Лабораторная работа No5. Основы работы с Midnight Commander (mc). Структура программы на языке ассемблера NASM. Системные вызовы в ОС GNU Linux

Простейший вариант

Диана Алексеевна Садова

Содержание

# 1 Цель работы

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

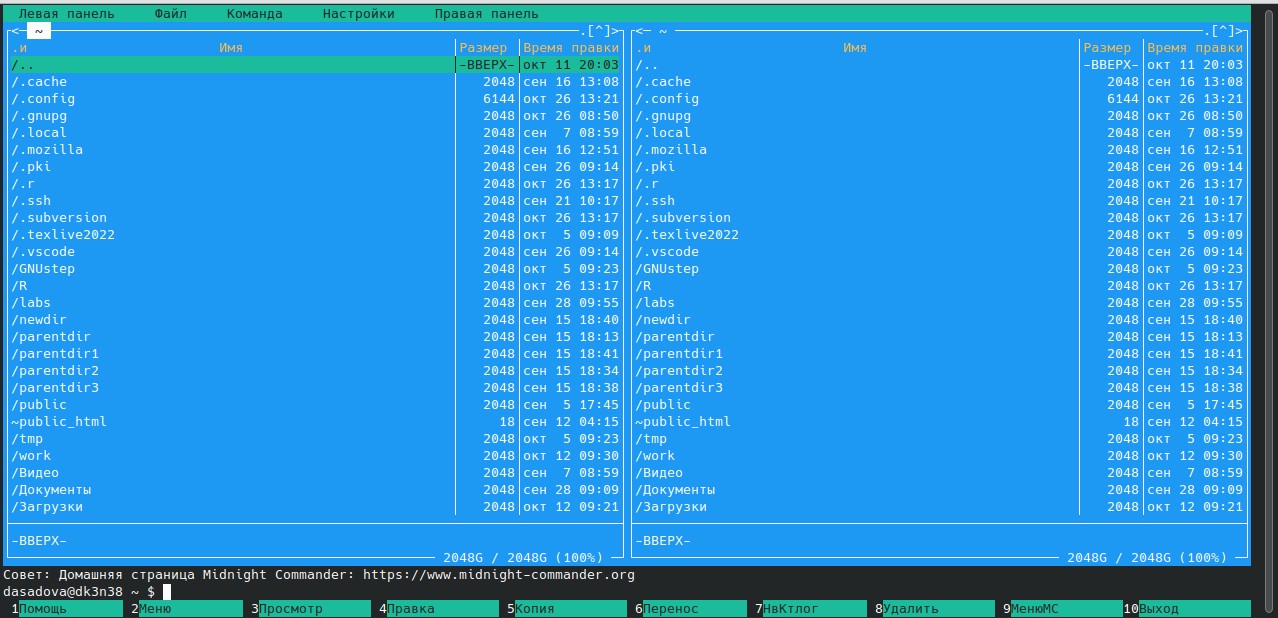
# 2 Задание

## 2.1 Порядок выполнения лабораторной работы

### 2.1.1 Откройте Midnight Commander (рис.2.1)(рис.2.2)

Вводим команду mc

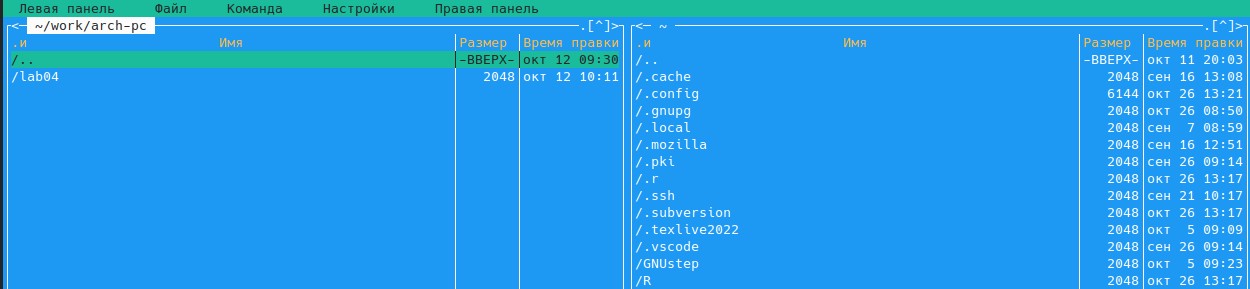
Вводим команду mc



Окно Midnight

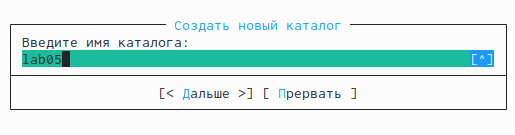
С помощью команды mc мы переходим в Midnight и продолжаем работу

### 2.1.2 Пользуясь клавишами вверх, вниз и Enter перейдите в каталог ~/work/arch-pc созданный при выполнении лабораторной работы No4 (рис.2.3).

