Архитектура компьютера

Отчёт по лабораторной работе №5

Ибрахим Мохсейн Алькамал

Содержание

# 1 Цель работы

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

# 2 Задание

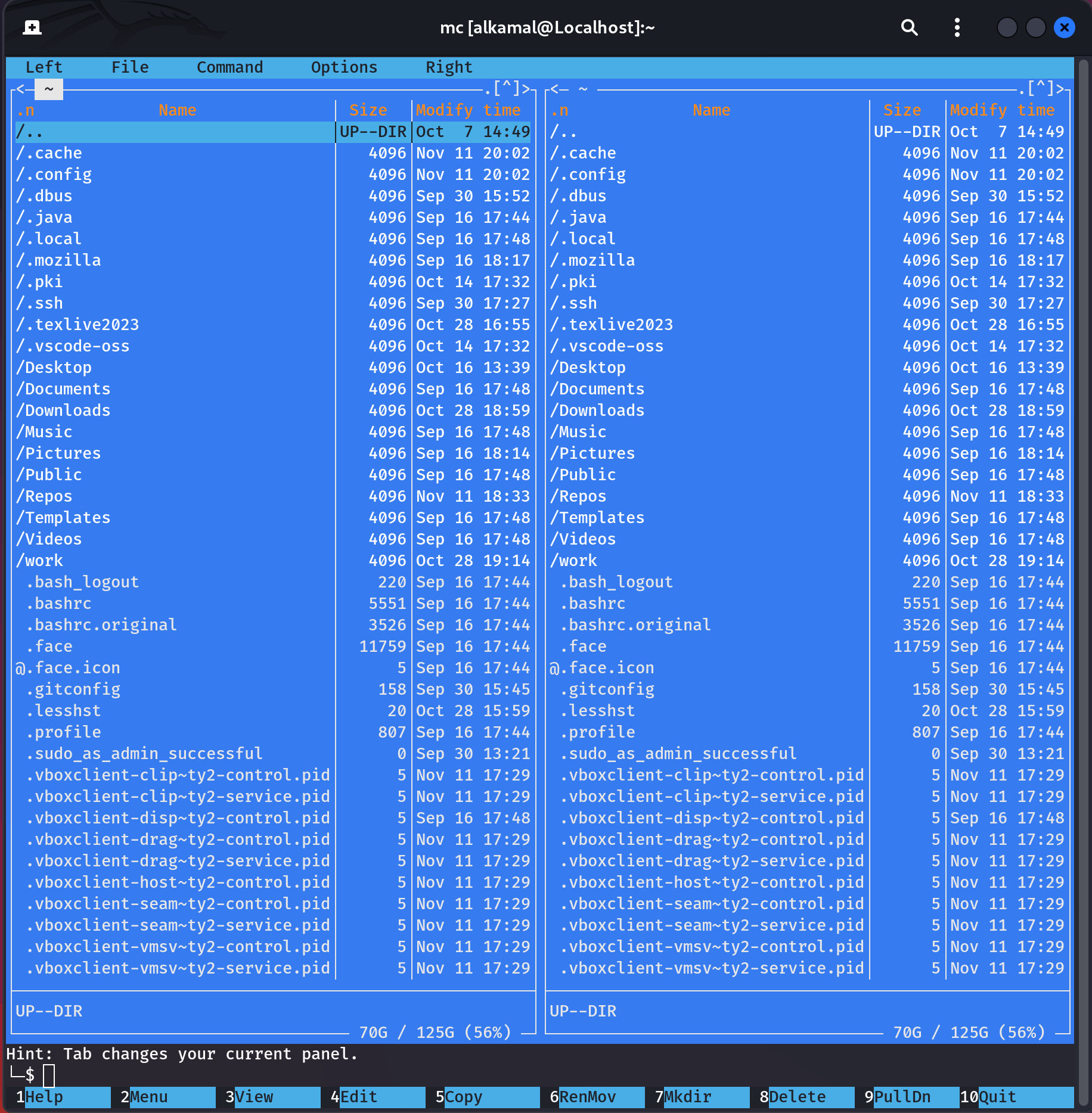
1. Откройте Midnight Commander user@dk4n31:~$ mc
2. Пользуясь клавишами ↑ , ↓ и Enter перейдите в каталог ~/work/arch-pc созданный при выполнении лабораторной работы №4.
3. С помощью функциональной клавиши F7 создайте папку lab05 и перейдите в созданный каталог.
4. Пользуясь строкой ввода и командой 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 использование 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). Таблица 5.1. Функциональные клавиши Midnight Commander 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

