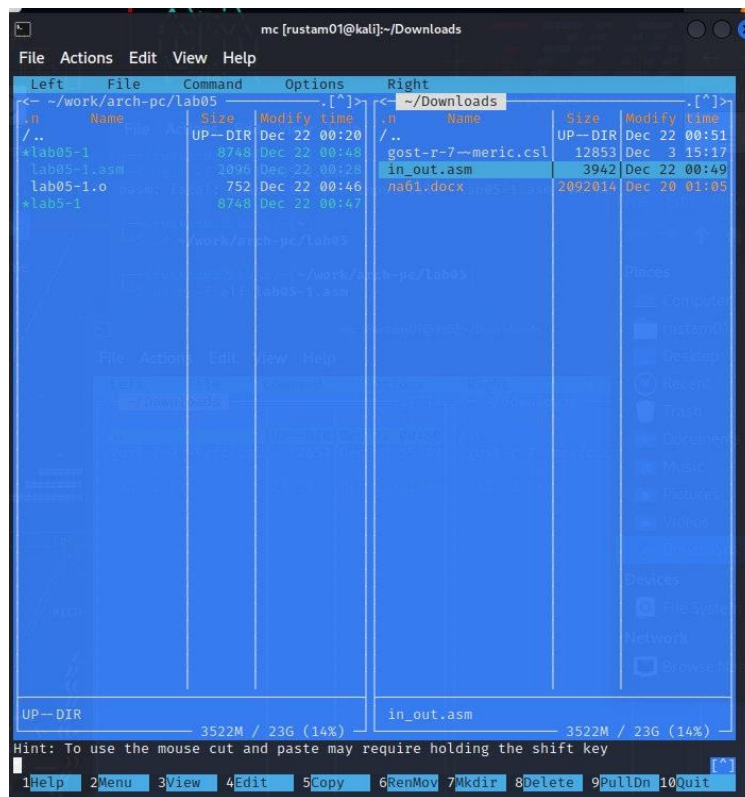


Рис. 4.12: 11.два окна

10) С помощью F5 копирую файл в нужную папку

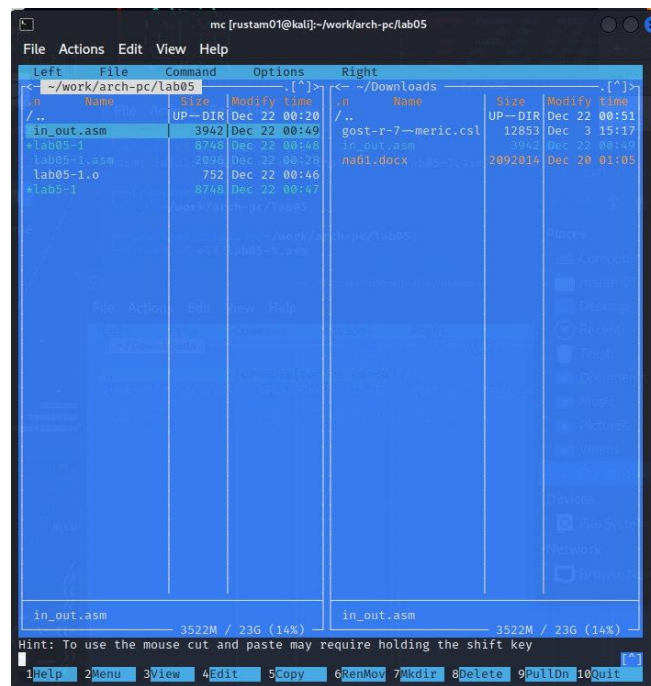


Рис. 4.13: 12. Копирование файла

11) С помощью F5 создаю копию файла lab5-1.asm с именем lab5-2.asm. (13-14)

