

Лабораторная работа № 5

ЧжуЖуйи

9 ноября 2025 г.

Цель работы

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

Порядок выполнения лабораторной работы

1. Откройте Midnight Commander(рис.1)

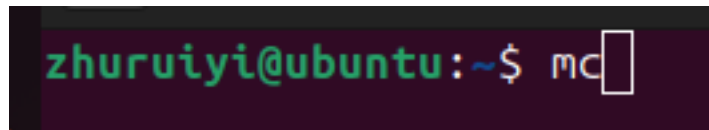


Рис. 1: Откройте Midnight Commander

2. Пользуясь клавишами ↑ , ↓ и Enter перейдите в каталог ~/work/arch-pc созданный при выполнении лабораторной работы №4 .(рис.2)

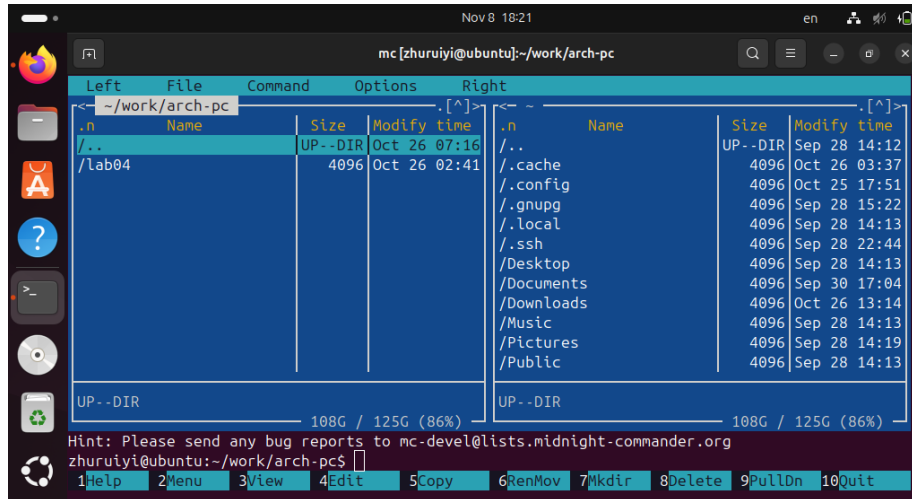


Рис. 2: Окно Midnight Commander. Смена текущего каталога

3. С помощью функциональной клавиши F7 создайте папку lab05 и перейдите в созданный каталог.(рис.3)

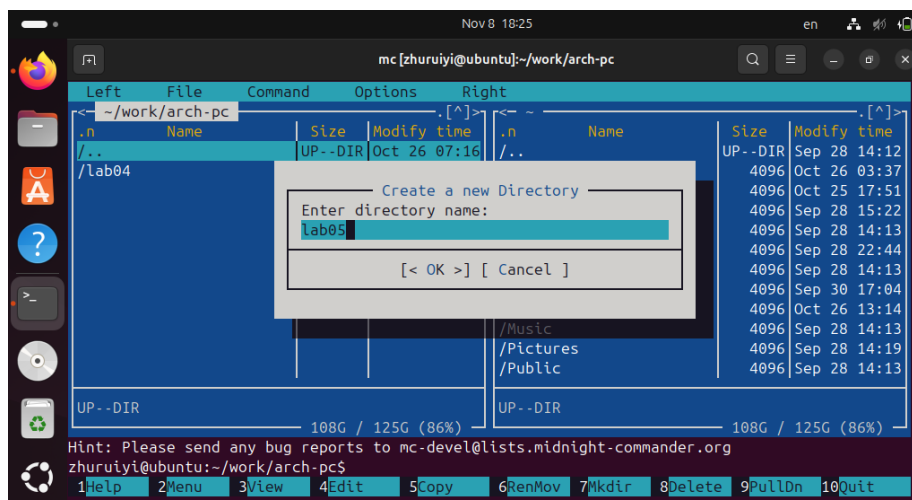


Рис. 3: Окно Midnight Commander. Создание каталога

4. Пользуясь строкой ввода и командой touch создайте файл lab5-1.asm .(рис.4)

