Отчёт по лабораторной работе №4

дисциплина: Архитектура вычислительных систем

Мосолов Александр Денисович

Содержание

# 1 Цель работы

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

# 2 Выполнение лабораторной работы

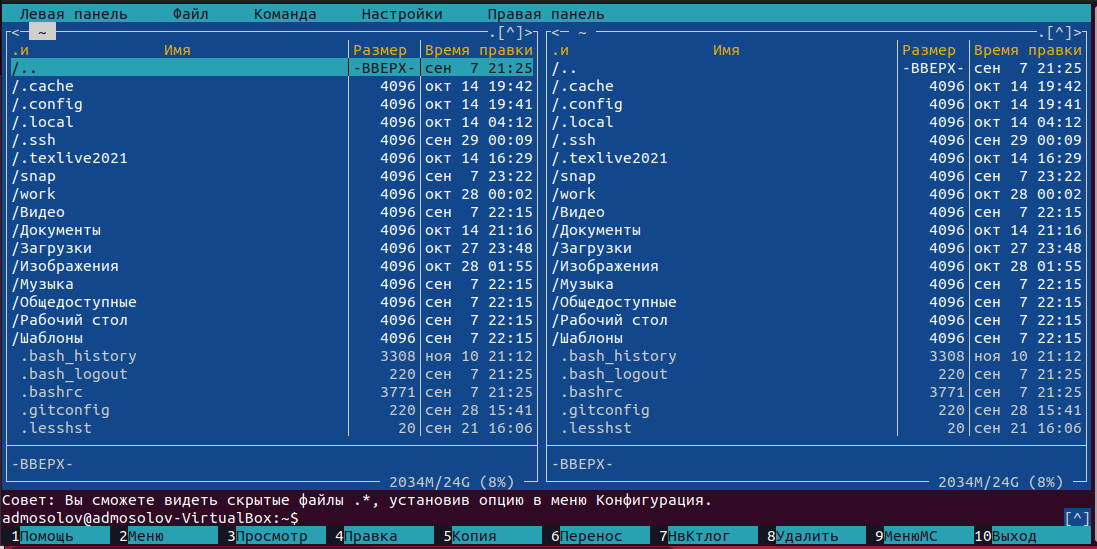
## 2.1 Программа вывода сообщения на экран и ввода строки с клавиатуры

С помощью команды *mc* открываем *Midnight Commander*

Используем команду mc

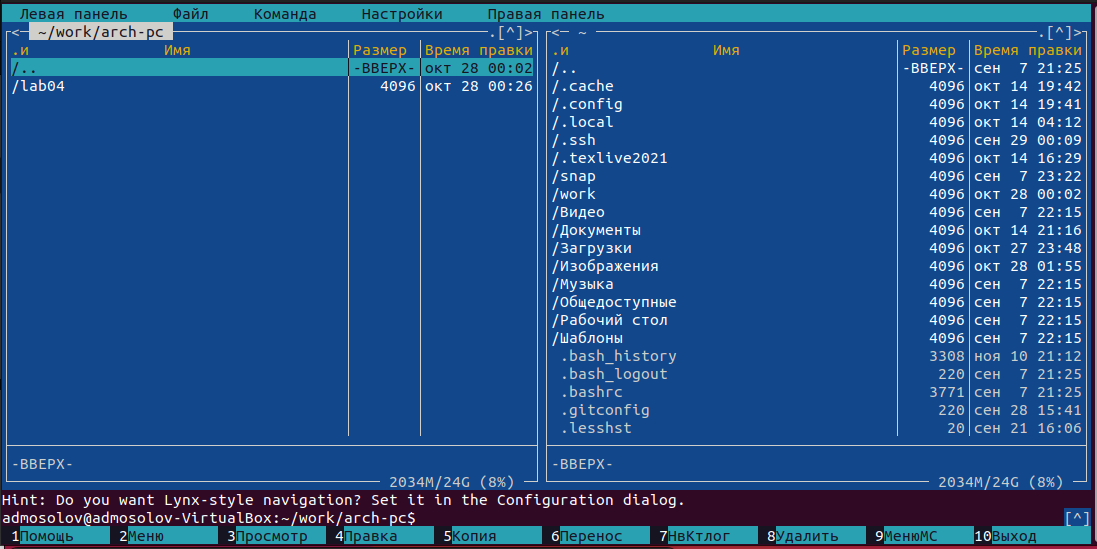
Используем команду mc

С помощью клавиш *вверх*, *вниз* можно перемещаться по меню, переходить в каталоги с помощью *Enter*.