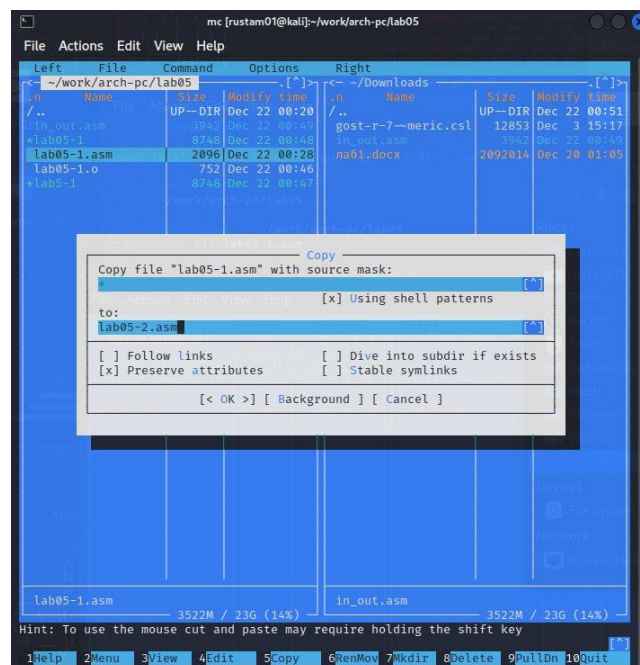


Рис. 4.14: 13. Копирование файла

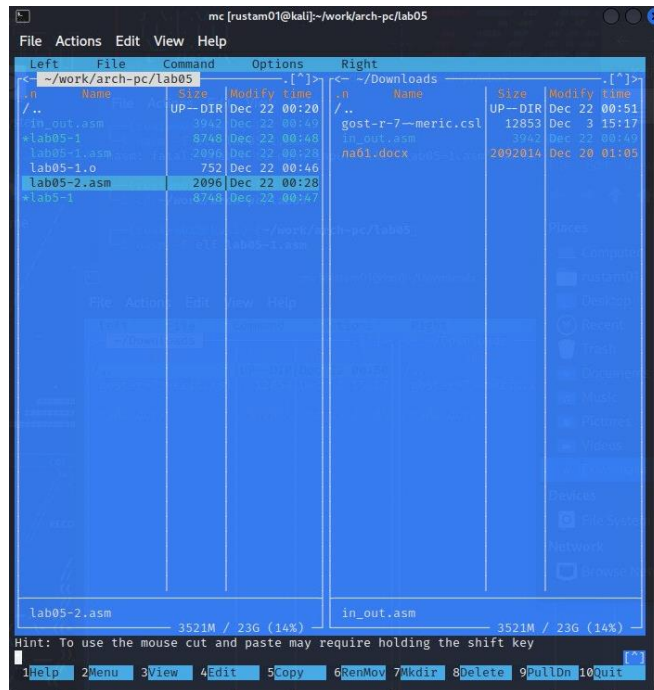
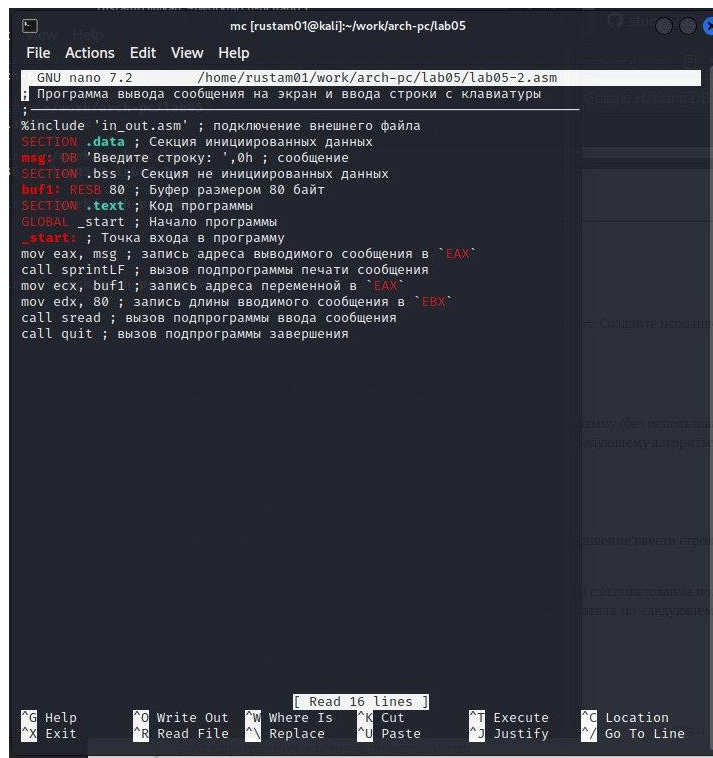