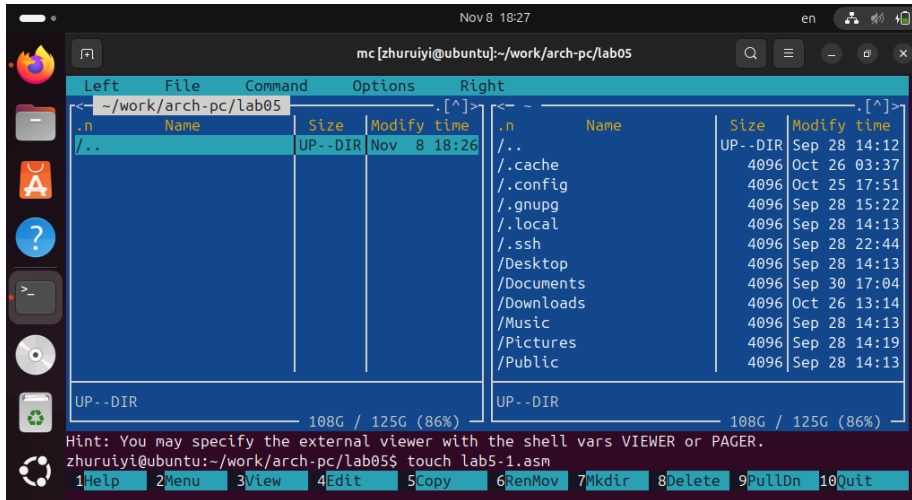
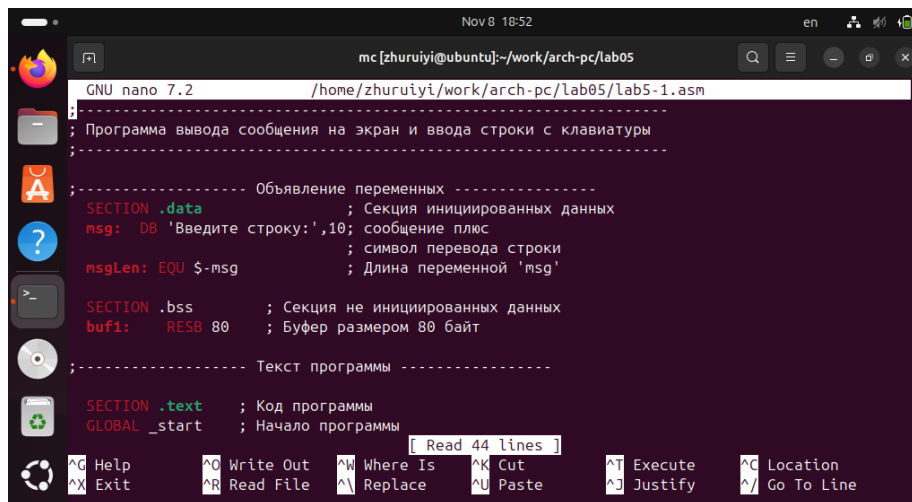


Рис. 4: Окно Midnight Commander. Создание файла

5. С помощью функциональной клавиши F4 откройте файл lab5-1.asm для редактирования во встроенном редакторе.

6. Введите текст программы из листинга 5.1 (можно без комментариев), сохраните изменения и закройте файл.(рис.5)



```
GNU nano 7.2 /home/zhuruiyi/work/arch-pc/lab05/lab5-1.asm
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
;----- Объявление переменных -----
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
[ Read 44 lines ]
```

Рис. 5: Окно Midnight Commander. Редактор nano

7. С помощью функциональной клавиши F3 откройте файл lab5-1.asm для просмотра. Убедитесь, что файл содержит текст программы.(рис.6)

