

# **Лабораторная работа №5**

**Основы работы с Midnight Commander (mc). Структура программы  
на языке ассемблера NASM. Системные вызовы в ОС GNU Linux**

Акунаева Антонина Эрдниевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
3.1	Midnight Commander . . . . .	7
3.2	Подключение внешнего файла in_out.asm . . . . .	16
<b>4</b>	<b>Описание результатов выполнения заданий для самостоятельной работы</b>	<b>22</b>
<b>5</b>	<b>Выводы</b>	<b>29</b>

# Список иллюстраций

3.1	Открытие Midnight Commander . . . . .	7
3.2	Midnight Commander . . . . .	7
3.3	Перемещение по МС. Каталог ~/work/ . . . . .	8
3.4	Перемещение по МС. Каталог ~/work/arch-pc/ . . . . .	9
3.5	Создание каталога в МС . . . . .	10
3.6	Созданный в МС каталог . . . . .	10
3.7	Использование touch . . . . .	11
3.8	Созданный в Midnight Commander файл . . . . .	11
3.9	МС. Mcedit . . . . .	12
3.10	МС. Mcedit - Ввод листинга 5.1 . . . . .	13
3.11	МС. Mcedit - Ввод листинга 5.1 . . . . .	14
3.12	МС. Просмотр файла . . . . .	15
3.13	Трансляция и компоновка исполняемого файла NASM . . . . .	16
3.14	Скачанный файл in_out.asm . . . . .	16
3.15	МС. Панели с файлами lab5-1.asm и in_out.asm . . . . .	17
3.16	МС. Копирование in_out.asm в ~/work/arch-pc/lab05/ . . . . .	18
3.17	МС. Файл in_out.asm в ~/work/arch-pc/lab05/ . . . . .	18
3.18	МС. Создание копии lab5-1.asm . . . . .	19
3.19	МС. Mcedit lab5-2.asm . . . . .	20
3.20	Трансляция и компоновка исполняемого файла lab5-2 . . . . .	20
3.21	Изменённая строка sprint . . . . .	21
3.22	Запуск изменённого исполняемого файла lab5-2 . . . . .	21
4.1	МС. Копирование файла lab5-1.asm . . . . .	23
4.2	МС. Mcedit. Файл lab5-3.asm . . . . .	24
4.3	Создание и проверка работы исполняемого файла lab5-3 . . . . .	25
4.4	МС. Копирование файла lab5-2.asm . . . . .	26
4.5	МС. Mcedit. Файл lab5-4.asm . . . . .	27
4.6	Создание и проверка работы исполняемого файла lab5-4 . . . . .	28

## **Список таблиц**

# 1 Цель работы

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

## 2 Задание

Научиться работать с файлами и каталогами в Midnight Commander.

Освоить пользование Midnight Commander при помощи функциональных клавиш.

Освоить NASM и инструкции языка ассемблера `mov`, `int`.

# 3 Выполнение лабораторной работы

## 3.1 Midnight Commander

3.1.1. Откройте Midnight Commander.

```
aeakunaeva@fedora:~$ mc
```

Рис. 3.1: Открытие Midnight Commander

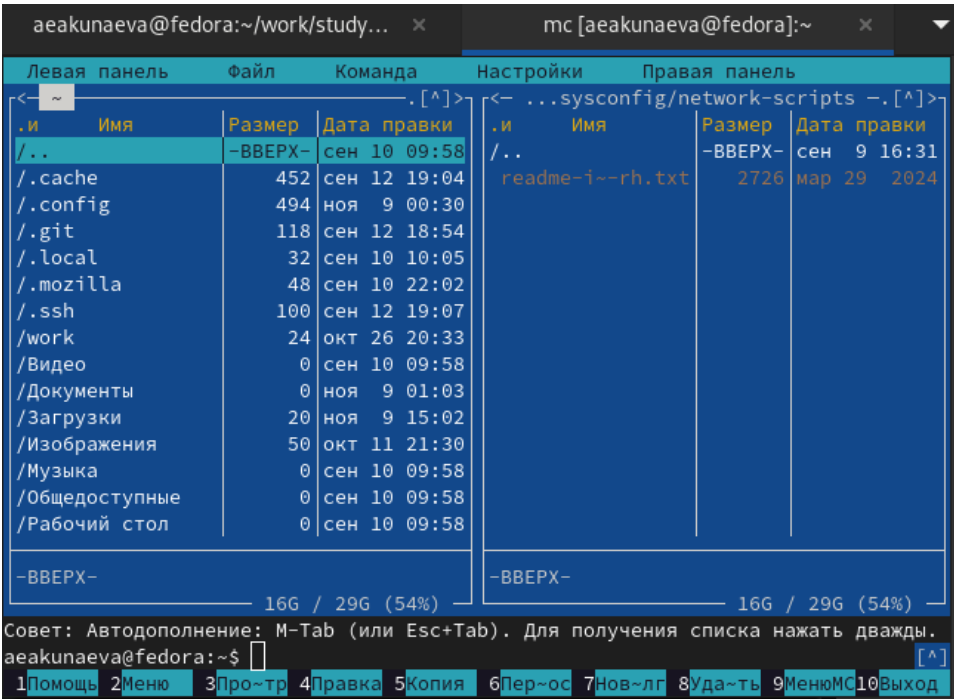


Рис. 3.2: Midnight Commander

Откроем МС при помощи команды mc в терминале.