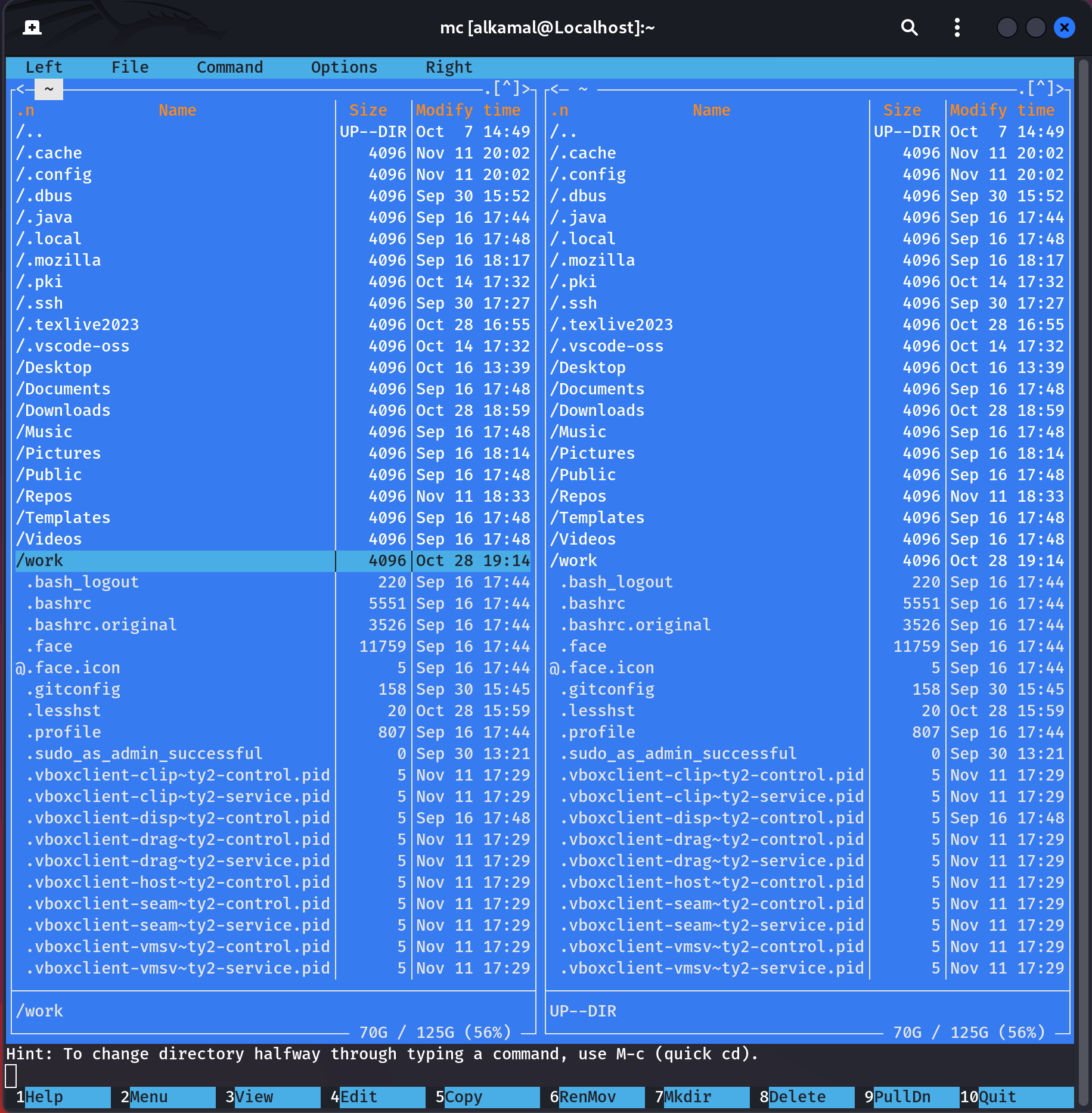
# 4 Выполнение лабораторной работы

1. Открываю Midnight Commander с помощью команды mc

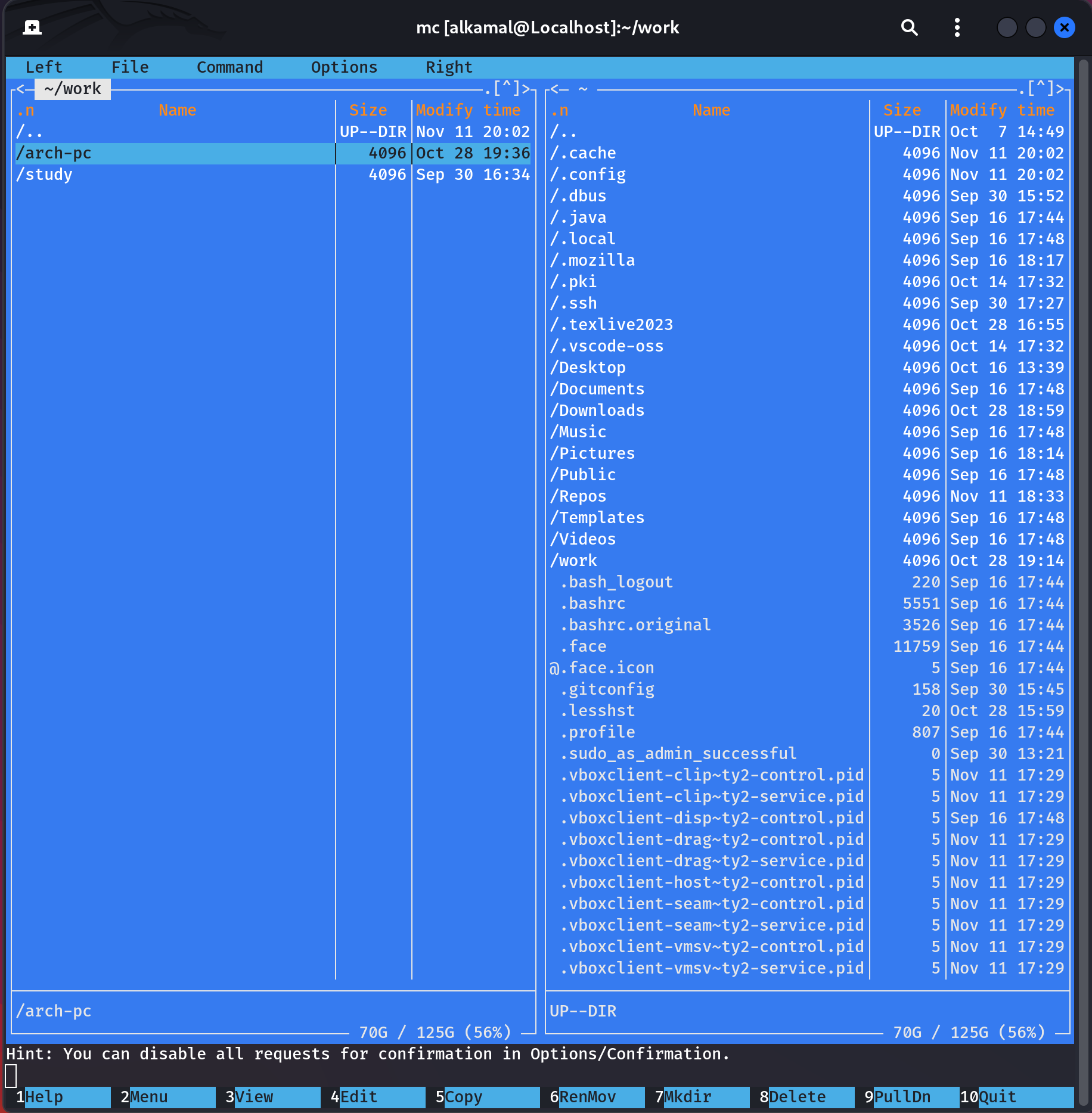


1. Midnight Commander

1. с помощью клавиш ↑ , ↓ и Enter перехожу в каталог ~/work/arch-pc созданный при выполнении лабораторной работы №4 (рисунки 2 и 3)