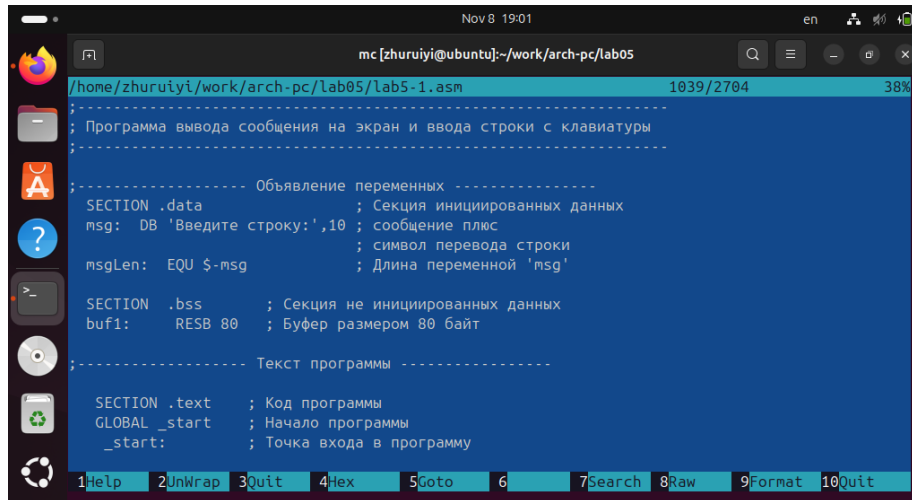


Рис. 6: откройте файл lab5-1.asm для просмотра. Убедитесь, что файл содержит текст программы.

8. Оттранслируйте текст программы lab5-1.asm в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл. Программа выводит строку 'Введите строку:' и ожидает ввода с клавиатуры. На запрос введите Ваши ФИО.(рис.7)

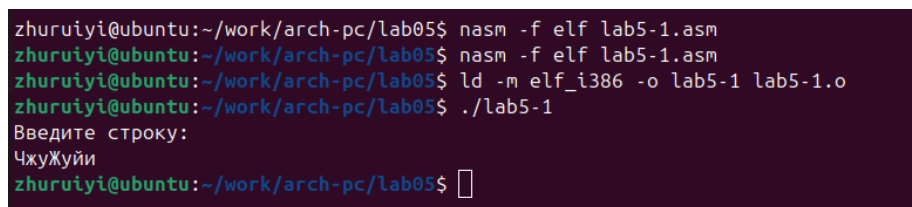


Рис. 7: Выполните

```
nasm -f elf lab5-1.asm
ld -m elf_i386 -o lab5-1 lab5-1.o
./lab5-1
```

Введите строку:
ЧжуЖуйи

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. В одной из панелей mc откройте каталог с файлом lab5-1.asm. В другой панели каталог со скаченным файлом in_out.asm (для перемещения между панелями используйте Tab). Скопируйте файл in_out.asm в каталог с файлом lab5-1.asm с помощью функциональной клавиши F5 .(рис.8)

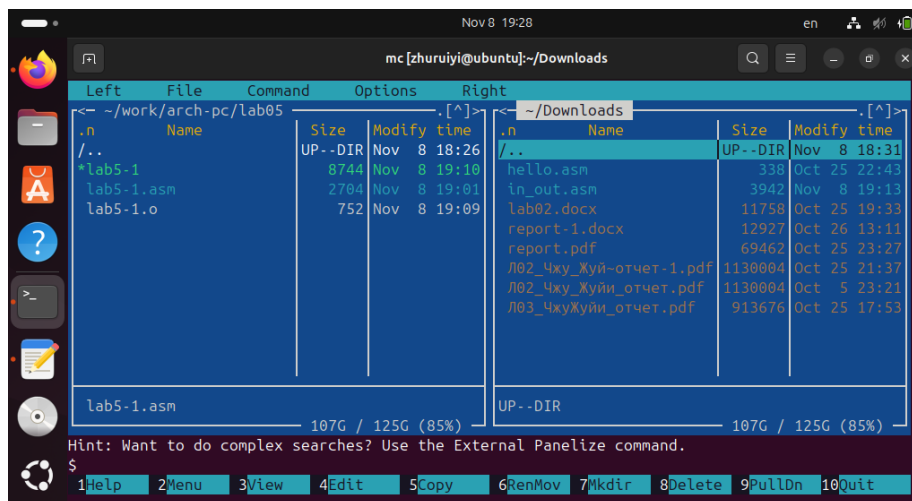


Рис. 8: Окно Midnight Commander. Копирование файла

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.(рис.9)