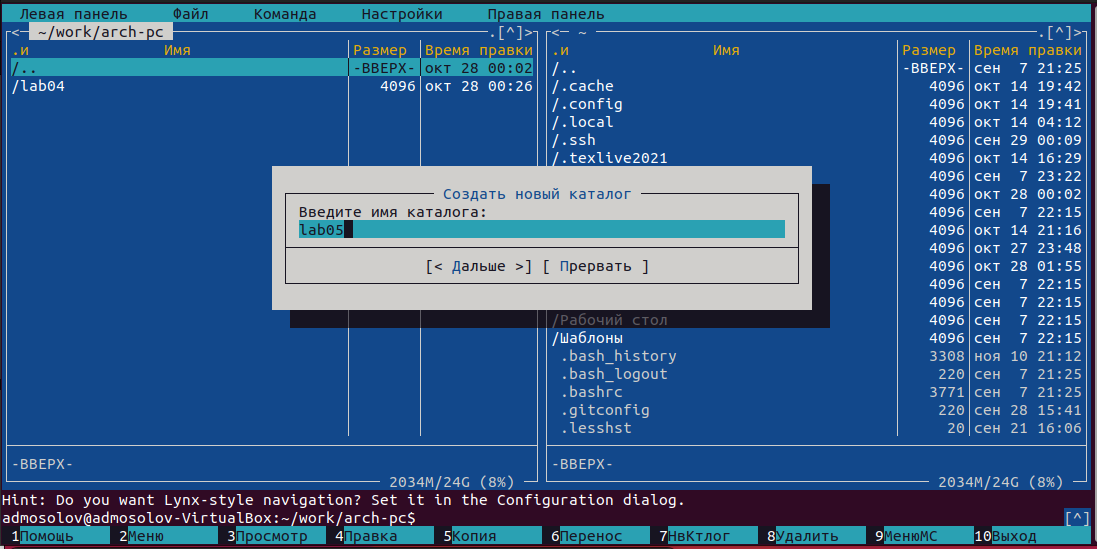
Структура Midnight Commander

Переходим в каталог *~/work/arch-pc*.



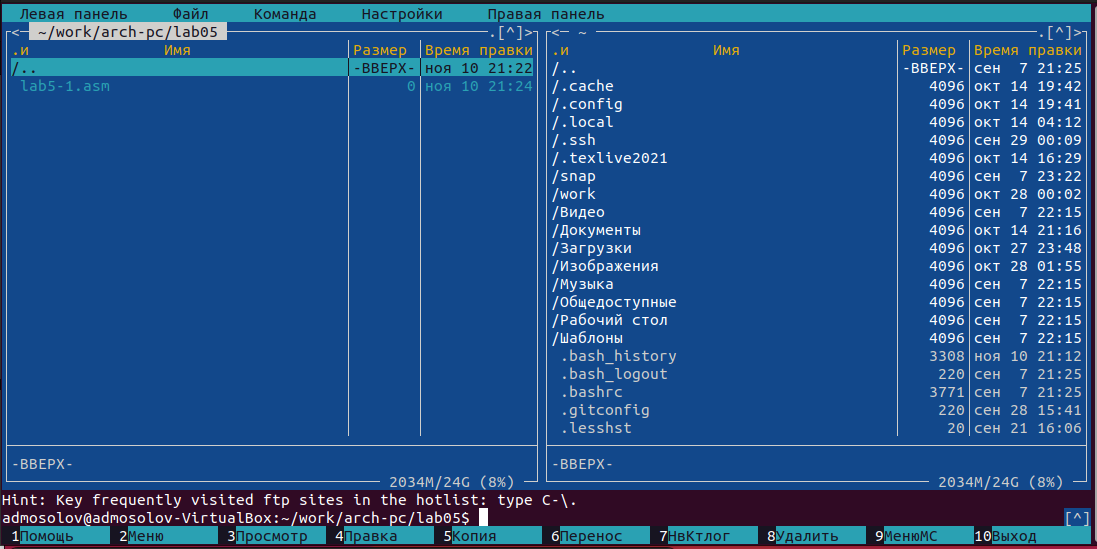
Каталог arch-pc в Midnight Commander

С помощью функциональной клавиши *F7* создаём папку *lab05* и переходим в созданный каталог.