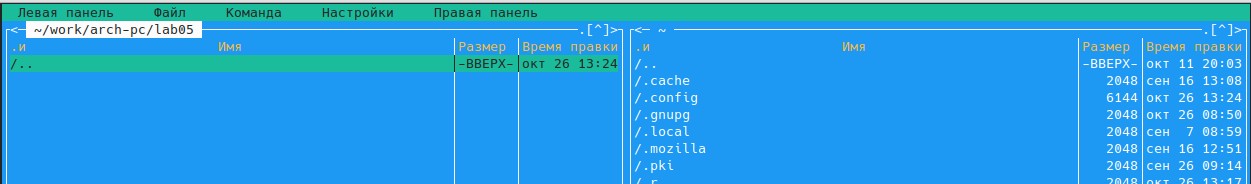


Переходим в каталог ~/work/arch-pc

### 2.1.3 С помощью функциональной клавиши F7 создайте папку lab05 (рис.2.4) и перейдите в созданный каталог(рис.2.5).

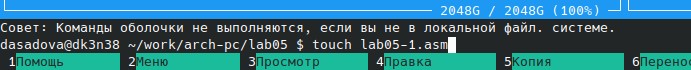


Создаём папку lab05

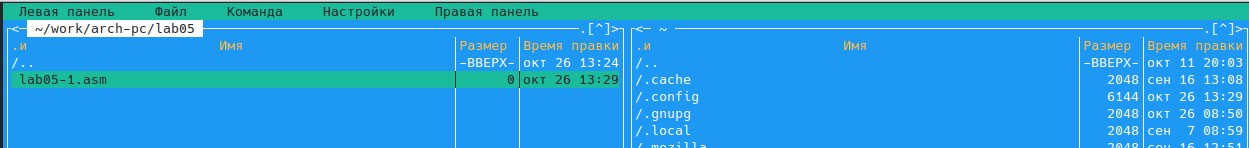


Переходим в папку lab05

### 2.1.4 Пользуясь строкой ввода и командой touch создайте файл lab05-1.asm (рис.2.6), (рис.2.7).