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

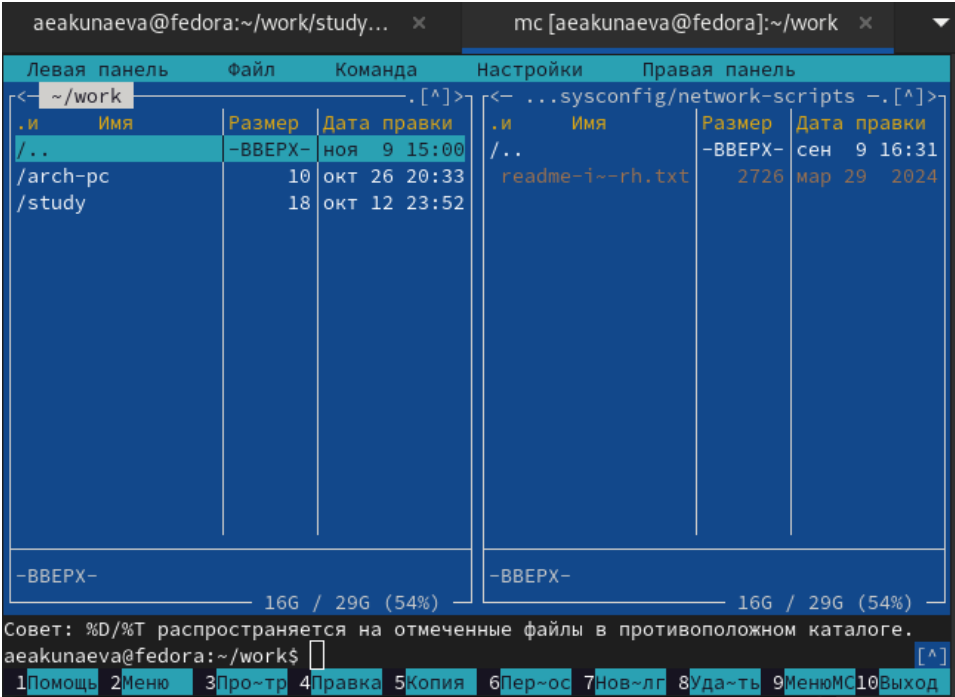


Рис. 3.3: Перемещение по МС. Каталог ~/work/



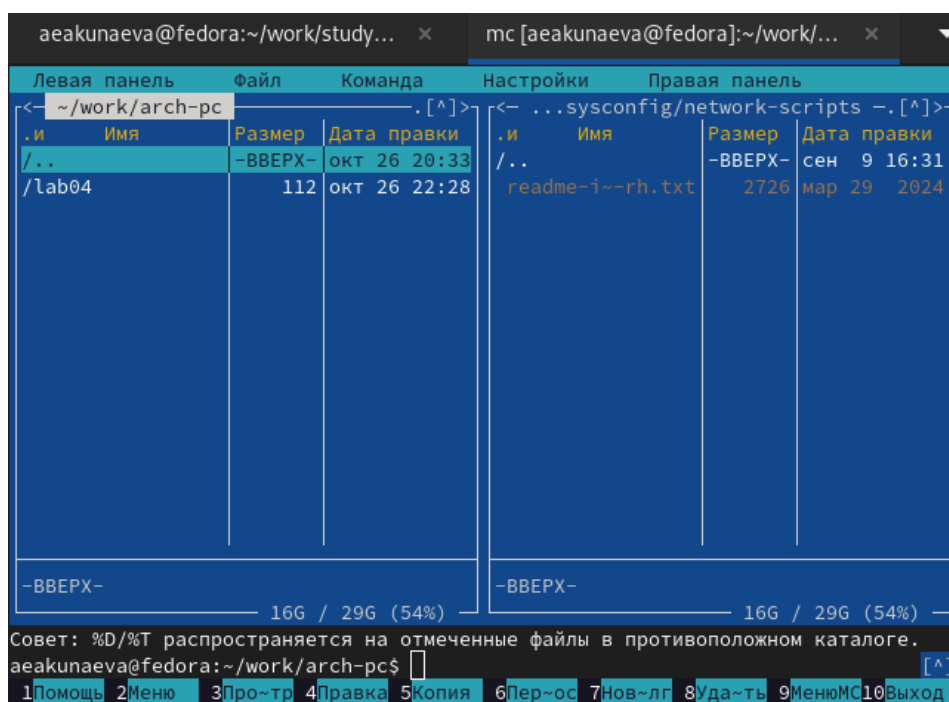


Рис. 3.4: Перемещение по МС. Каталог ~/work/arch-pc/

Переходим в требуемый каталог при помощи горячих клавиш.

3.1.3. С помощью функциональной клавиши F7 создайте папку lab05 и перейдите в созданный каталог.