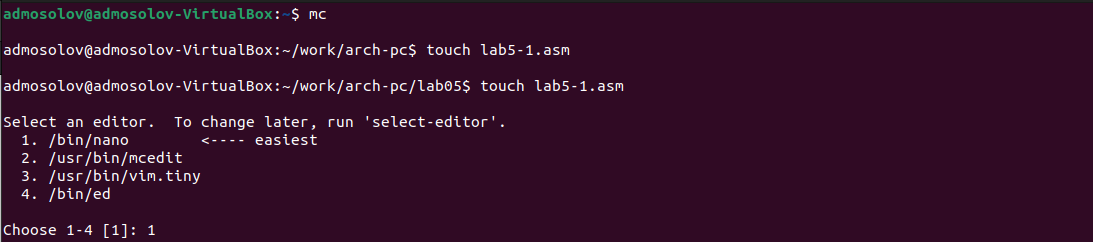
Создаём папку lab05

Пользуясь строкой ввода и командой touch создаём файл *lab5-1.asm*, убеждаемся в *Midnight Commander*, что файл был создан.



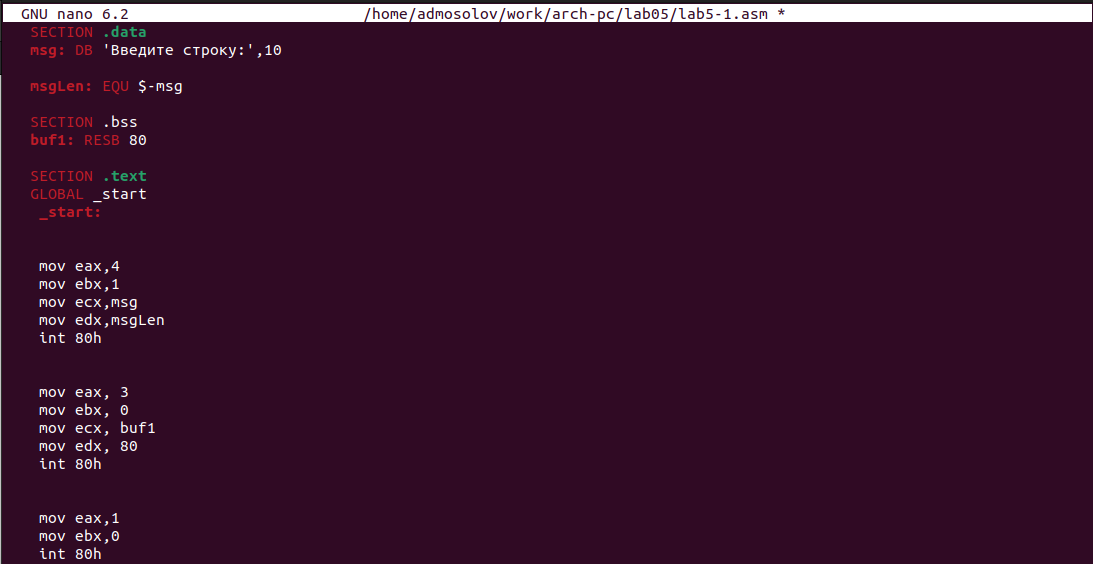
Создаём файл lab5-1.asm

С помощью функциональной клавиши *F4* открываем файл *lab5-1.asm* для редактирования во встроенном редакторе. Выбираем в качестве основного редактора *nano* с помощью клавиши *1* в терминале.