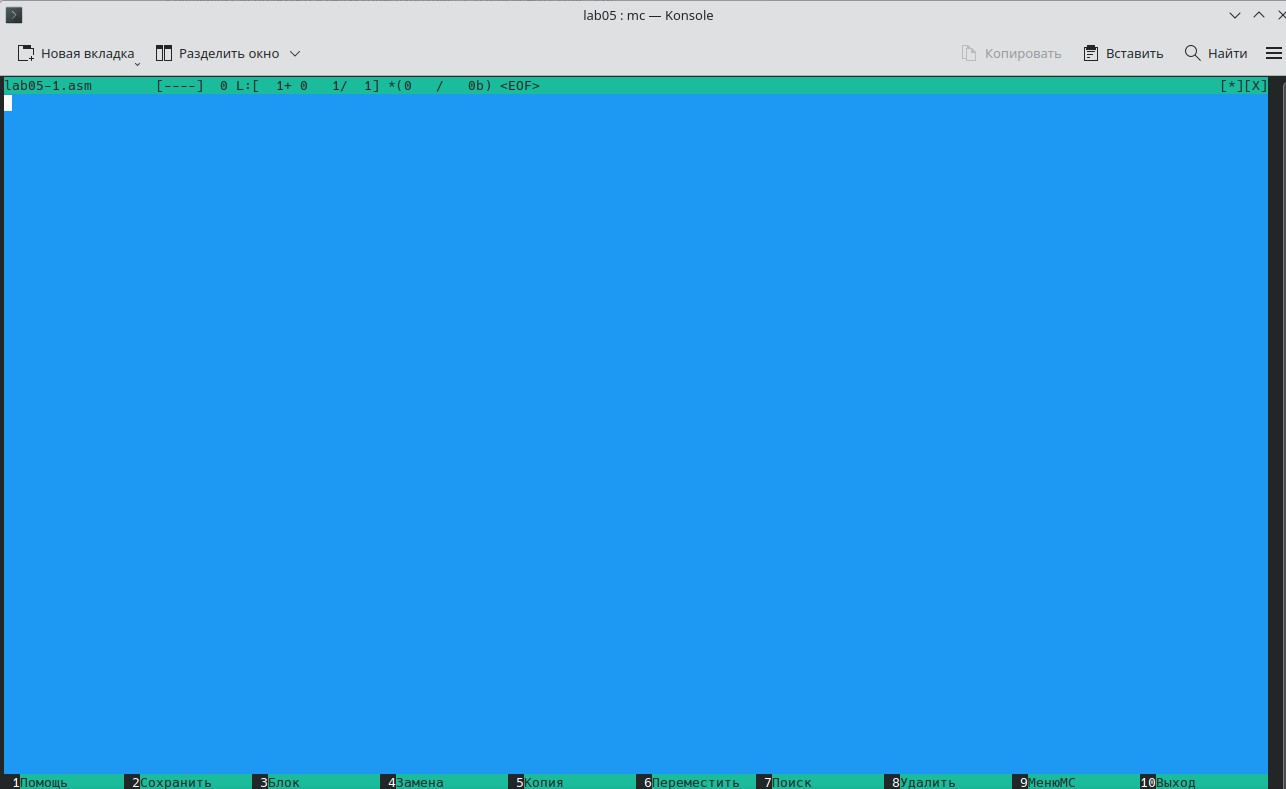


Создаём файл lab05-1.asm



Проверяем наличие файла lab05-1.asm

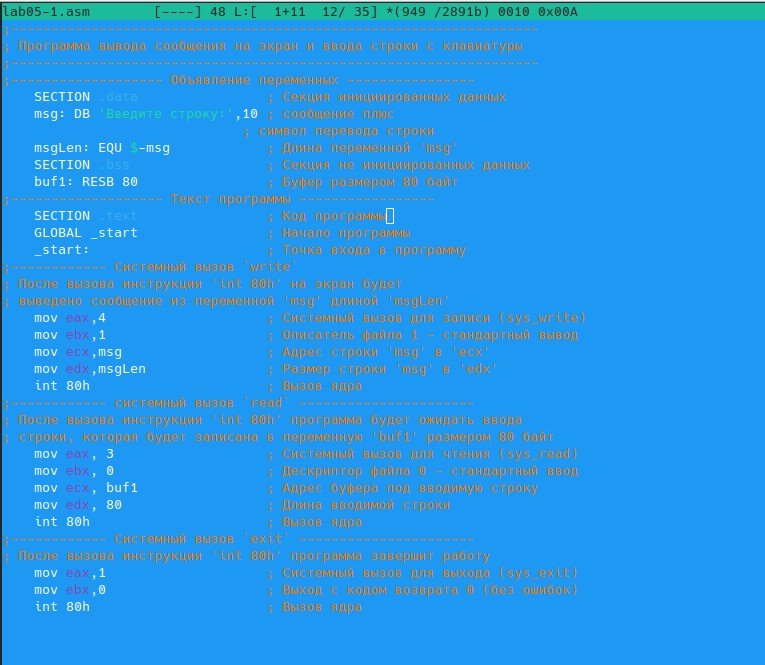
### 2.1.5 С помощью функциональной клавиши F4 откройте файл lab05-1.asm для редактирования во встроенном редакторе. Как правило в качестве встроенного редактора Midnight Commander используется редакторы nano или mcedit (рис.2.8).