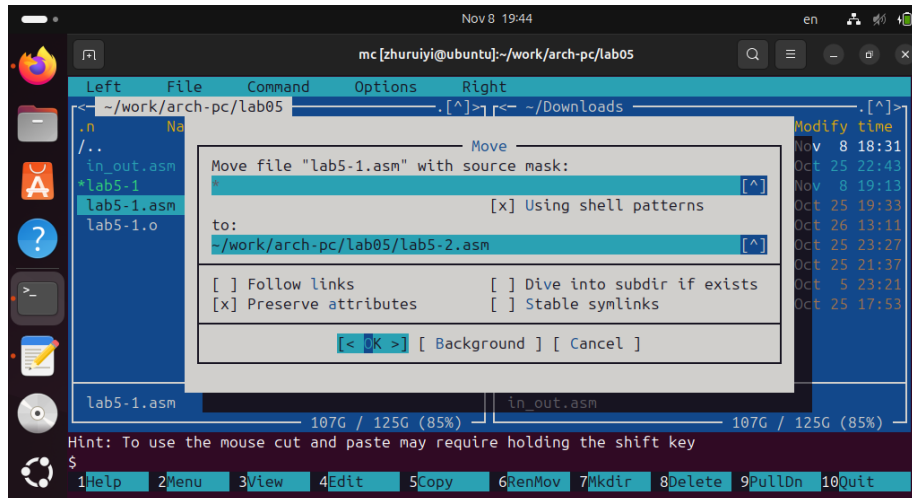


Рис. 9: Окно Midnight Commander. Создание копии файла

12. Исправьте текст программы в файле lab5-2.asm с использование подпрограмм из внешнего файла in_out.asm (используйте подпрограммы sprintf, sread и quit) в соответствии с листингом 5.2. Создайте исполняемый файл и проверьте его работу.(рис.10)

```

GNU nano 7.2 /home/zhuruiyi/work/arch-pc/lab05/lab5-2.asm
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
#include 'in_out.asm' ; подключение внешнего файла

SECTION .data ; Секция инициированных данных
msg: DB 'Введите строку: ',0h ; сообщение

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, msg ; запись адреса выводимого сообщения в `EAX`
call sprintf ; вызов подпрограммы печати сообщения

^O Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line

```

Рис. 10: in_out.asm

```

;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----

#include 'in_out.asm' ; подключение внешнего файла

SECTION .data ; Секция инициированных данных
msg: DB 'Введите строку: ',0h ; сообщение

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, msg ; запись адреса выводимого сообщения в `EAX`
call sprintf ; вызов подпрограммы печати сообщения

```



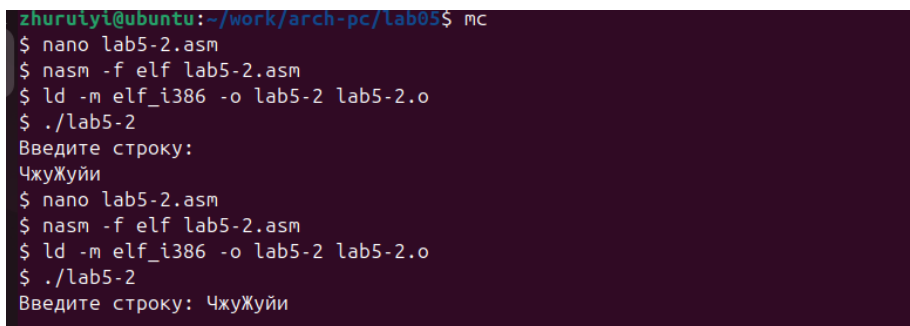
```

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

```

13. В файле lab5-2.asm замените подпрограмму `sprintLF` на `sprint`. Создайте исполняемый файл и проверьте его работу. В чем разница?(рис.11)