Рис. 4.15: 14. Скопированный файл

12) Открываю lab5-2.asm для редактирования в mc edit



```
mc [rustam01@kali]~/work/arch-pc/lab05
GNU nano 7.2 /home/rustam01/work/arch-pc/lab05/lab05-2.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 sprintf ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса переменной в `EAX`
mov edx, 80 ; запись длины вводимого сообщения в `EBX`
call read ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения

[ Read 16 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^N Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

Рис. 4.16: 15. Редактирование файла

13) Оттранслирую в объектный файл и выполняю компоновку файла (16-17)

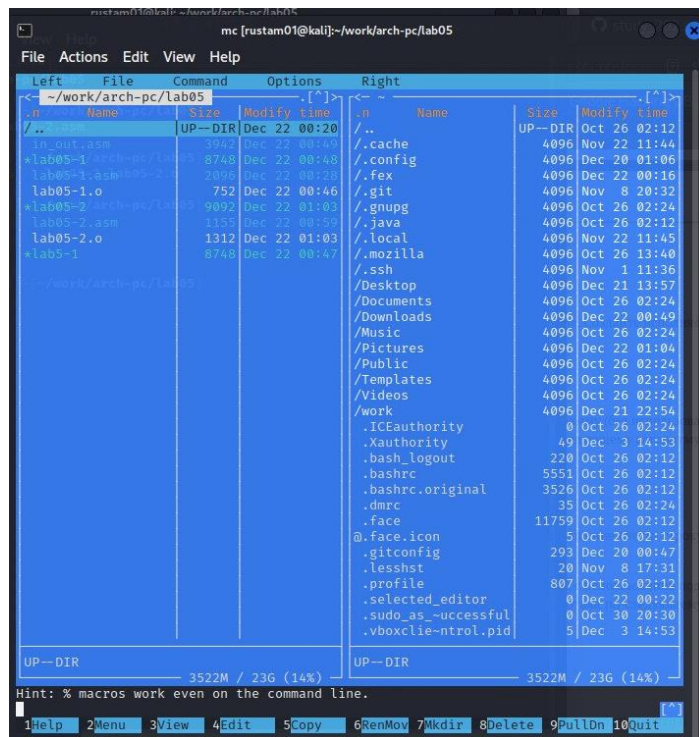


Рис. 4.17: 16. Созданный объектный файл

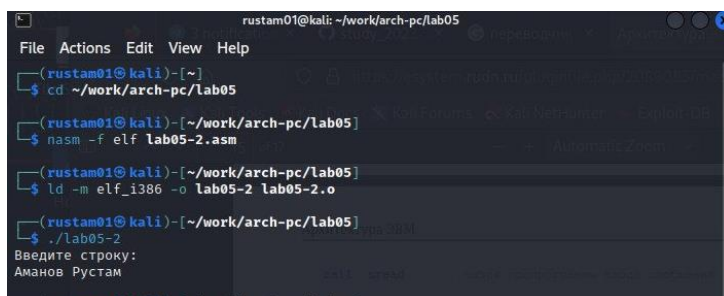
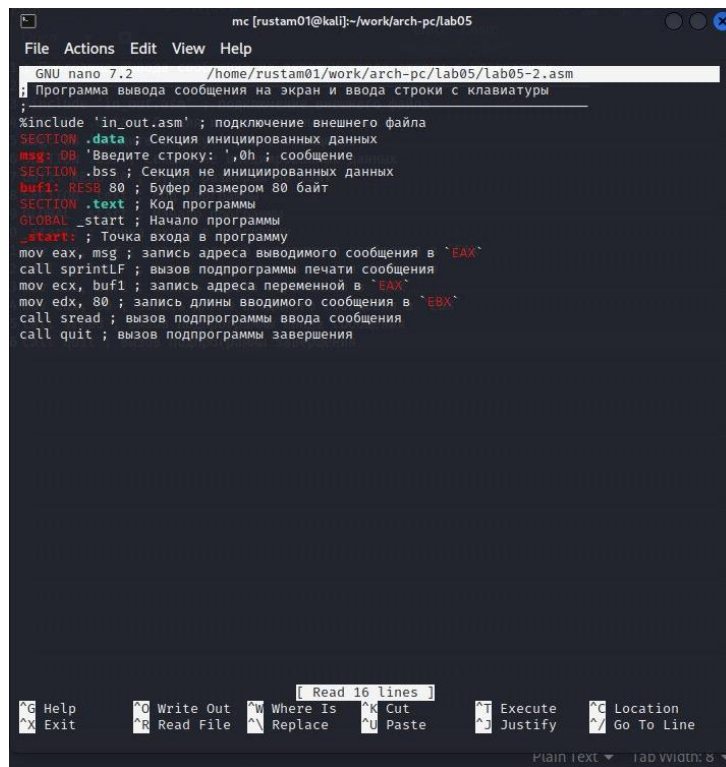


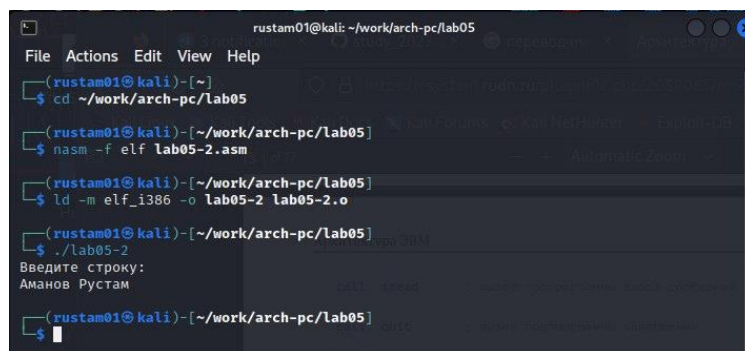
Рис. 4.18: 17. Создание и работы исполняемого файла

14) Заменяю подпрограмму sprintLF на sprint, создаю файл и проверяю его работу (18-19)