Открываем файл lab05-1.asm в редакторе mcedit

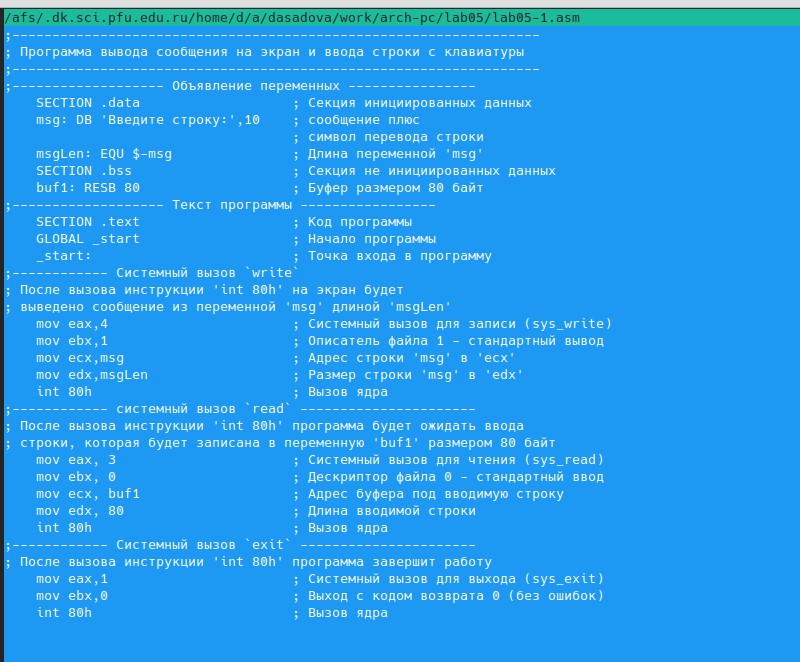
### 2.1.6 Введите текст программы из листинга 5.1 (можно без комментариев), сохраните изменения и закройте файл (рис.2.9).

Листинг 5.1. Программа вывода сообщения на экран и ввода строки с клавиатуры



Вводим текст программы

### 2.1.7 С помощью функциональной клавиши F3 откройте файл lab05-1.asm для просмотра. Убедитесь, что файл содержит текст программы (рис.2.10).

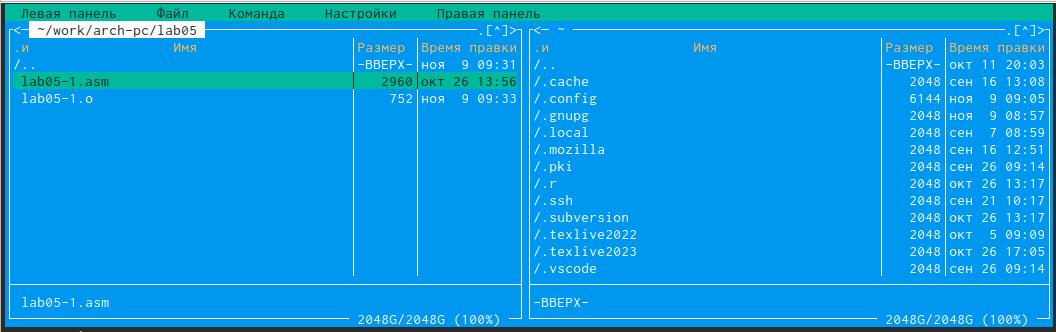


Открываем файл lab05-1.asm для просмотра

### 2.1.8 Оттранслируйте текст программы lab05-1.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.2.11), (рис.2.12), (рис.2.13), (рис.2.14), (рис.2.15), (рис.2.16).

Вводим команду

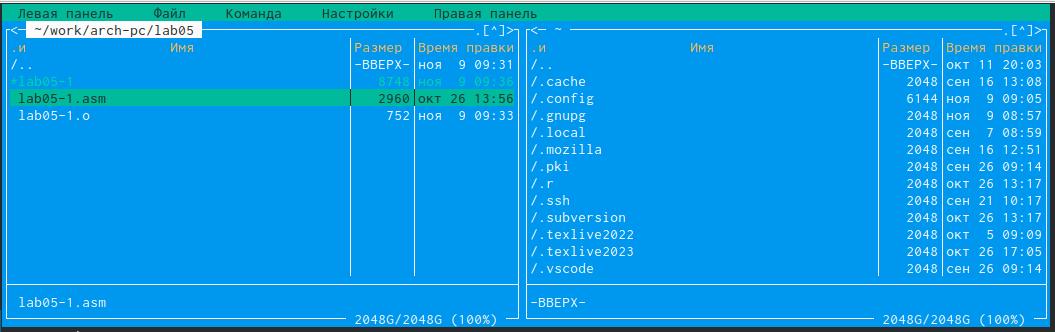
Вводим команду



Проверяем корректность выполнения программы

Вводим команду

Вводим команду



Удостоверемся в правильности выполненной программы

Вводим команду

Вводим команду

Возводим свои имя и фамилию

Возводим свои имя и фамилию

## 2.2 Подключение внешнего файла in\_out.asm

Для упрощения написания программ часто встречающиеся одинаковые участки кода (такие как, например, вывод строки на экран или выход их программы) можно оформить в виде подпрограмм и сохранить в отдельные файлы, а во всех нужных местах поставить вызов нужной подпрограммы. Это позволяет сделать основную программу более удобной для написания и чтения.