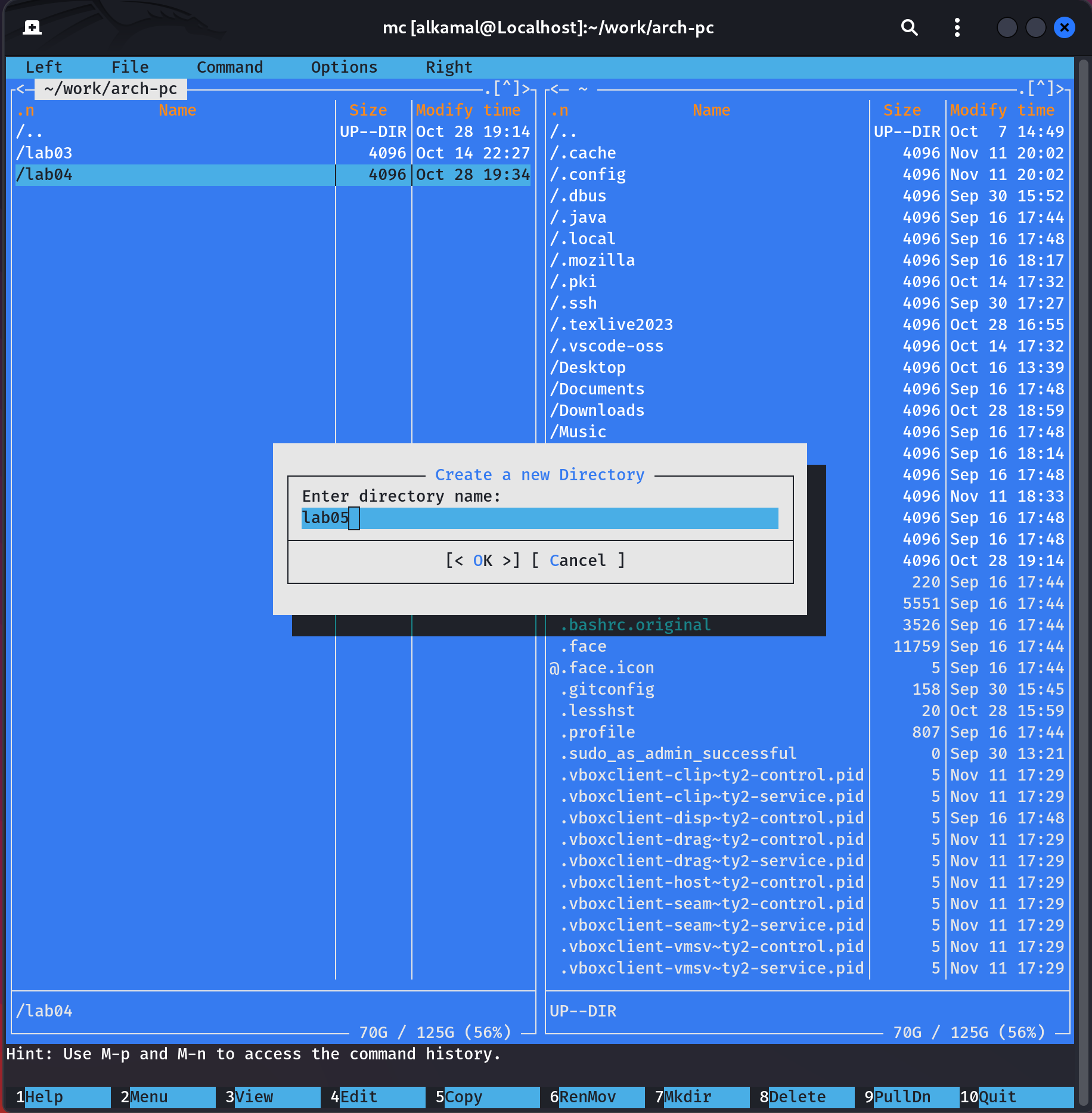


2. Переход в work



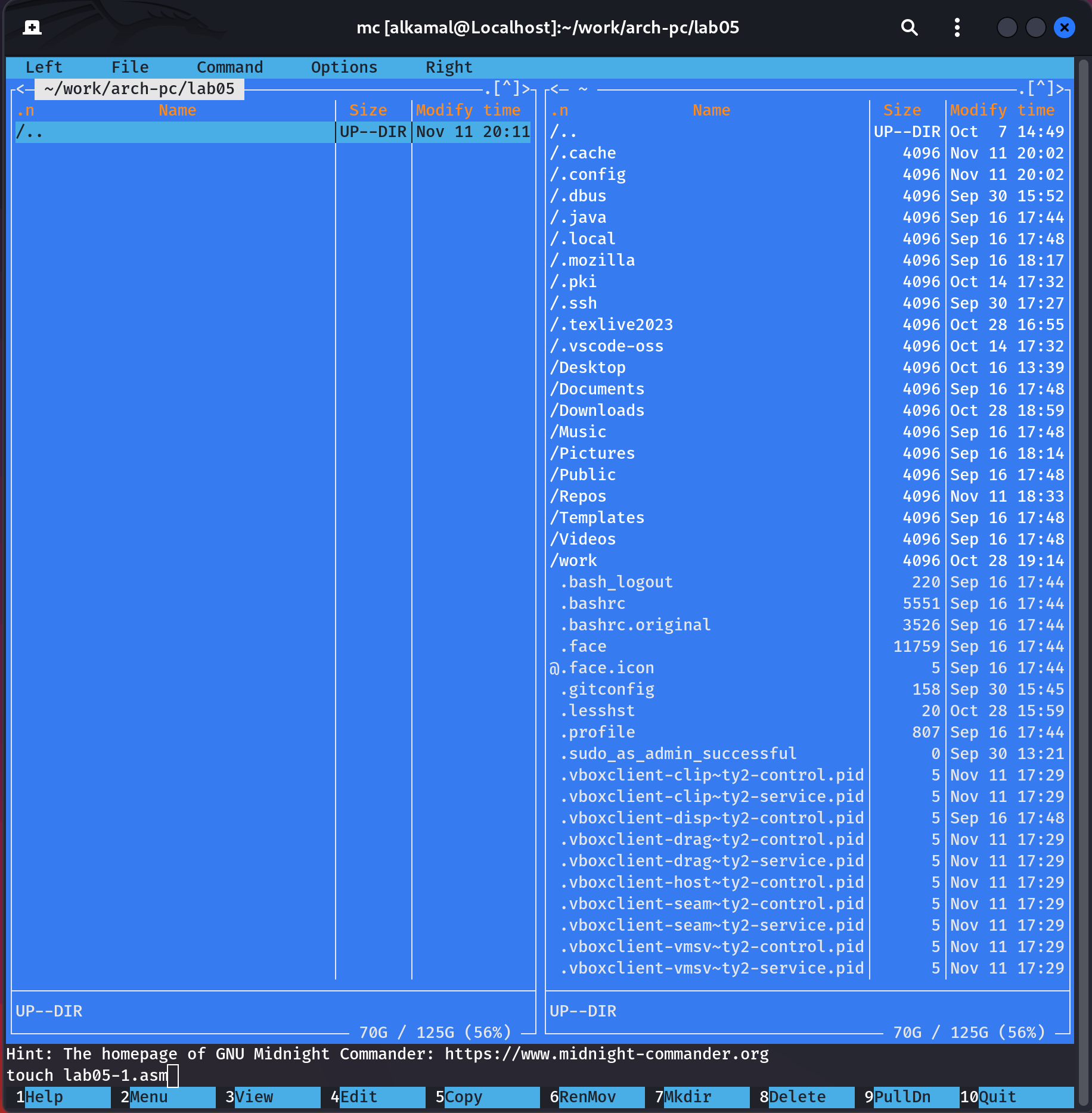
3. Переход в arch-pc

1. С помощью F7 создаю папку lab05



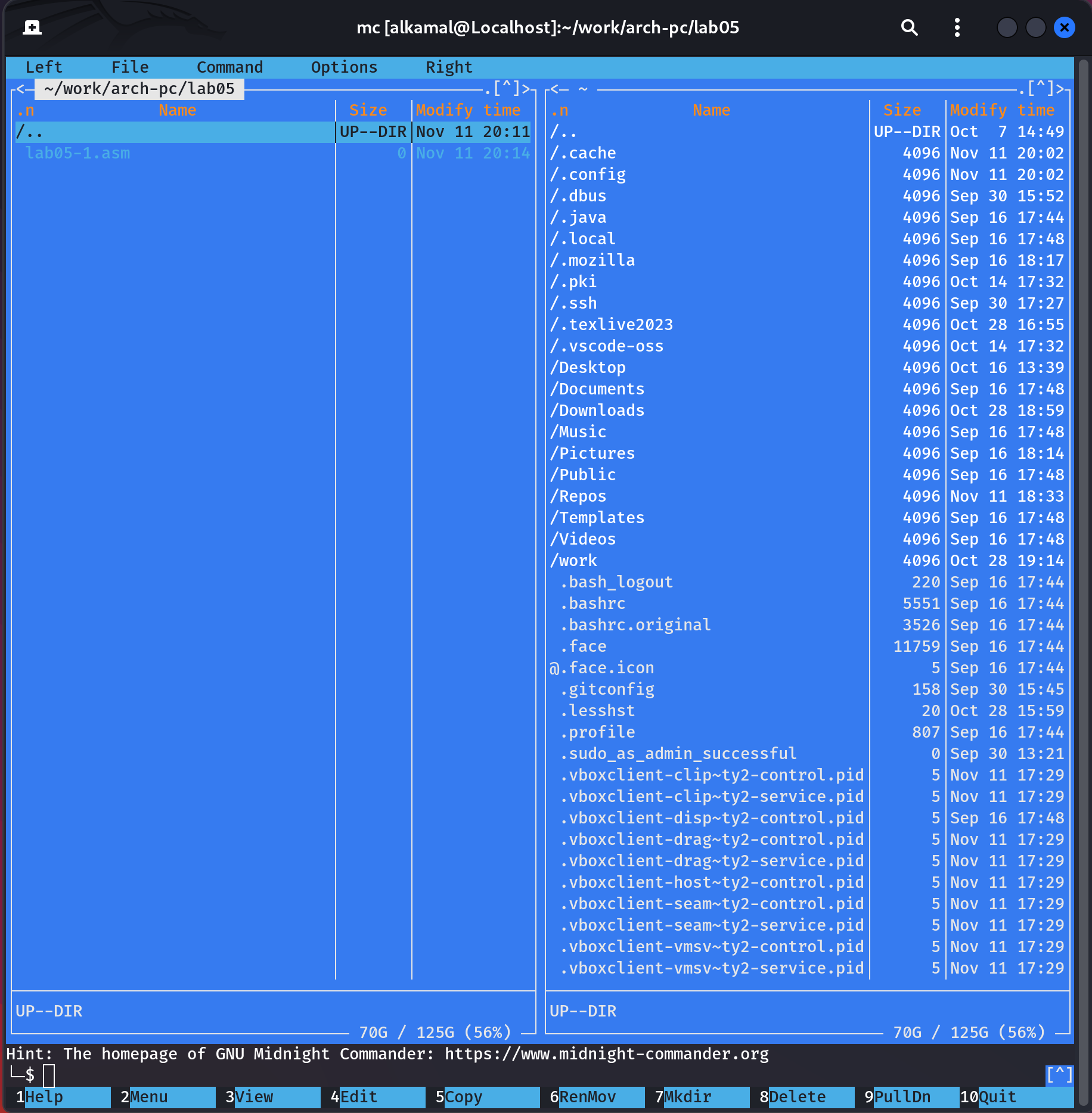
4. Создание папки lab05

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



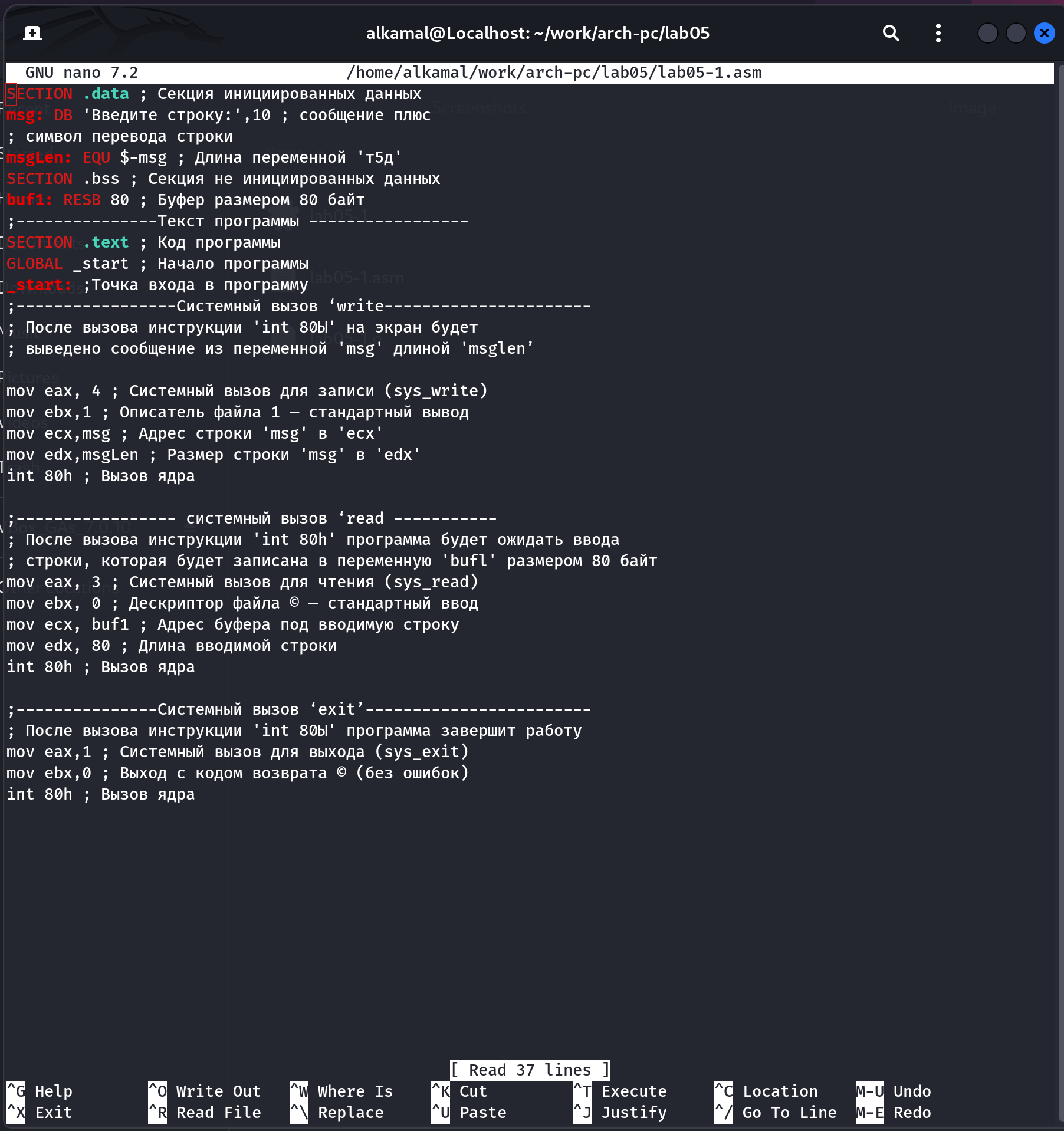
5. lab05