```
mc [rustam01@kali]:~/work/arch-pc/lab05
File Actions Edit View Help
GNU nano 7.2 /home/rustam01/work/arch-pc/lab05/lab05-2.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 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
```

Рис. 4.19: 18. Редактирование файла



```
rustam01@kali: ~/work/arch-pc/lab05
File Actions Edit View Help
(rustam01@kali)-[~]
$ cd ~/work/arch-pc/lab05
(rustam01@kali)-[~/work/arch-pc/lab05]
$ nasm -f elf lab05-2.asm
(rustam01@kali)-[~/work/arch-pc/lab05]
$ ld -m elf_i386 -o lab05-2 lab05-2.o
(rustam01@kali)-[~/work/arch-pc/lab05]
$ ./lab05-2
Введите строку:
Аманов Рустам
(rustam01@kali)-[~/work/arch-pc/lab05]
$
```

Рис. 4.20: 19. Работа файла

5. Выполнение заданий для самостоятельной работы

1) Копирую lab5-1.asm как lab5-1-1.asm

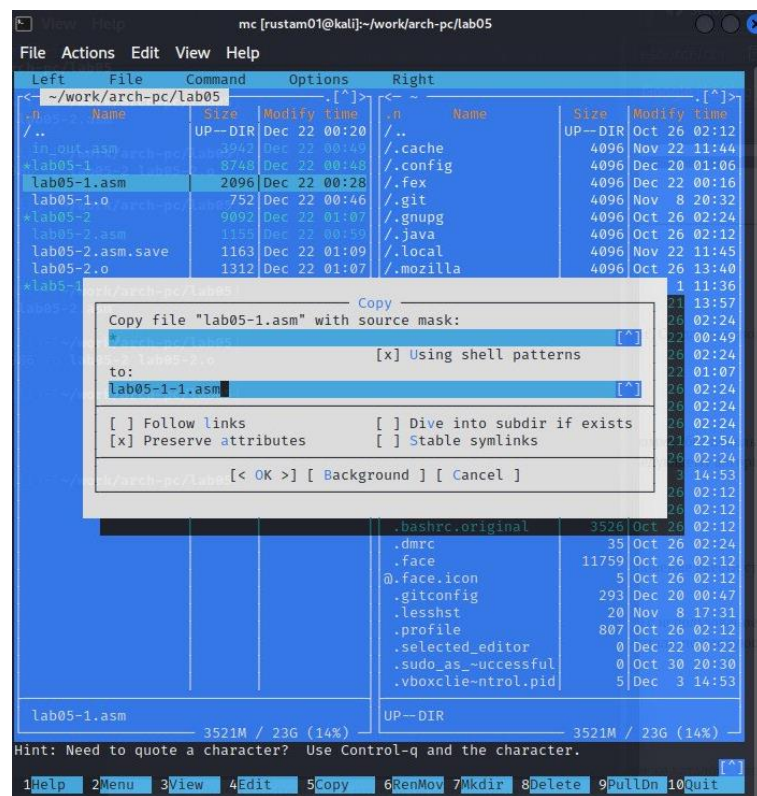
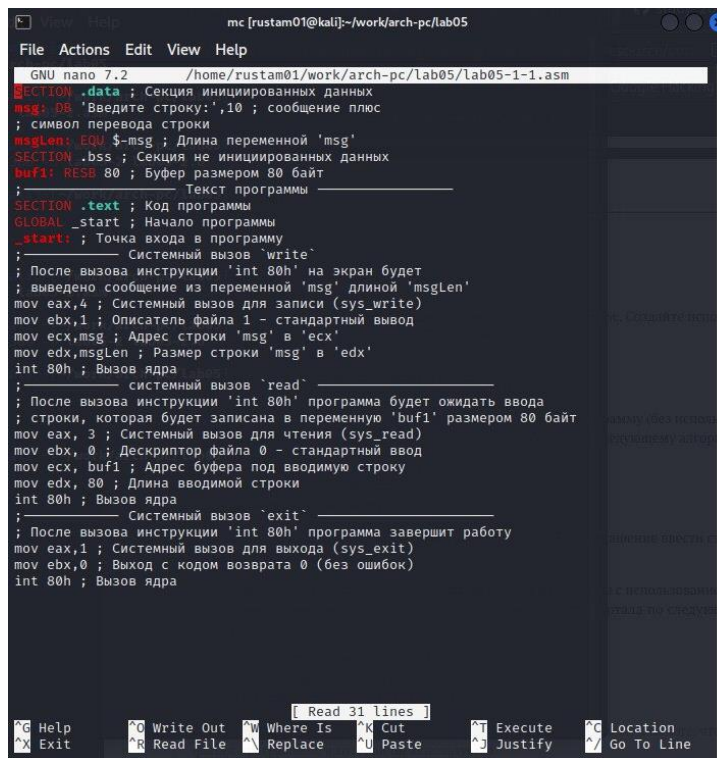


Рис. 5.1: 20. Копирование файла

2) Открываю его с помощью mcedit



```
GNU nano 7.2 /home/rustam01/work/arch-pc/lab05/lab05-1-1.asm
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 '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. 5.2: 21. Открытый файл

3) Вношу изменения так, чтобы программа возвращала введённое значение код программы:

```
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
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 ; mov ebx, 1 ; mov ecx, buf1 ;
mov edx,buf1 ; int 80h ; mov eax,1 ; Системный вызов для выхода
(sys_exit) mov ebx,0 ; Выход с кодом возврата 0 (без ошибок) int 80h ;
Вызов ядра

6. Выводы

Я приобрела практические навыки работы в Midnight Commander и освоила инструкции языка ассемблера mov и int

Список литературы