NASM позволяет подключать внешние файлы с помощью директивы %include, которая предписывает ассемблеру заменить эту директиву содержимым файла. Подключаемые файлы также написаны на языке ассемблера. Важно отметить, что директива %include в тексте программы должна стоять раньше, чем встречаются вызовы подпрограмм из подключаемого файла. Для вызова подпрограммы из внешнего файла используется инструкция call,которая имеет следующий вид

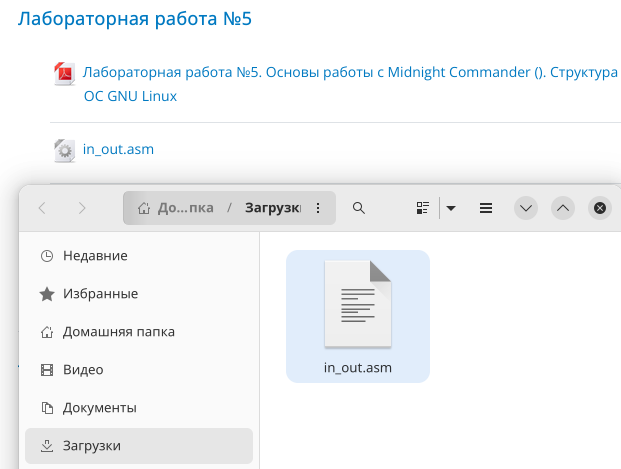
call <function>

где function имя подпрограммы.

Для выполнения лабораторных работ используется файл in\_out.asm1, который содержит следующие подпрограммы [4]:

• slen – вычисление длины строки (используется в подпрограммах печати сообщения для определения количества выводимых байтов);  
• sprint – вывод сообщения на экран, перед вызовом sprint в регистр eax необходимо записать выводимое сообщение (mov eax,<message>);  
• sprintLF – работает аналогично sprint, но при выводе на экран добавляет к сообщению символ перевода строки;  
• sread – ввод сообщения с клавиатуры, перед вызовом sread в регистр eax необходимо записать адрес переменной в которую введенное сообщение буд записано (mov eax,<buffer>) , в регистр ebx – длину вводимой строки (mov ebx,<N>); 1Файл in\_out.asm можно скачать на странице курса в ТУИС.  
• iprint – вывод на экран чисел в формате ASCII, перед вызовом iprint в регистр eax необходимо записать выводимое число (mov eax,<int>);  
• iprintLF – работает аналогично iprint, но при выводе на экран после числа добавляет к символ перевода строки;  
• atoi – функция преобразует ascii-код символа в целое число и записыает результат в регистр eax, перед вызовом atoi в регистр eax необходимо записать число (mov eax,<int>);  
• quit – завершение программы.

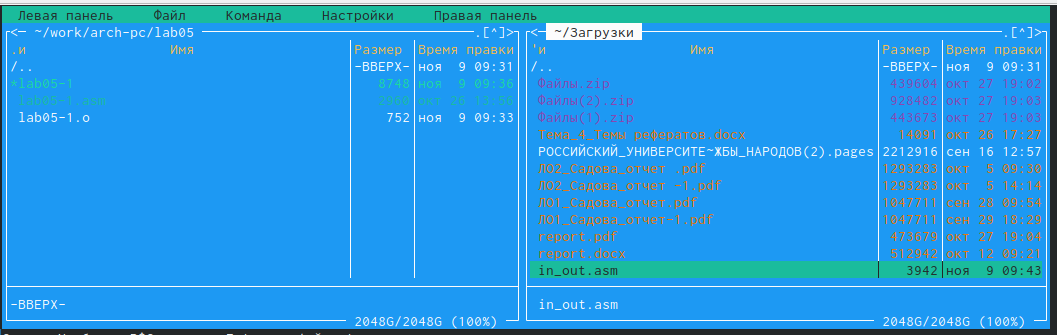
### 2.2.1 Скачайте файл in\_out.asm со страницы курса в ТУИС (рис.2.17).