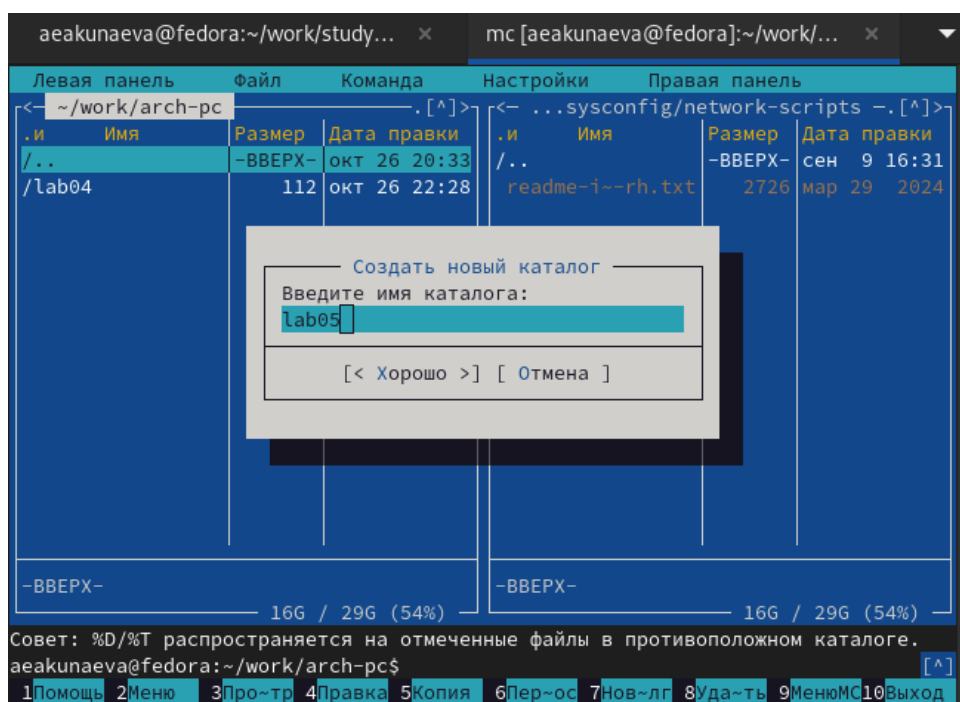


Рис. 3.5: Создание каталога в МС

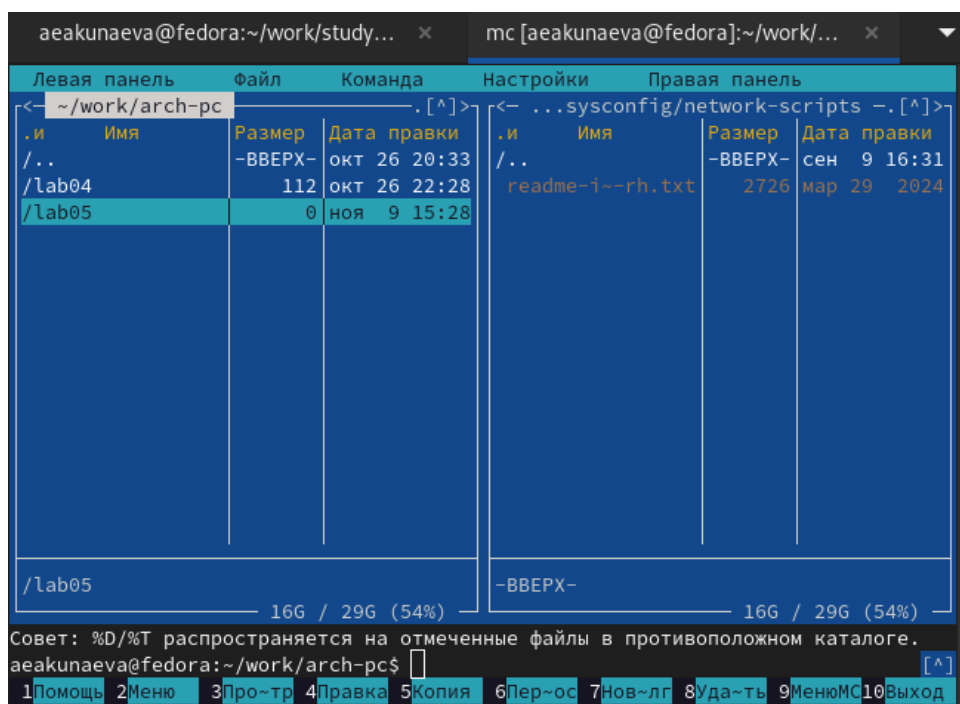
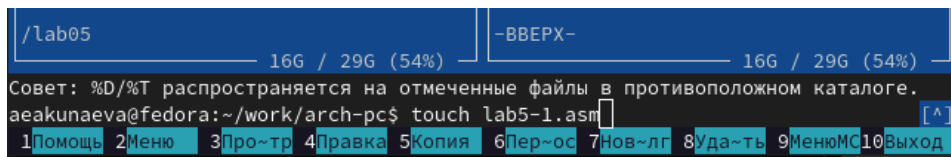


Рис. 3.6: Созданный в МС каталог

Горячей клавишей F7 создадим каталог и введём его название lab05.

3.1.4. Пользуясь строкой ввода и командой touch создайте файл lab5-1.asm.



```
/lab05 16G / 29G (54%) -ВВЕРХ- 16G / 29G (54%)  
Совет: %D/%T распространяется на отмеченные файлы в противоположном каталоге.  
aeakunaeva@fedora:~/work/arch-pc$ touch lab5-1.asm  
1Помощь 2Меню 3Про-тр 4Правка 5Копия 6Пер-ос 7Нов-лг 8Уда-ть 9МенюМС10Выход
```

Рис. 3.7: Использование touch

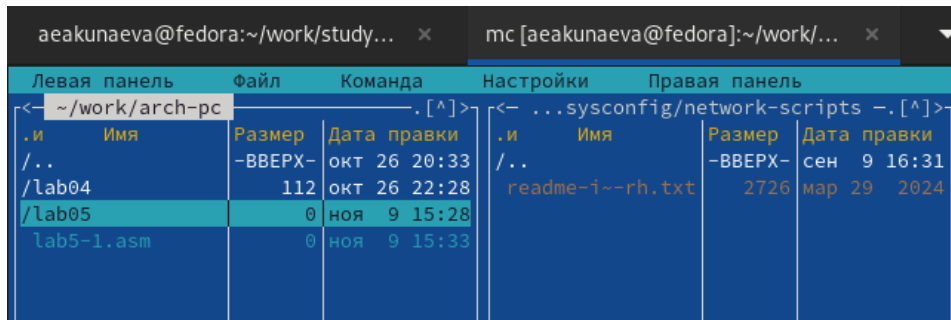


Рис. 3.8: Созданный в Midnight Commander файл

При помощи команды touch в текущей директории в терминала окна Midnight Commander создадим файл lab5-1.asm и убедимся в его наличии, проверив визуально в МС.

3.1.5. 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.

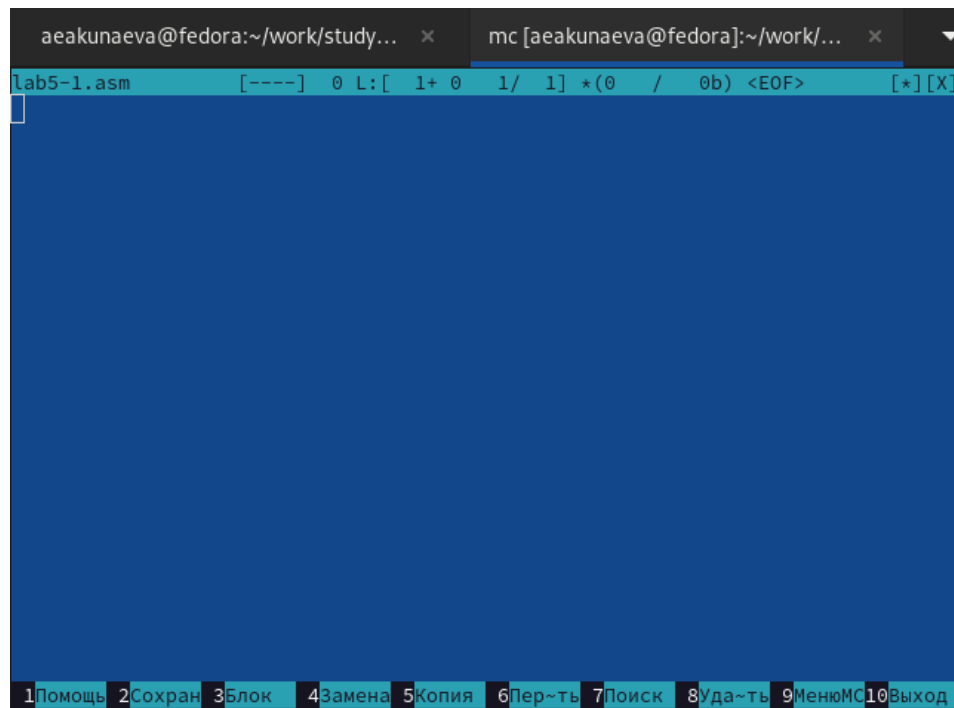


Рис. 3.9: MC. Mcedit

Горячей клавишей F4 откроем файл во встроенном редакторе, по умолчанию на данном устройстве - в mcedit.