Созданный файл



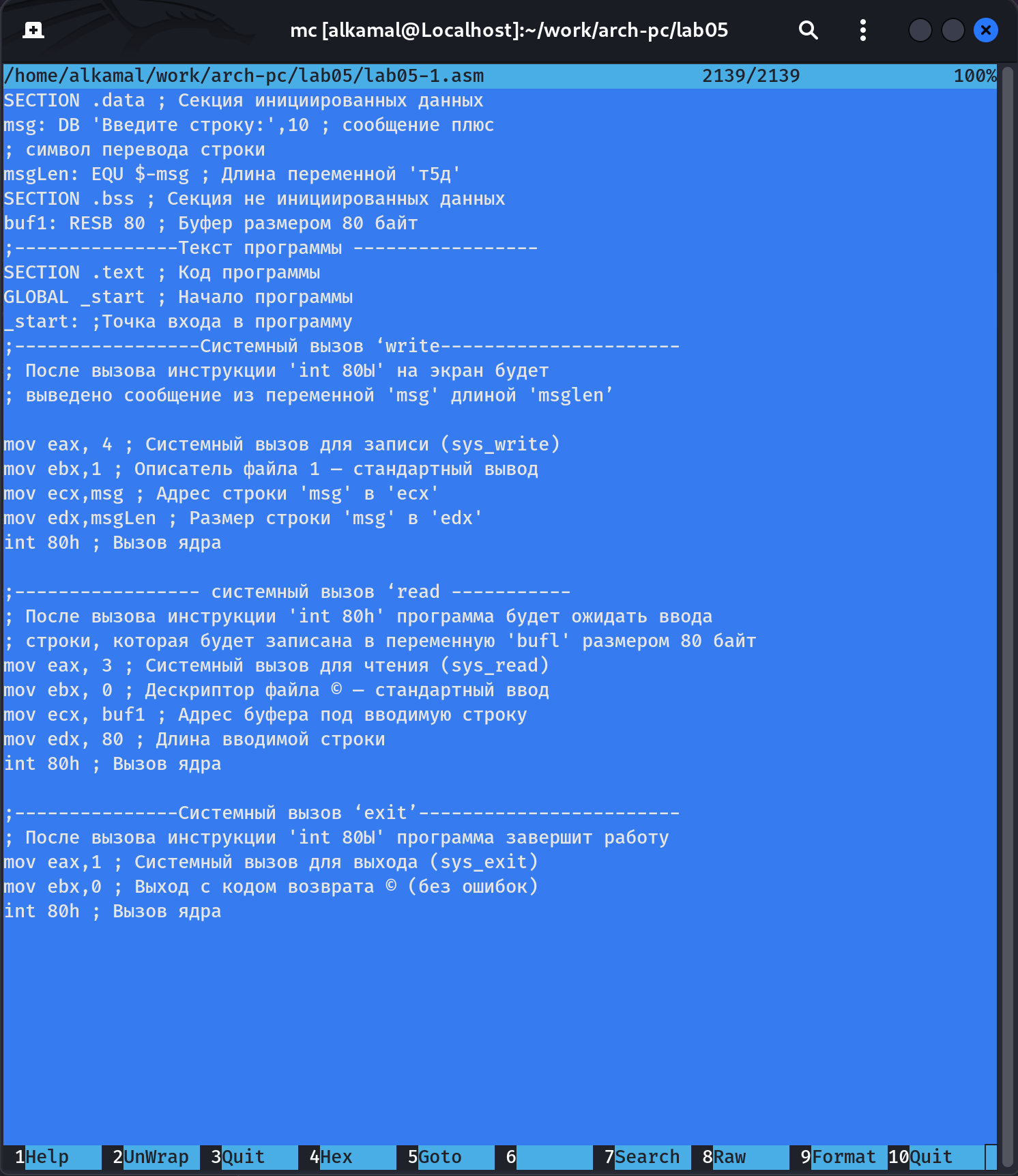
6. Созданный файл

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



7. Открытие файла в nano

1. С помощью F3 открываю файл и смотрю на сохраненные изменения

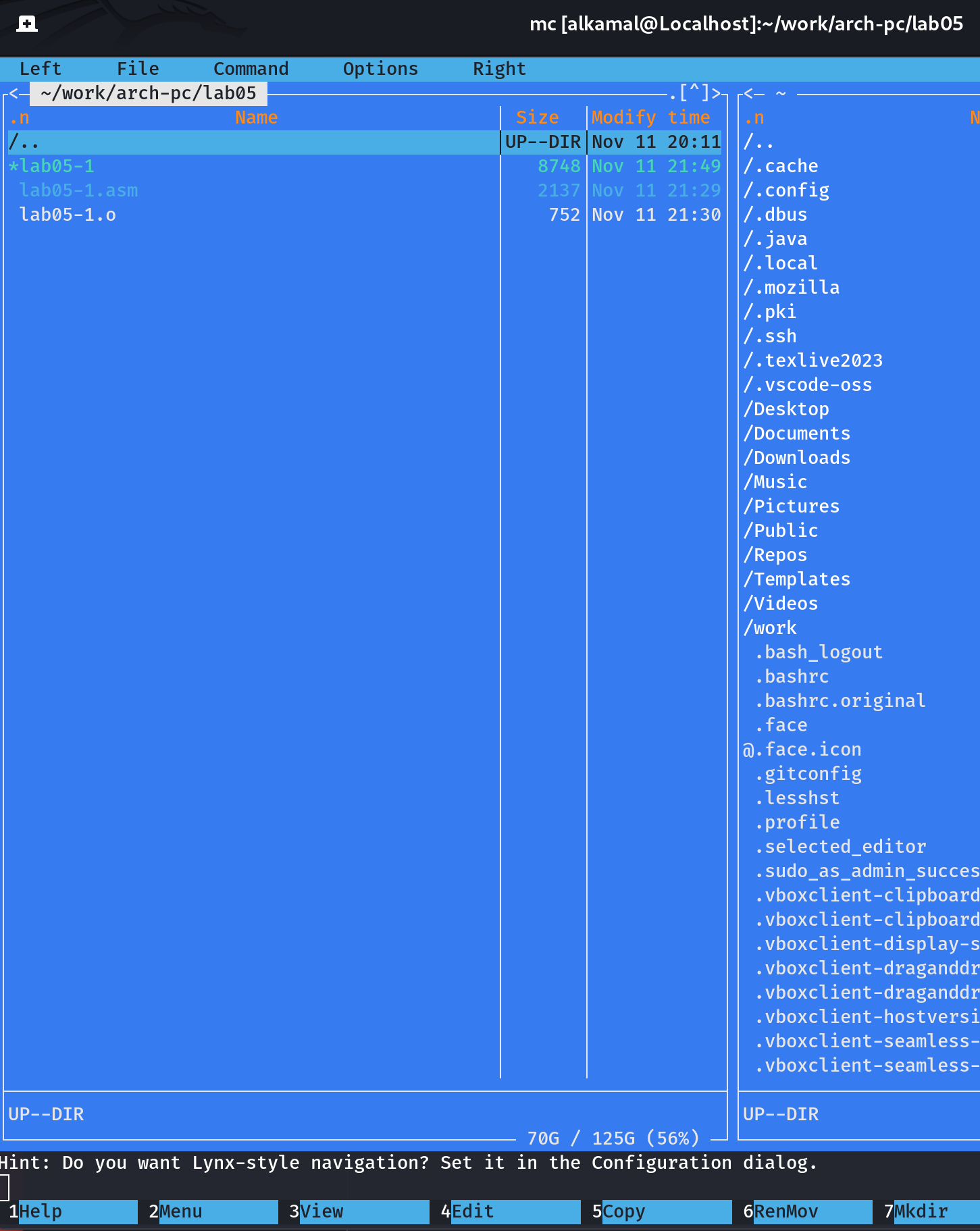


7.1 Просмотр файла

1. Оттранслирую в объектный файл и выполняю компоновку файла (8-9)