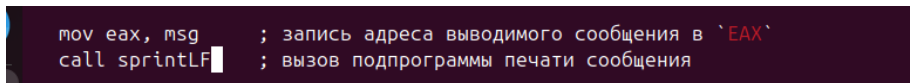
```

zhuruiyi@ubuntu:~/work/arch-pc/lab05$ mc
$ nano lab5-2.asm
$ nasm -f elf lab5-2.asm
$ ld -m elf_i386 -o lab5-2 lab5-2.o
$ ./lab5-2
Введите строку:
ЧжуЖуйи
$ nano lab5-2.asm
$ nasm -f elf lab5-2.asm
$ ld -m elf_i386 -o lab5-2 lab5-2.o
$ ./lab5-2
Введите строку: ЧжуЖуйи

```

Рис. 11: разница

`sprintLF`: Автоматический перенос строки после выводимой информации.(рис.12)



```

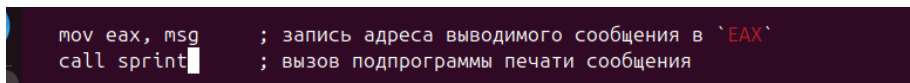
mov eax, msg      ; запись адреса выводимого сообщения в `EAX`
call sprintLF     ; вызов подпрограммы печати сообщения

```

Рис. 12: `sprintLF`

Введите строку:
[Курсор здесь, следующая строка.]

`sprint`: Нет новой строки после выходной информации.(рис.13)



```

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

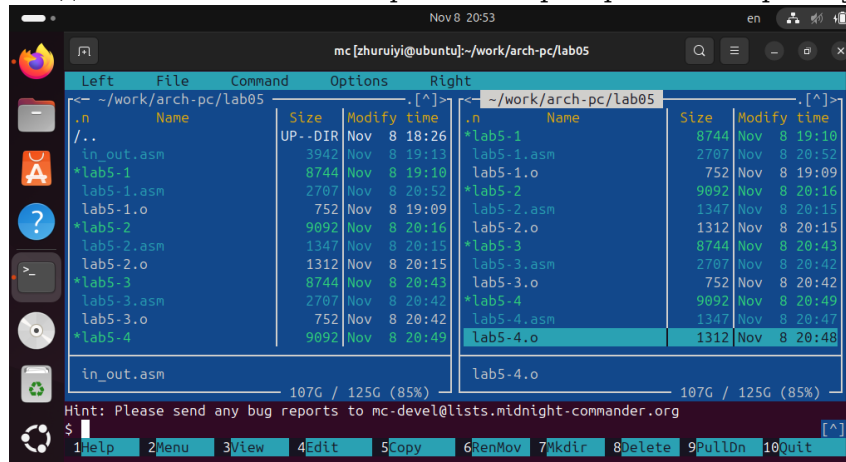
```

Рис. 13: `sprint`

Введите строку: [Курсор находится здесь, на той же строке.]

Задание для самостоятельной работы

1. Создайте копию файла lab5-1.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 введенную строку на экран.
2. Получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите свою фамилию.
3. Создайте копию файла lab5-2.asm. Исправьте текст программы с использованием подпрограмм из внешнего файла in_out.asm, так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введенную строку на экран. Не забудьте, подключаемый файл in_out.asm должен лежать в том же каталоге, что и файл с программой, в которой он используется.
4. Создайте исполняемый файл и проверьте его работу.



```
$ ./lab5-2
Введите строку: ЧжуЖуйи
$ ./lab5-1
Введите строку:
ЧжуЖуйи
$ cp lab5-1.asm lab5-3.asm
cp: cannot stat 'lab5-1.asm': No such file or directory
$ nano lab5-3.asm
$ nasm -f elf lab5-3.asm
$ ld -m elf_i386 -o lab5-3 lab5-3.o
$ ./lab5-3
Введите строку:
ЧжуЖуйи
$ cp lab5-2.asm lab5-4.asm
$ nano lab5-4.asm
$ nasm -f elf lab5-4.asm
$ ld -m elf_i386 -o lab5-4 lab5-4.o
$ ./lab5-4
Введите строку: ЧжуЖуйи
```

Рис. 14: ответ

Вывод:

В ходе выполнения лабораторной работы №5 мы научились уверенно использовать Midnight Commander для управления файлами и освоили написание и отладку программ на ассемблере NASM. Мы получили хорошее понимание различий между системными вызовами и библиотечными функциями, а также приобрели способность выполнять полный цикл разработки ассемблерных программ.