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

```
aeakunaeva@fedora:~/work/stud... x mc [aeakunaeva@fedora]:~/work... x
lab5-1.asm [-M--] 32 L:[ 1+ 0 1/ 45] *(32 /2509b) 0045 0x02D [*] [X]
;-----
; 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
1По~щъ 2Сохран 3Блок 4Замена 5Копия 6Пер~ть 7Поиск 8Уда~ть 9МенюMC10Выход
```

Рис. 3.10: MC. Mcedit - Ввод листинга 5.1

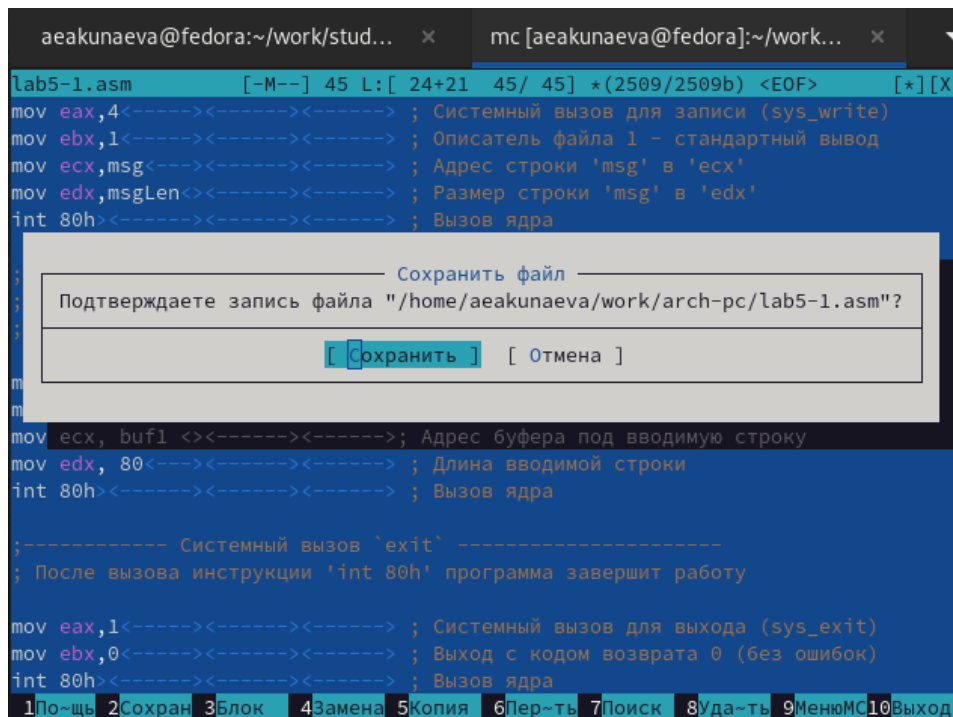


Рис. 3.11: MC. Mcedit - Ввод листинга 5.1

Вставим текст листинга 5.1 из лабораторной в mcedit при помощи зажатой SHIFT + средней кнопки мыши (колёсико). Сохраним изменения горячей клавишей F2 и выйдем из редактора с F10.

3.1.7. С помощью функциональной клавиши F3 откройте файл lab5-1.asm для просмотра. Убедитесь, что файл содержит текст программы.

```
aeakunaeva@fedora:~/work/stu... x mc [aeakunaeva@fedora]:~/wo... x
/home/aeakunaeva/work/arch-pc/lab5-1.asm 2509/2509 100%
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
;----- 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

;----- 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. 3.12: МС. Просмотр файла

Нажмём F3 и откроем режим просмотра файла. Проверим корректность проведённых операций и текста листинга 5.1.

3.1.8. Оттранслируйте текст программы lab5-1.asm в объектный файл. Выпол-

ните компоновку объектного файла и запустите получившийся исполняемый файл. Программа выводит строку 'Введите строку:' и ожидает ввода с клавиатуры. На запрос введите Ваши ФИО.