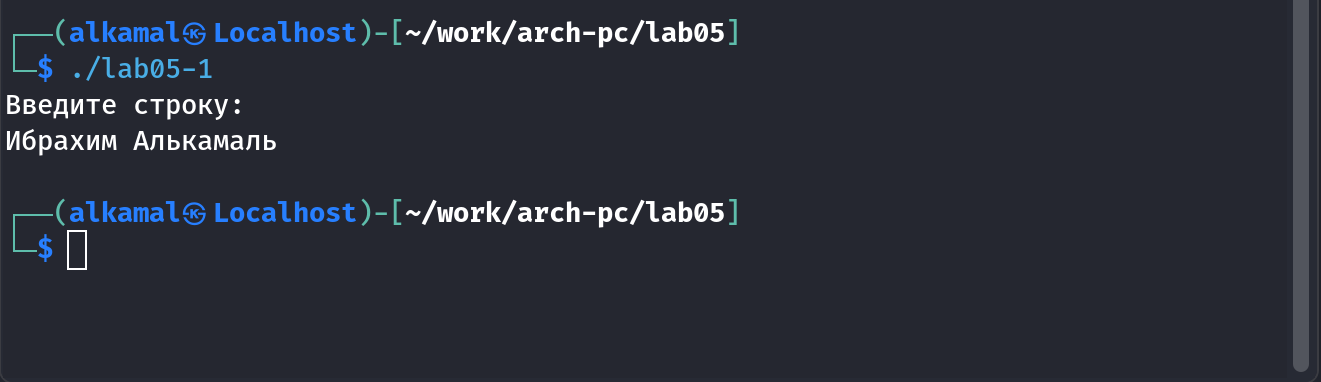


8. Оттранслирую файл



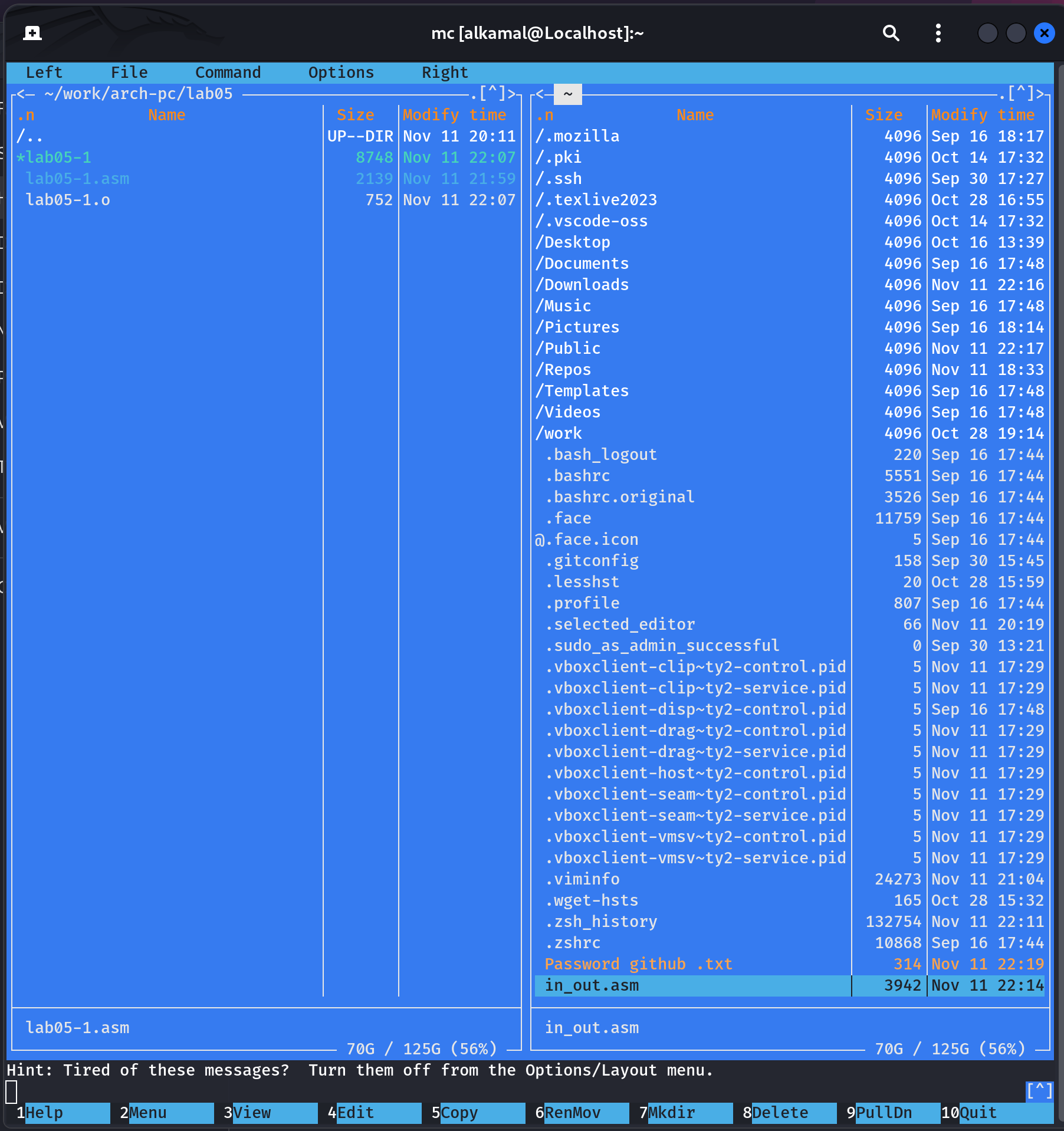
9. Компоную файл

1. Работа файла



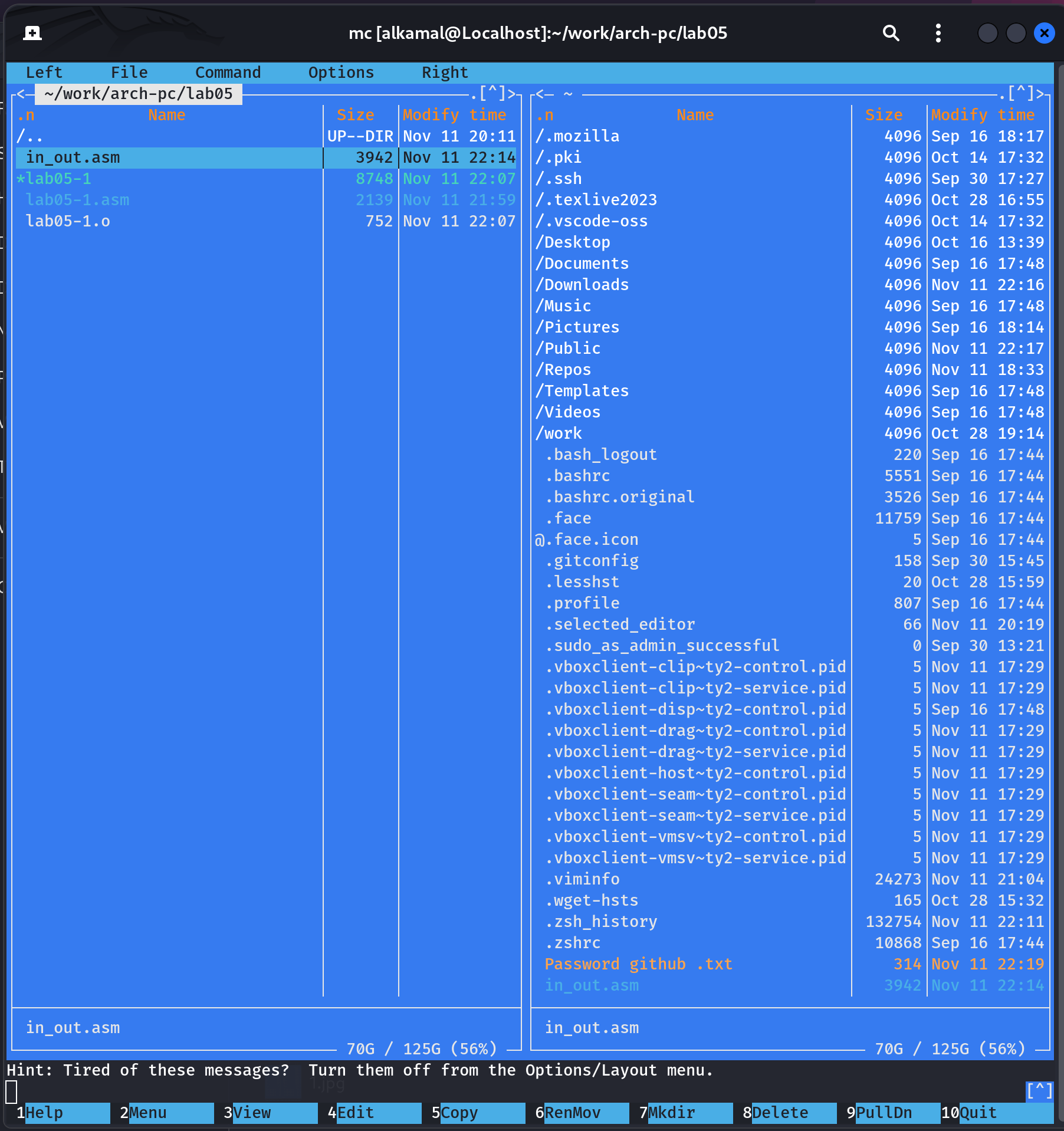
10. Работа файла

1. Скачиваю in\_out.asm и открываю во втором окне



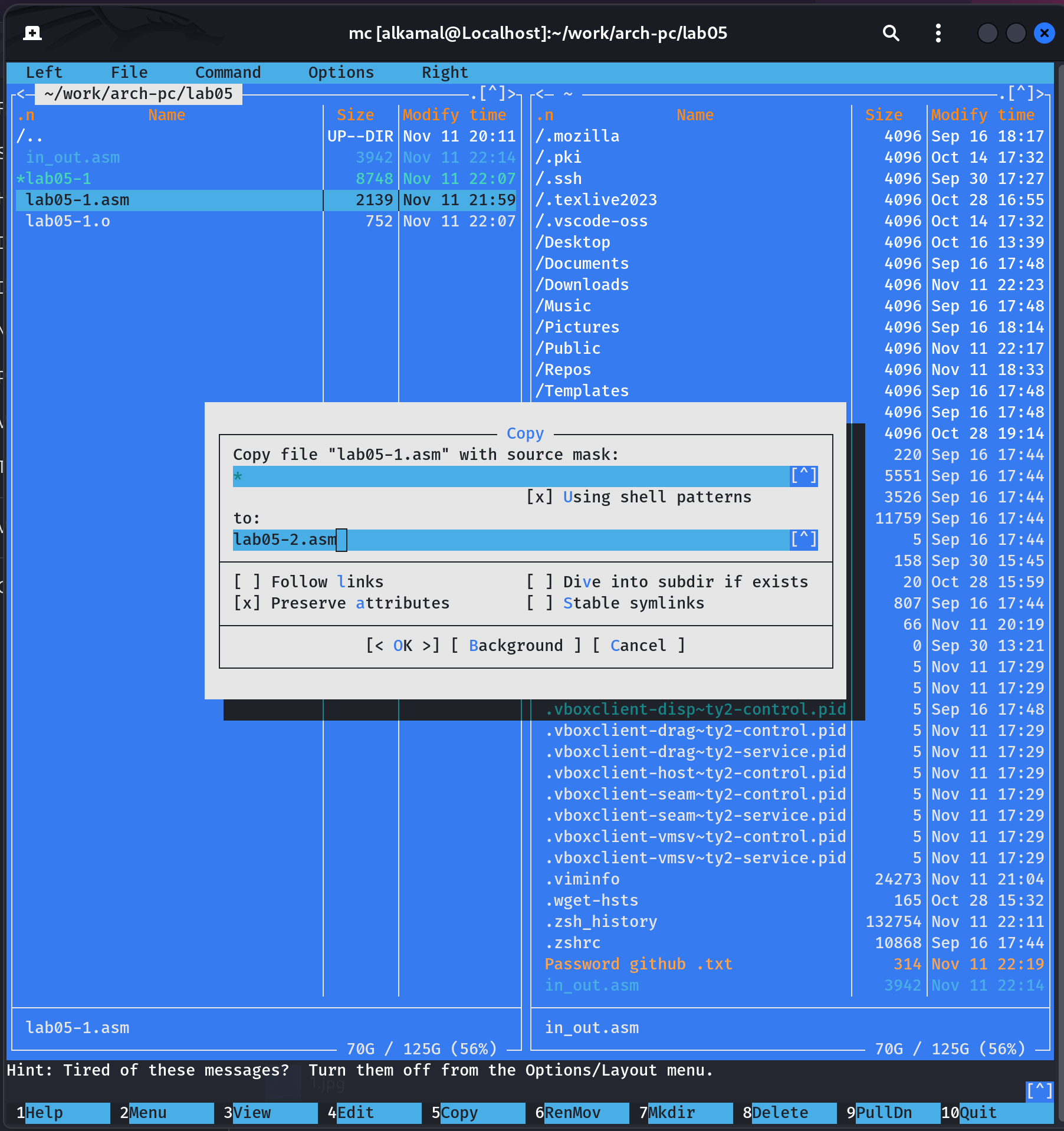
11. два окна

1. С помощью F5 копирую файл в нужную папку

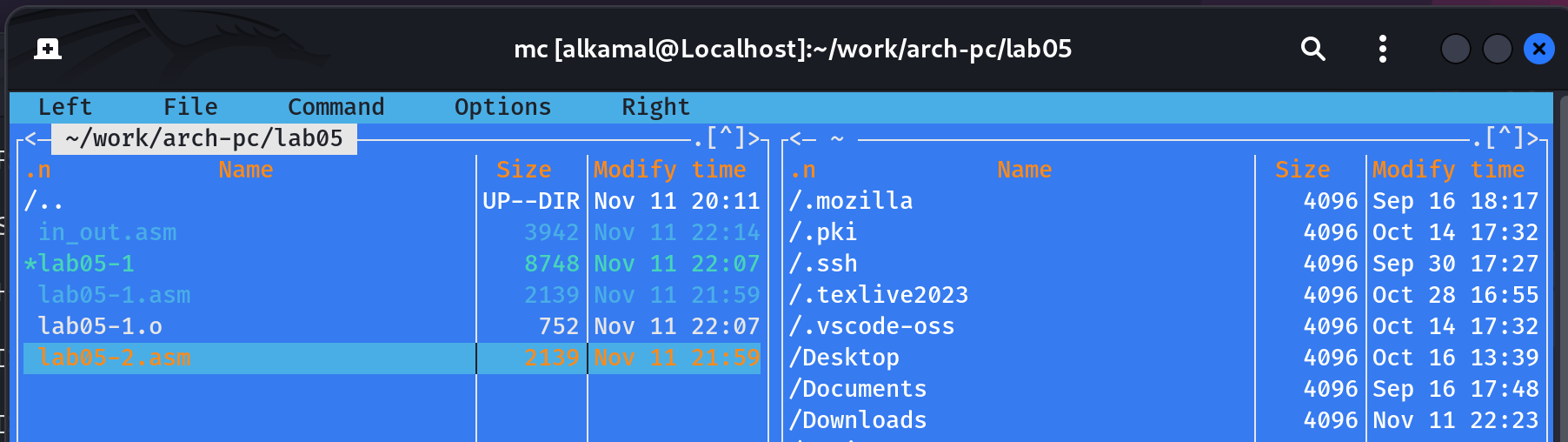


12. Копирование файла