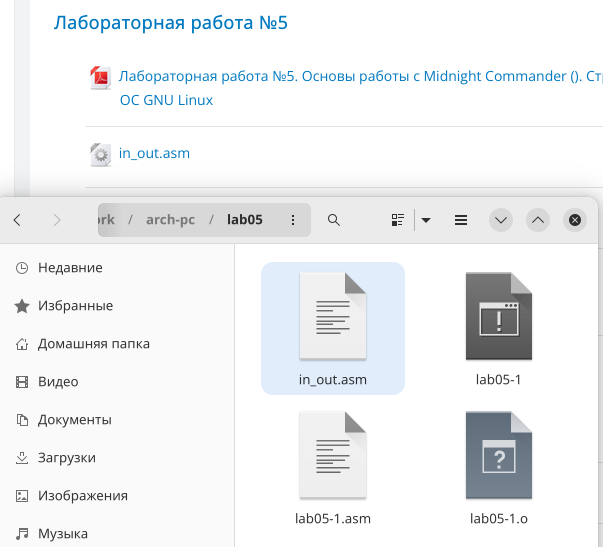
Скачиваем файл in\_out.asm с ТУИС

### 2.2.2 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 lab05-1.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 lab05-1.asm с помощью функциональной клавиши F5 (рис.2.18), (рис.2.19).



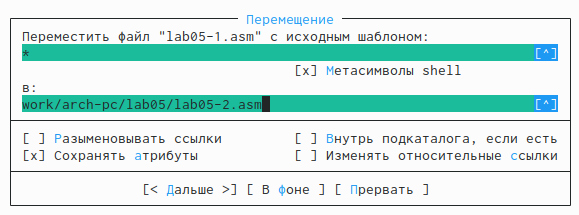
Копируем файл in\_out.asm и вставляем в каталог lab05



Проверяем корректность выполненной программы

Рис. 5.7. Окно Midnight Commander. Копирование файла

### 2.2.3 С помощью функциональной клавиши F6 создайте копию файла lab05-1.asm с именем lab05-2.asm. Выделите файл lab05-1.asm, нажмите клавишу F6 , введите имя файла lab05-2.asm и нажмите клавишу Enter (рис.2.20).

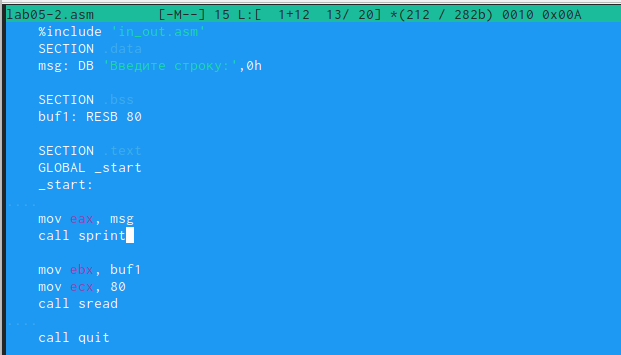


С помощью клавиши F6 перемещаем файл lab05-1.asm в каталог lab05 с новым именем

Клавиша F6 - это клавиша для переноса файла или группы файлов из каталога, отображаемого в первой панели, в каталог, отображаемый на второй панели. Так же, возможно переписывать название файла во второй панели.

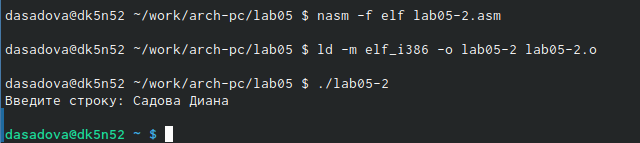
### 2.2.4 Исправьте текст программы в файле lab05-2.asm с использование подпрограмм из внешнего файла in\_out.asm (используйте подпрограммы sprintLF, sread и quit) в соответствии с листингом 5.2. Создайте исполняемый файл и проверьте его работу (рис.2.21).