```
aeakunaeva@fedora:~/work/arch-pc$ nasm -f elf lab5-1.asm
aeakunaeva@fedora:~/work/arch-pc$ ld -m elf_i386 -o lab5-1 lab5-1.o
aeakunaeva@fedora:~/work/arch-pc$ ./lab5-1
Введите строку:
Акунаева Антонина Эрдниева
aeakunaeva@fedora:~/work/arch-pc$
```

Рис. 3.13: Трансляция и компоновка исполняемого файла NASM

Проведём трансляцию файла lab5-1.asm в объектный, скомпилируем его в elf для создания исполняемого файла и запускаем его через обращение ./lab5-1. Объектный и исполняемый файлы сохраняют название .asm файла. После успешного запуска в терминале выводится запрос, на что вводим ФИО.

## 3.2 Подключение внешнего файла in\_out.asm

3.2.1. Скачайте файл in\_out.asm со страницы курса в ТУИС.

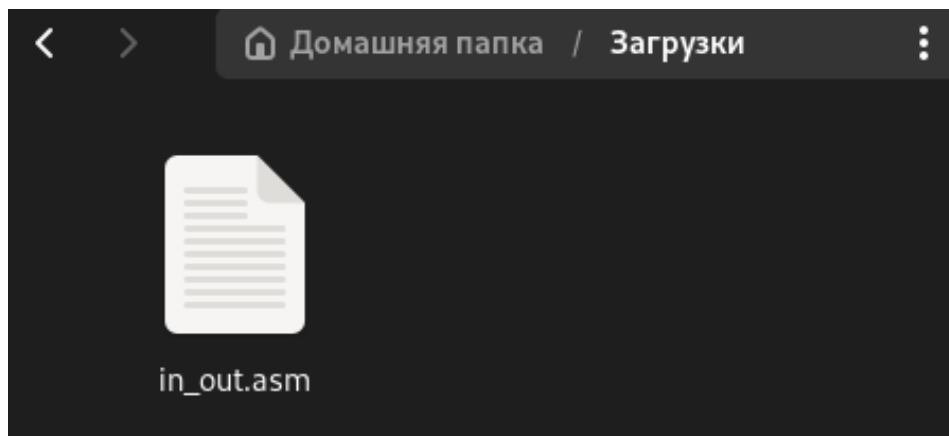


Рис. 3.14: Скачанный файл in\_out.asm

Скачиваем файл с ТУИС. Файл расположен в Загрузках.

3.2.2. 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.

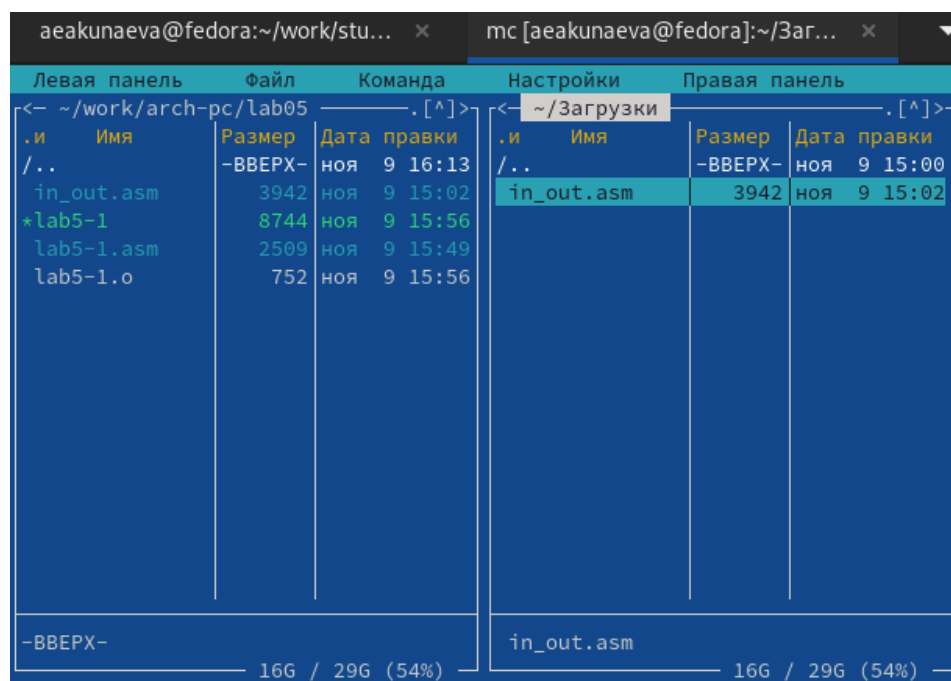


Рис. 3.15: МС. Панели с файлами lab5-1.asm и in\_out.asm

При помощи горячих клавиш перейдём в рабочий каталог ~/work/arch-pc/lab05/ и в каталог с файлом in\_out.asm ~/Загрузки.

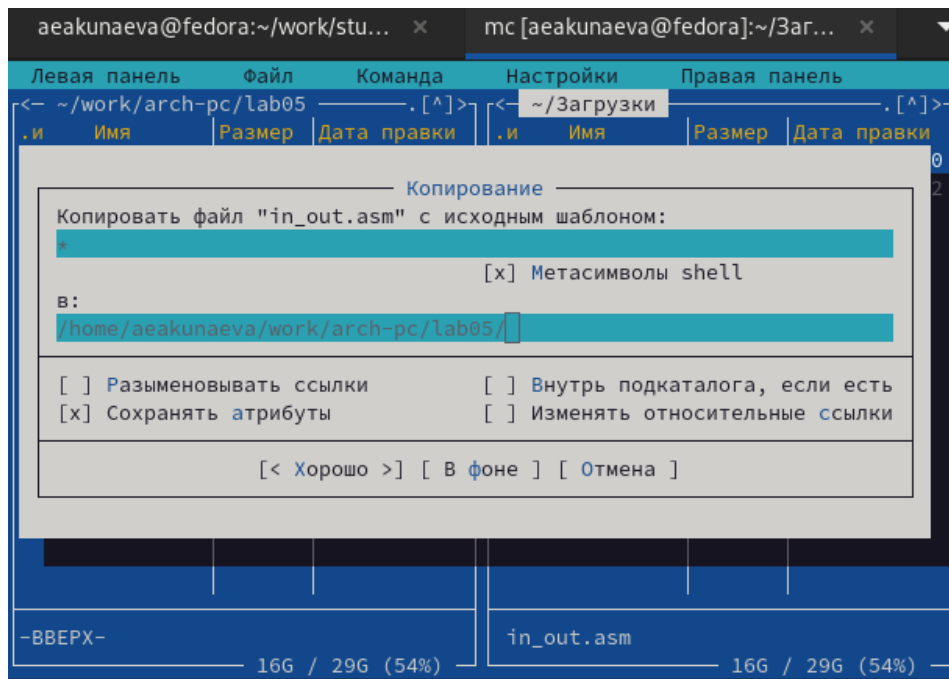


Рис. 3.16: MC. Копирование in\_out.asm в ~/work/arch-pc/lab05/

Функциональной клавишей F5 скопируем файл in\_out.asm в рабочий каталог, который выбран в левой панели в Midnight Commander.

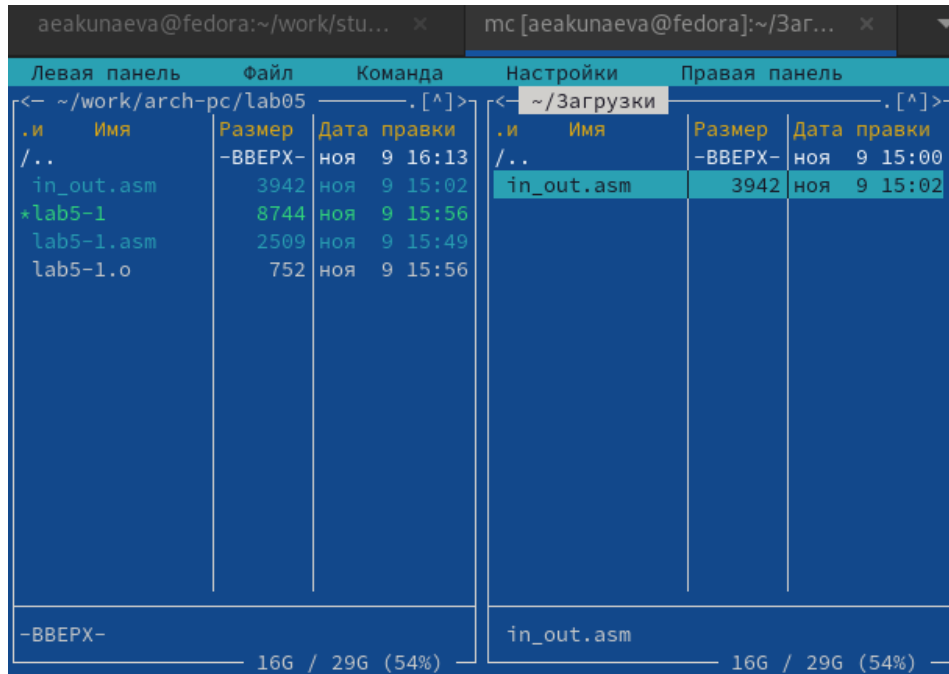


Рис. 3.17: MC. Файл in\_out.asm в ~/work/arch-pc/lab05/

После подтверждения действия видим файл in\_out.asm в левой панели с рабочим каталогом, где расположен файл lab5-1.asm.

3.2.3. 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.

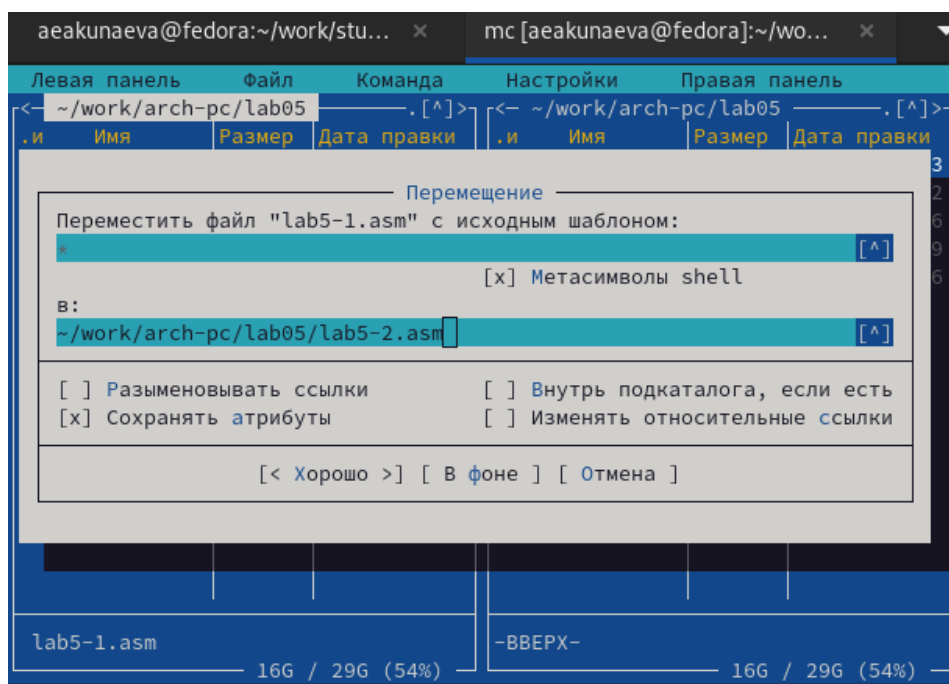


Рис. 3.18: MC. Создание копии lab5-1.asm

Функциональной клавишей F6 копируем файл lab5-1.asm в рабочий каталог (та же директория) с указанием нового имени для файла.

3.2.4. 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.