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

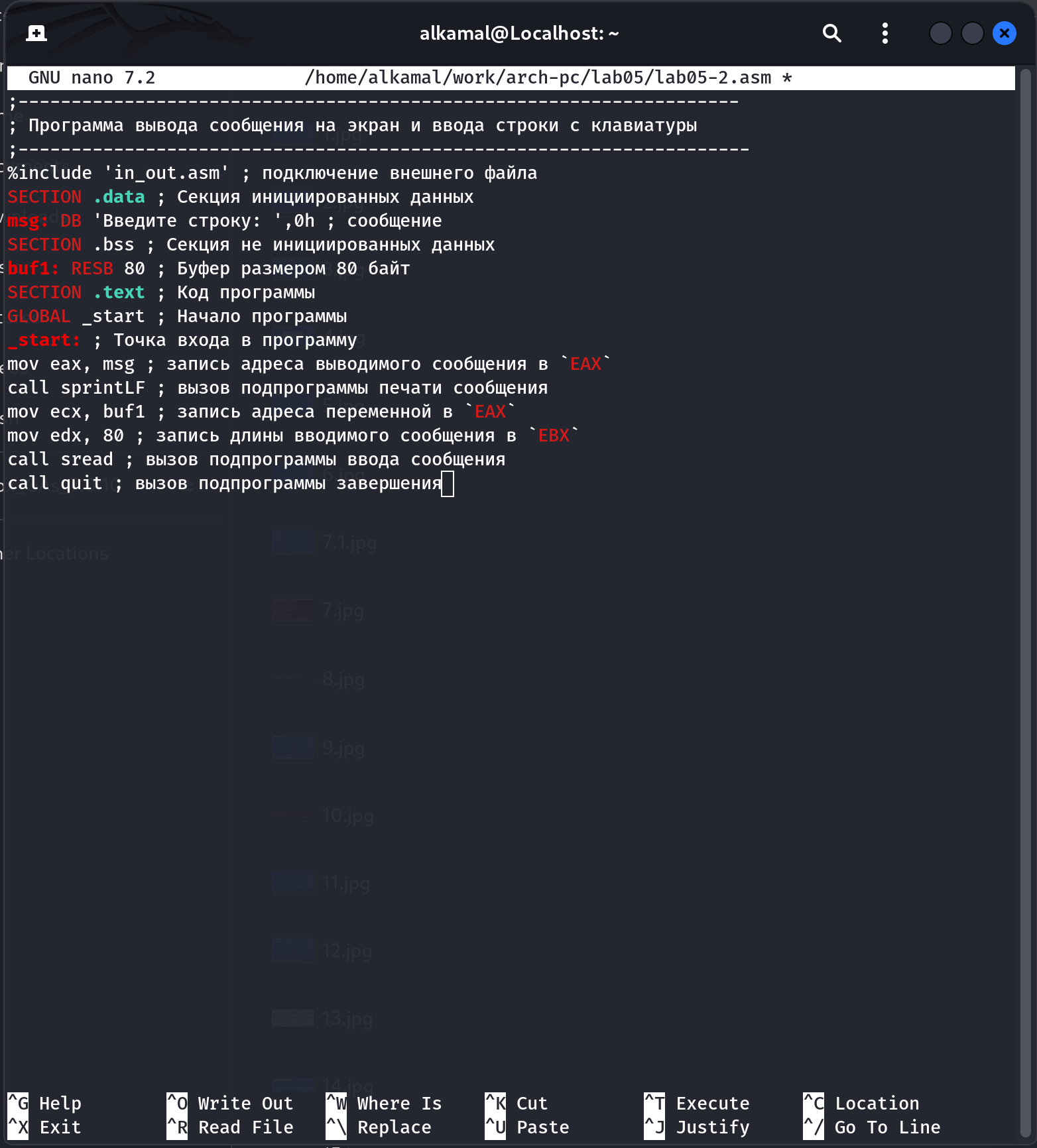


13. Копирование файла



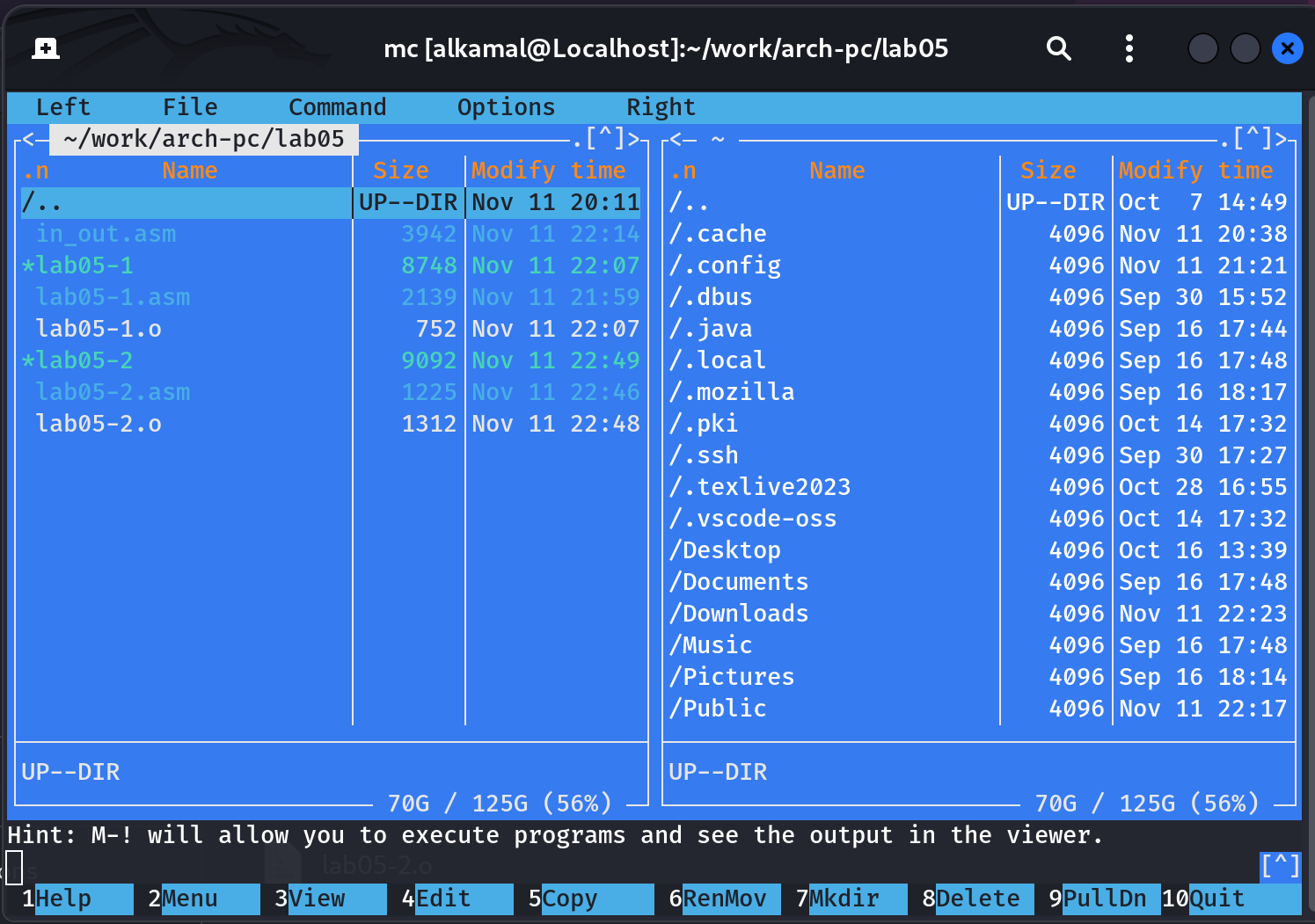
14. Скопированный файл

1. Открываю lab5-2.asm для редактирования в mcedit

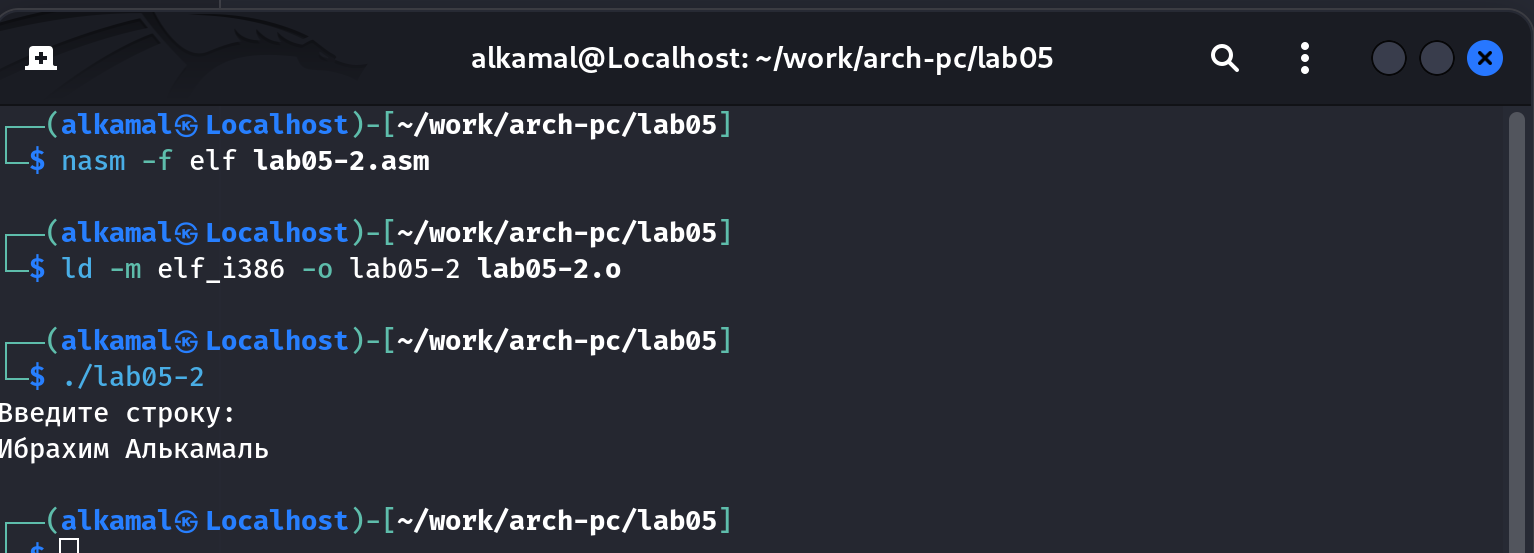


15. Редактирование файла

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



16. Созданный объектный файл

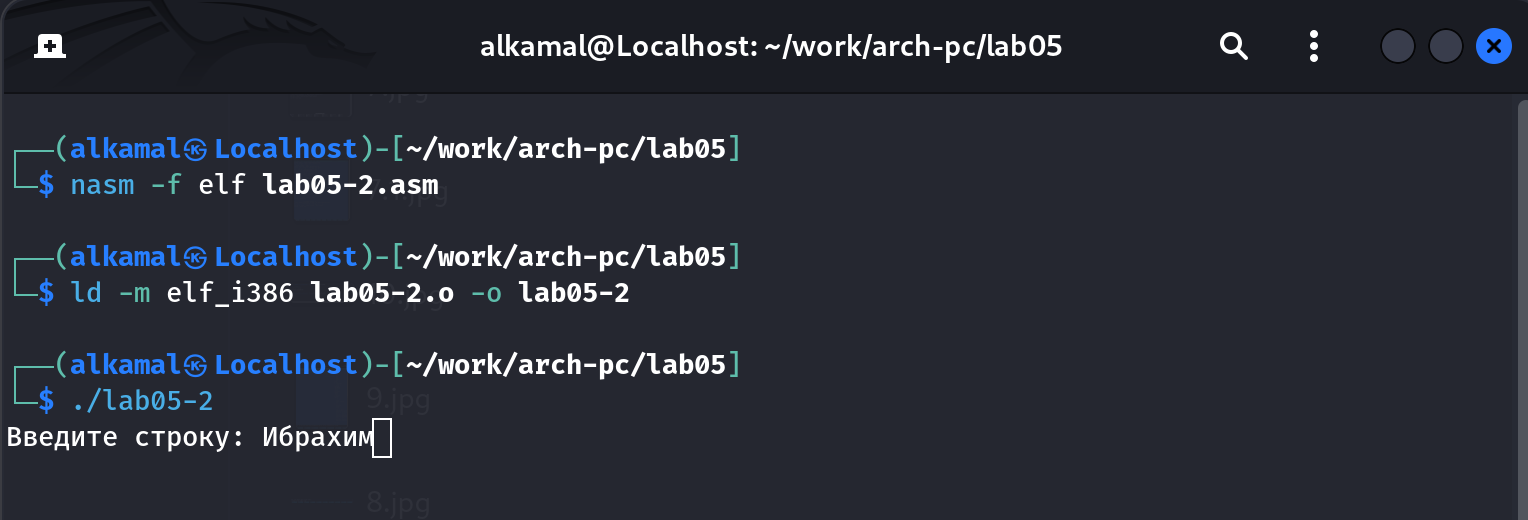