Листинг 5.2. Программа вывода сообщения на экран и ввода строки с клавиатуры c использованием файла in\_out.asm



Вводим код программы

### 2.2.5 В файле lab5-2.asm замените подпрограмму sprintLF на sprint. Создайте исполняемый файл и проверьте его работу. В чем разница? (рис.2.22)



Редактируем и выводим работу программы в консоль

Ответ на вопрос: Мы записываем наши данные не на следующей строке, а после двоеточия

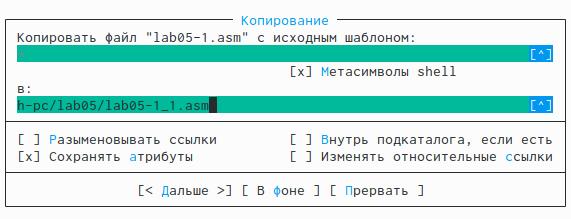
# 3 Теоретическое введение

# 4 Выполнение лабораторной работы

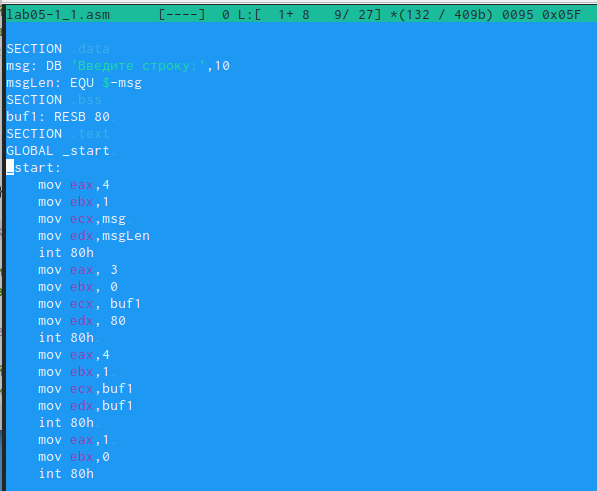
## 4.1 Задание для самостоятельной работы

### 4.1.1 Создайте копию файла lab5-1.asm. Внесите изменения в программу (без использования внешнего файла in\_out.asm), так чтобы она работала по следующему алгоритму: (рис.2.23), (рис.2.24)

• вывести приглашение типа “Введите строку:”;  
• ввести строку с клавиатуры;  
• вывести введённую строку на экран.

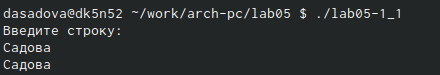


Создаем копию файла lab5-1.asm с именем lab5-1\_1.asm



Вводим код программы

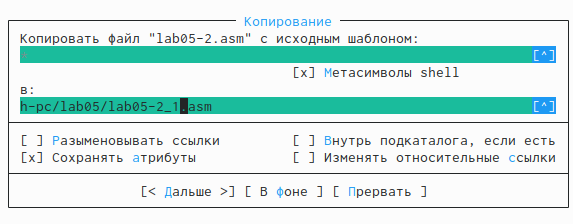
### 4.1.2 Получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите свою фамилию (рис.2.25).



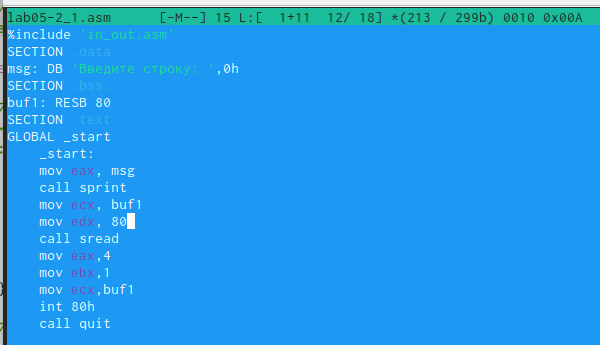
Проверяем корректность выполненной программы

### 4.1.3 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: (рис.2.26), (рис.2.27)

• вывести приглашение типа “Введите строку:”;  
• ввести строку с клавиатуры;  
• вывести введённую строку на экран.

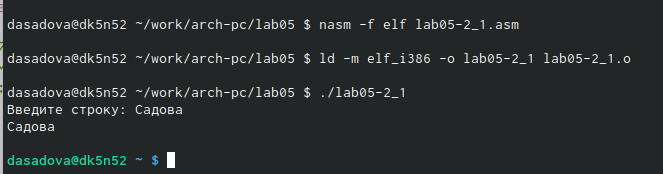


Создаем копию файла lab5-2.asm с именем lab5-2\_1.asm



Вводим код програмы

### 4.1.4 Создайте исполняемый файл и проверьте его работу (рис.2.28).



Проверяем коректность выполненой программы

# 5 Выводы

Приобретенны практические навыкы работы в Midnight Commander. Освоянны инструкции языка ассемблера mov и int.

# Список литературы