```

aeakunaeva@fedora:~/work/stu...  x  mc [aeakunaeva@fedora]:~/wo...  x
lab5-2.asm  [----]  0 L: [ 1+ 8  9/ 25]  *(482 /1298b)  0010 0x0[*] [X]
-----
; 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 read<-><-----> ; вызов подпрограммы ввода сообщения
1По-щ 2Со-ан 3Блок 4Замена 5Копия 6Пе-ть 7Поиск 8Уда-ть 9МенюMC10Выход

```

Рис. 3.19: MC. Mcedit lab5-2.asm

Откроем текстовый редактор mcedit в Midnight Commander с файлом lab5-2.asm и исправим исходный текст в соответствии с листингом 5.2. Сохраним при помощи функциональной клавиши F2 и выйдем из редактора с F10.

```

aeakunaeva@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
aeakunaeva@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
aeakunaeva@fedora:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
-
aeakunaeva@fedora:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
-
aeakunaeva@fedora:~/work/arch-pc/lab05$ ls
in_out.asm  lab5-1  lab5-1.asm  lab5-1.o  lab5-2  lab5-2.asm  lab5-2.o
aeakunaeva@fedora:~/work/arch-pc/lab05$

```

Рис. 3.20: Трансляция и компоновка исполняемого файла lab5-2

Оттранслируем lab5-2.asm в объектный файл, скомпонуем и создадим исполняемый файл в elf. Запустим lab5-2. Результат идентичен lab5-1.

3.2.5. В файле lab5-2.asm замените подпрограмму sprintf на sprint. Создайте

исполняемый файл и проверьте его работу. В чем разница?

```
mov eax, msg <-----> ; запись адреса выводимого сообщения в `EAX`  
call sprint > ; вызов подпрограммы печати сообщения  
....  
mov ecx, buf1 <-----> ; запись адреса переменной в `EAX`  
mov edx, 80 <-----> ; запись длины выводимого сообщения в `EDX`
```

Рис. 3.21: Изменённая строка sprint

Изменим строку с sprintLF на sprint.