17. Создание и работы исполняемого файла

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



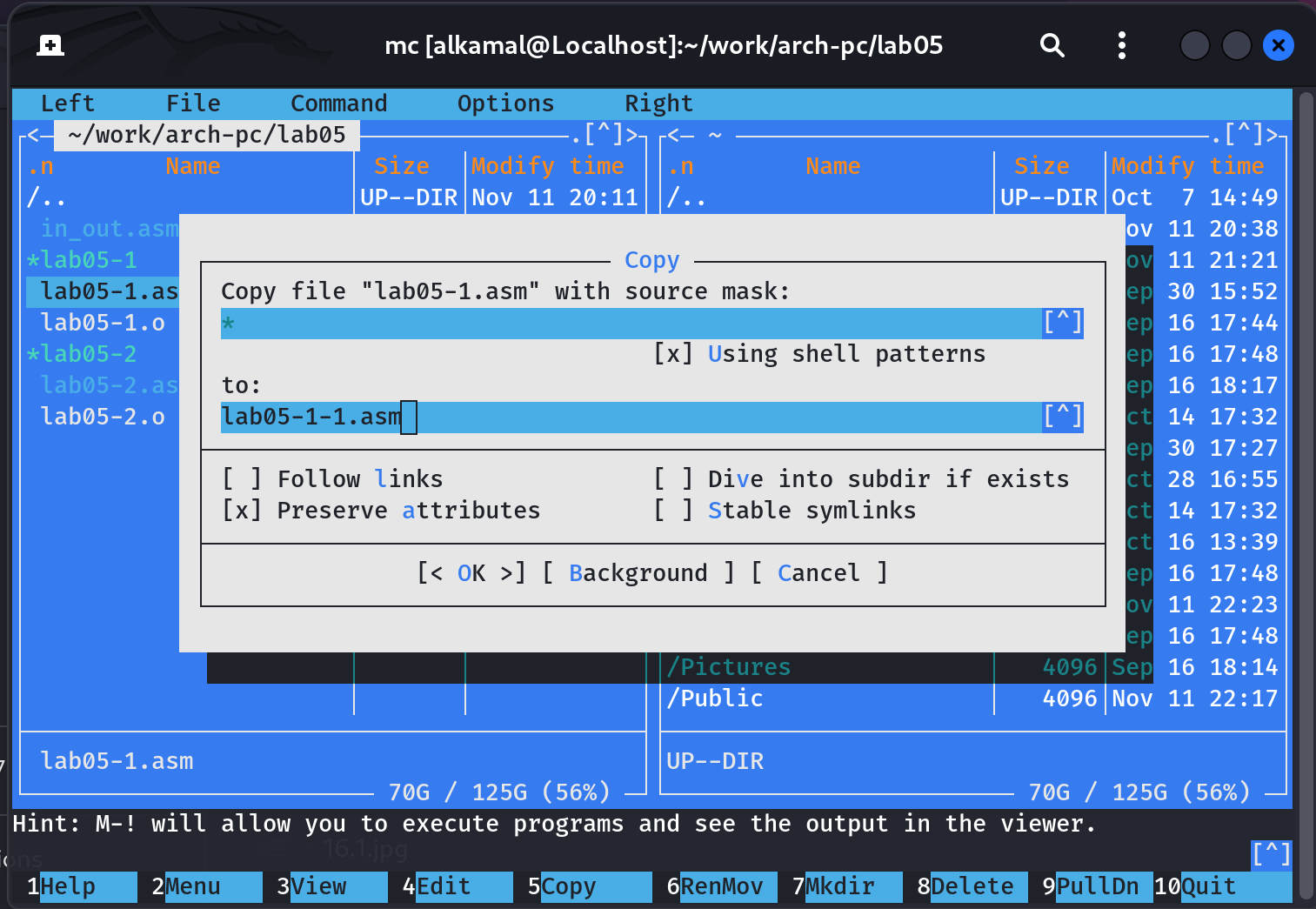
18. Редактирование файла



19. Работа файла

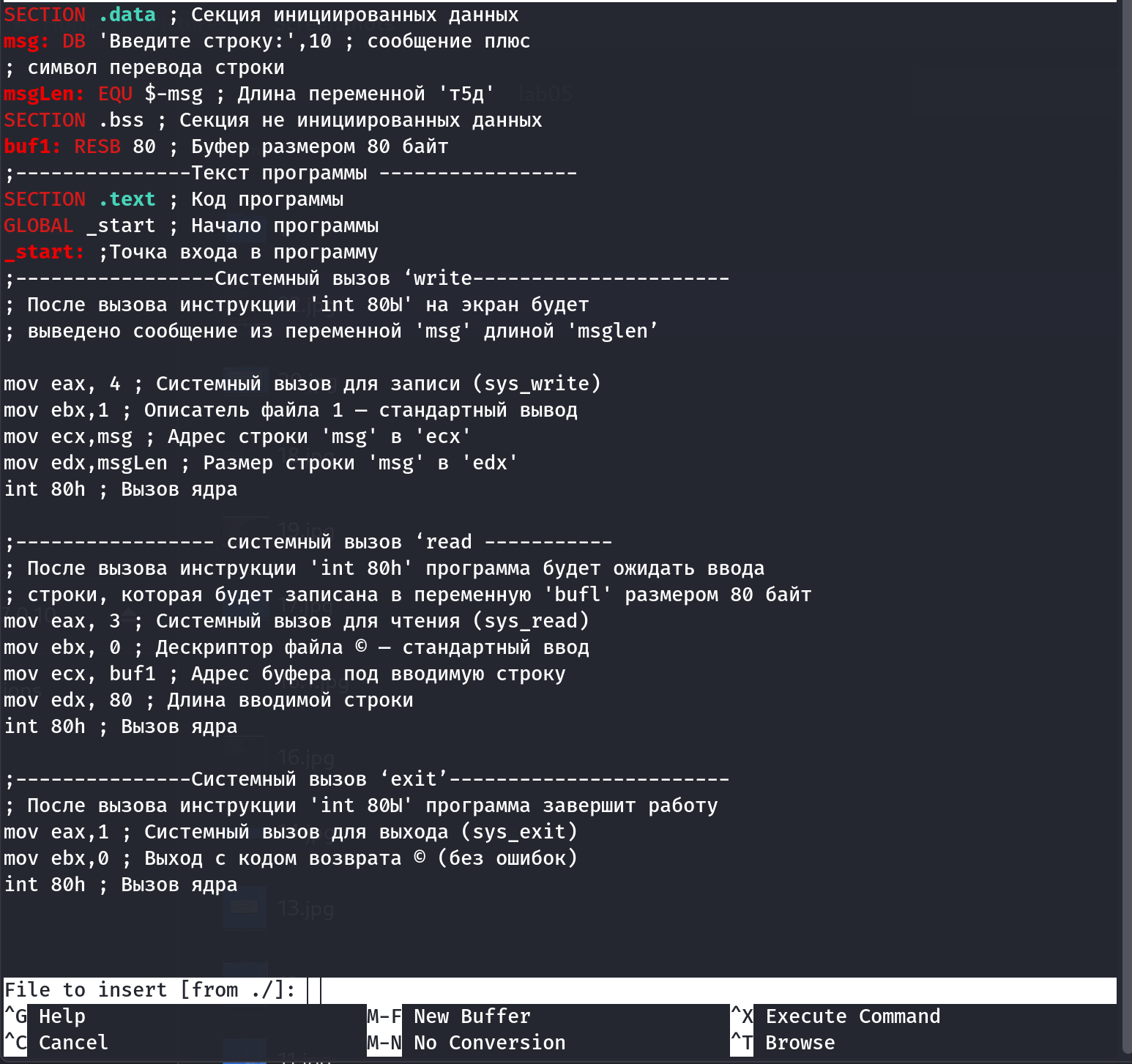
# 5 Выполнение заданий для самостоятельной работы

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



20. Копирование файла

1. Открываю его с помощью mcedit

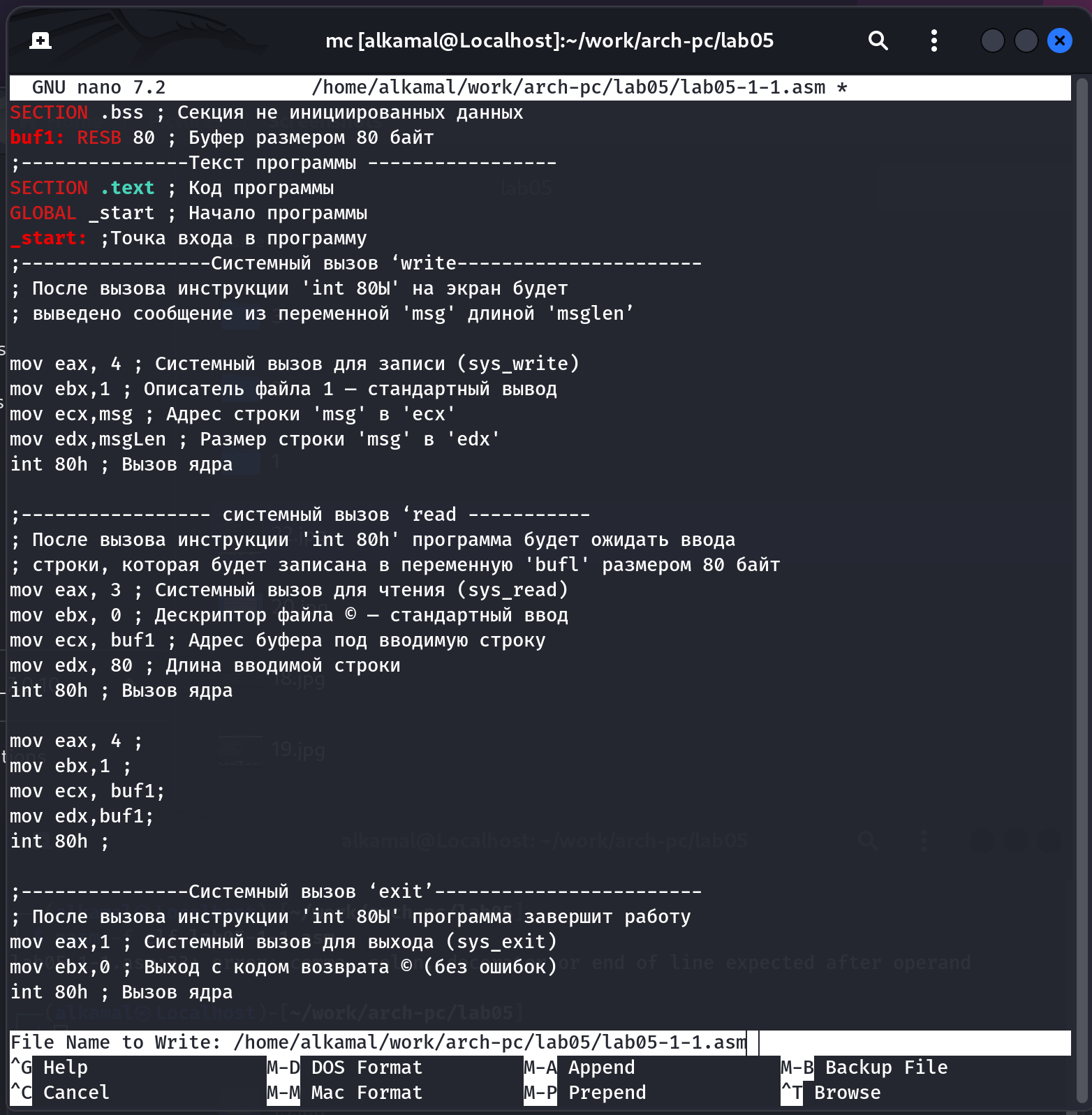


21. Открытый файл