```
aeakunaeva@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm  
aeakunaeva@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o  
aeakunaeva@fedora:~/work/arch-pc/lab05$ ./lab5-2  
Введите строку: Акунаева Антонина Эрдниевна  
aeakunaeva@fedora:~/work/arch-pc/lab05$ ./lab5-1  
Введите строку:  
Акунаева Антонина Эрдниевна  
aeakunaeva@fedora:~/work/arch-pc/lab05$
```

Рис. 3.22: Запуск изменённого исполняемого файла lab5-2

Снова оттранслируем lab5-2.asm в объектный файл, скомпилируем и создадим исполняемый файл в elf. Запустим lab5-2. В отличие от lab5-1, запрос lab5-2 отличается отсутствием переноса строки после фразы, что обусловлено разницей между командами sprintLF и sprint - первая содержит перенос строки после вывода текста (LF), вторая - нет.

## **4 Описание результатов выполнения заданий для самостоятельной работы**

4.1. 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 введённую строку на экран.

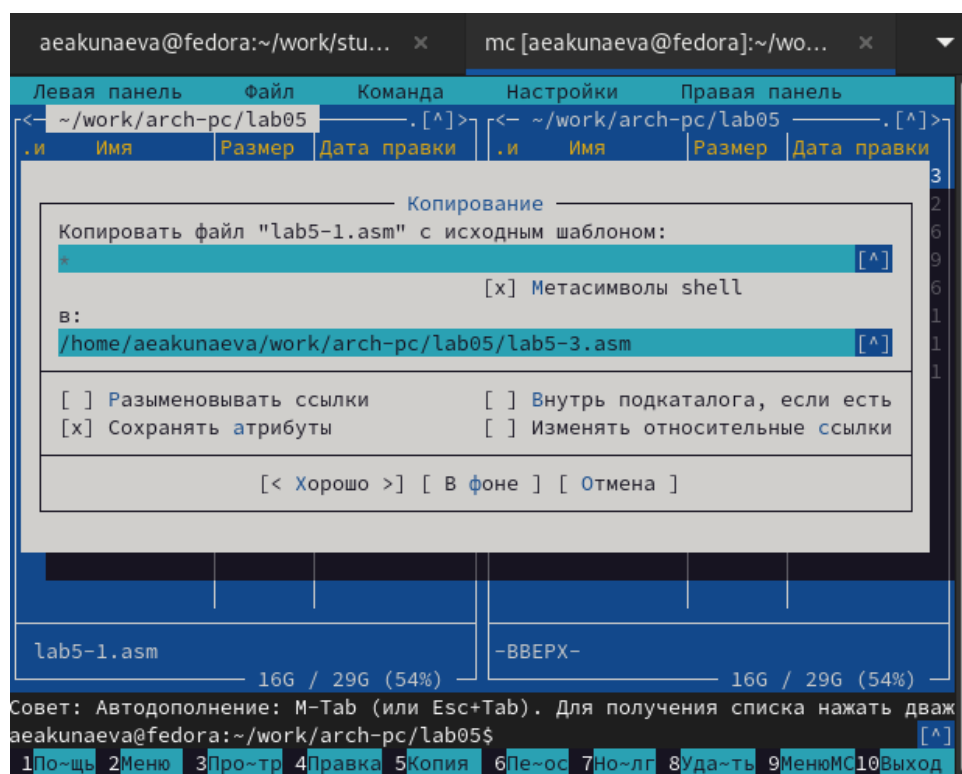


Рис. 4.1: МС. Копирование файла lab5-1.asm

Скопируем файл lab5-1.asm при помощи функциональной клавиши F5 и переименуем его в строке пункта назначения в lab5-3.asm.





текст из lab5-1.asm и изменяем/дополняем его в соответствии с требованиями, используя операнды mov, int:

mov eax,4 - системный вызов для записи sys\_write;

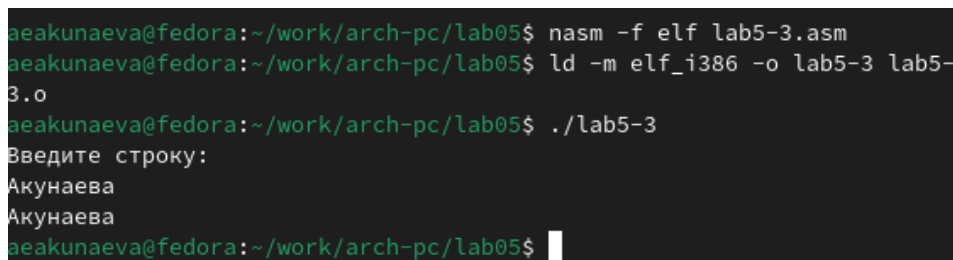
mov ebx,1 - дескриптор файла 1 обозначает стандартный вывод;

mov ecx,buf1 - адрес строки buf1 в ecx. Мы используем здесь переменную вводимой ранее строки buf1 (обозначена в .asm-файлах;

mov edx,80 - размер строки в edx. Мы используем значение 80, то же, которое было указано при вводе строки в адрес buf1;

int 80h - вызов ядра.

4.2. Получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите свою фамилию.