1. Вношу изменения так, чтобы программа возвращала введённое значение

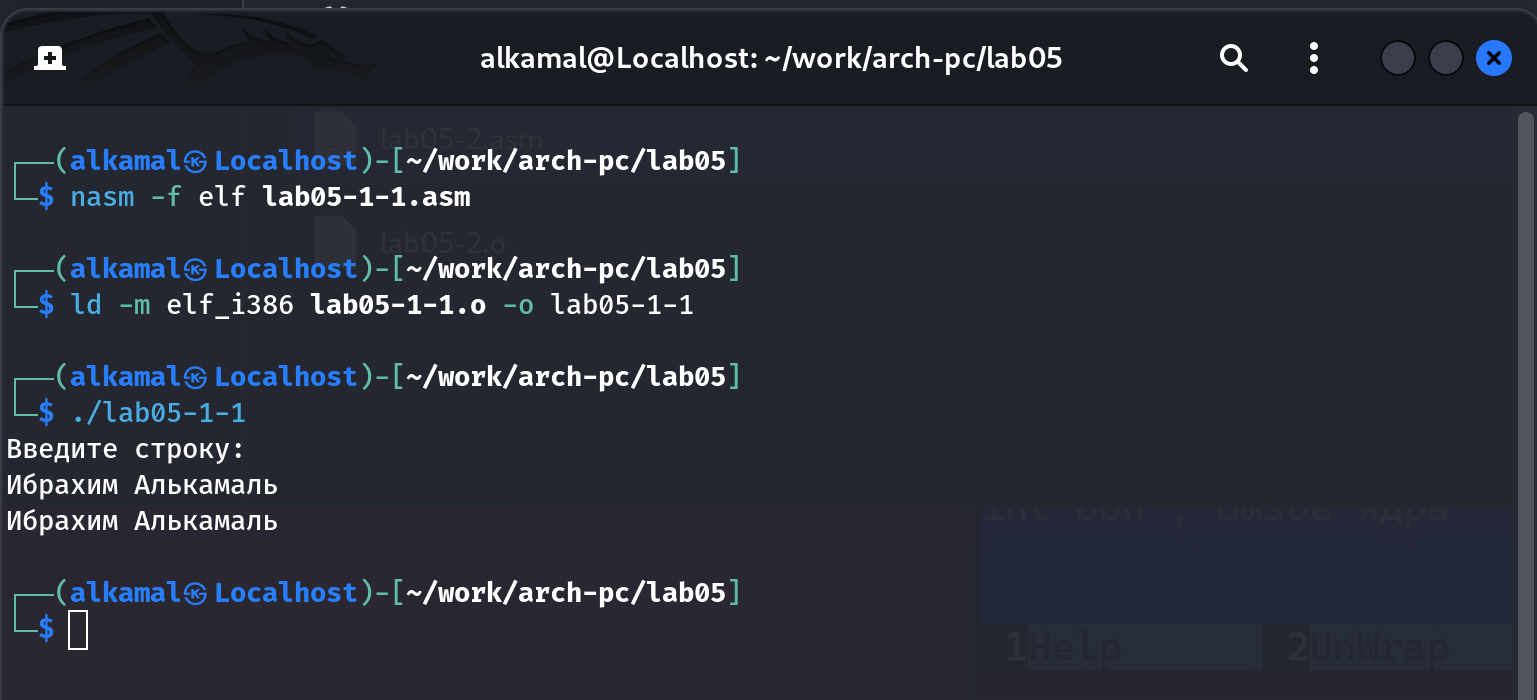
код программы:

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  
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



22. Код исправленной программы

1. Создаю исполняемый файл и запускаю программу



23. Работа файла

# 6 Выводы

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

# Список литературы