```
aeakunaeva@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-3.asm
aeakunaeva@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-3 lab5-3.o
aeakunaeva@fedora:~/work/arch-pc/lab05$ ./lab5-3
Введите строку:
Акунаева
Акунаева
aeakunaeva@fedora:~/work/arch-pc/lab05$
```

Рис. 4.3: Создание и проверка работы исполняемого файла lab5-3

Оттранслируем lab5-3.asm в объектный файл, скомпилируем и создадим исполняемый файл в elf. Запустим lab5-3. Введём свою фамилию. Как и требовалось, после ввода фамилии она выводится в терминал.

4.3. Создайте копию файла lab5-2.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 введенную строку на экран.

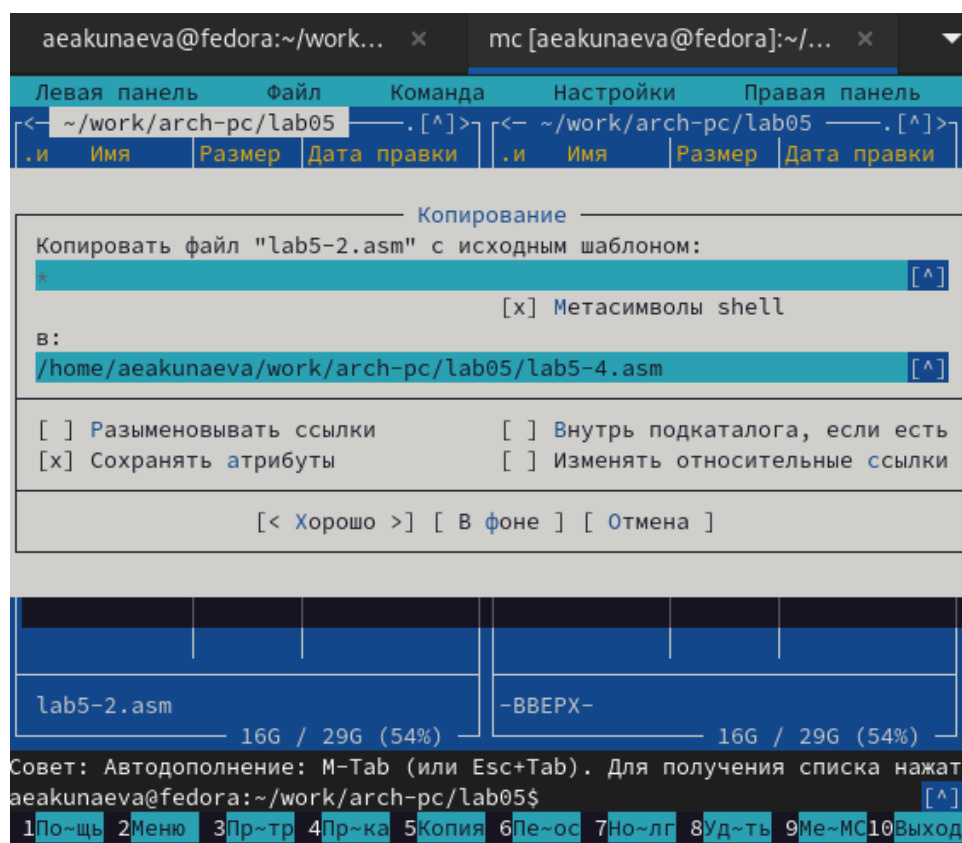
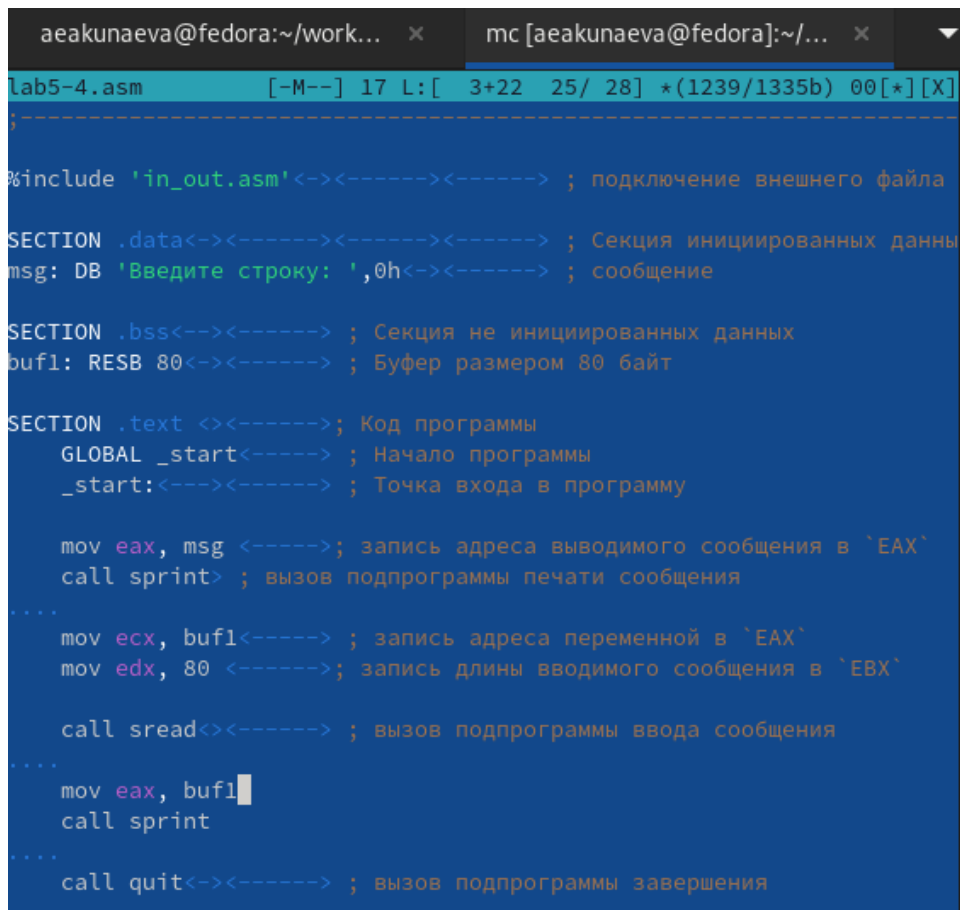


Рис. 4.4: MC. Копирование файла lab5-2.asm

Скопируем файл lab5-2.asm при помощи функциональной клавиши F5 и переименуем его в строке пункта назначения в lab5-4.asm.



```
aeakunaeva@fedora:~/work... x mc [aeakunaeva@fedora]:~/... x
lab5-4.asm [-M--] 17 L:[ 3+22 25/ 28] *(1239/1335b) 00[*] [X]
;-----
%include 'in_out.asm'<-><-----><-----> ; подключение внешнего файла
SECTION .data<-><-----><-----><-----> ; Секция иницированных данны
msg: DB 'Введите строку: ',0h<-><-----> ; сообщение
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, msg <----->; запись адреса выводимого сообщения в `EAX`
call sprint> ; 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
....
mov eax, buf1
call sprint
....
call quit<-><-----> ; вызов подпрограммы завершения
```

Рис. 4.5: MC. Mcedit. Файл lab5-4.asm

Т.к. нам нужно использовать внешний файл in\_out.asm, оставляем исходный текст из lab5-1.asm и изменяем/дополняем его в соответствии с требованиями, используя разные операнды:

mov eax,buf1 - запись адреса выводимого сообщения из переменной buf1 со значением введенной ранее строки в 'EAX';

call sprint - вызов подпрограммы вывода сообщения в терминал.

4.4. Создайте исполняемый файл и проверьте его работу.

```
aeakunaeva@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-4.asm
aeakunaeva@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-4 lab5-4.o
aeakunaeva@fedora:~/work/arch-pc/lab05$ ./lab5-4
Введите строку: Акунаева
Акунаева
aeakunaeva@fedora:~/work/arch-pc/lab05$
```

Рис. 4.6: Создание и проверка работы исполняемого файла lab5-4

Оттранслируем lab5-4.asm в объектный файл, скомпилируем и создадим исполняемый файл в elf. Запустим lab5-4. Введём свою фамилию. Как и требовалось, после ввода фамилии она выводится в терминал.

## 5 Выводы

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