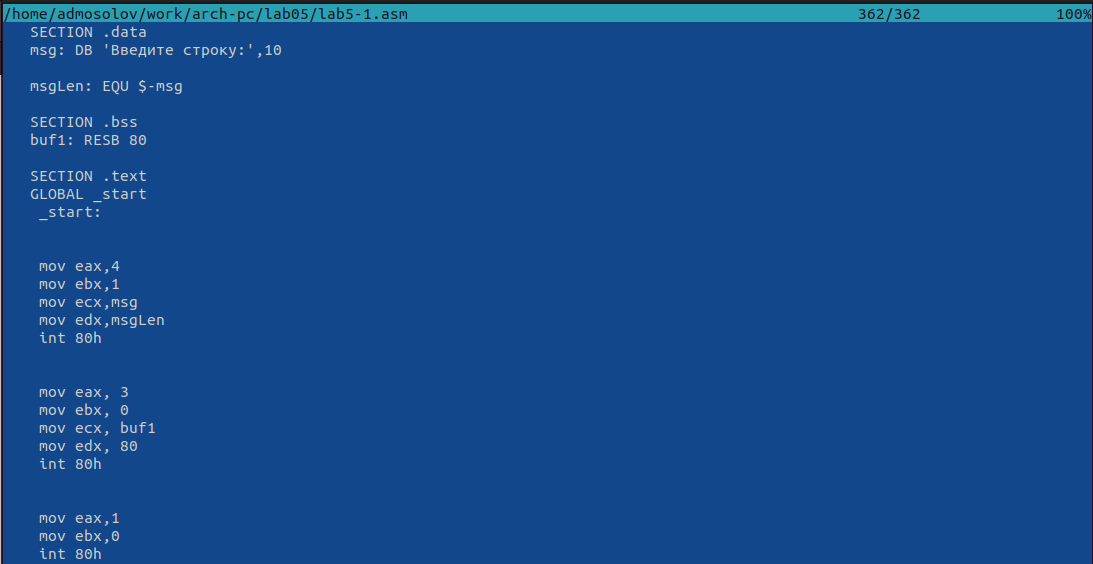
Выбираем редактор

Вводим текст программы из листинга *5.1*, сохраняем (*Ctrl+o*) изменения и закрываем файл (*Ctrl+x*).



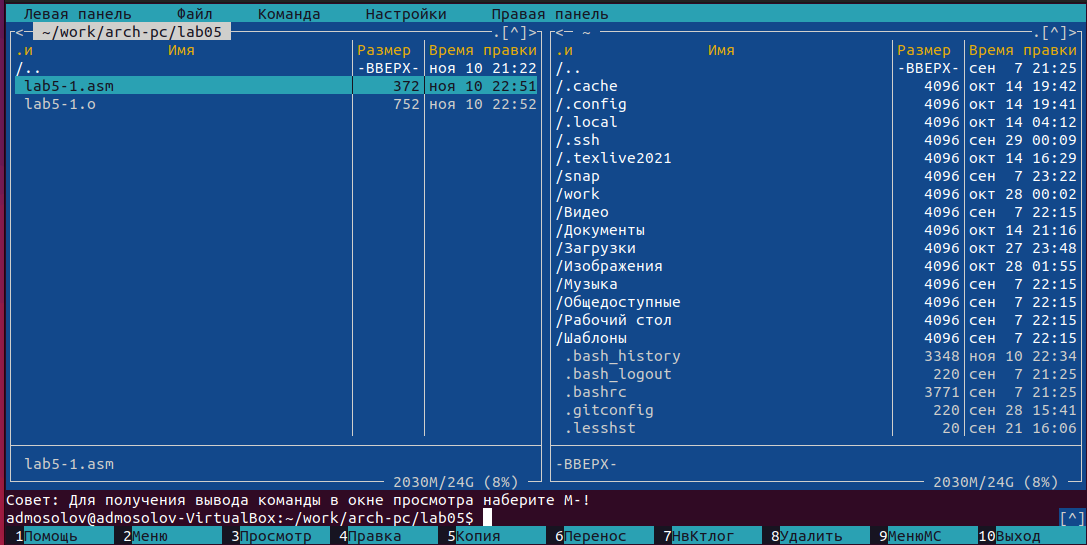
Текст программы из листинга 5.1

С помощью функциональной клавиши *F3* открываем файл *lab5-1.asm* для просмотра. Убеждаемся, что файл содержит текст программы.

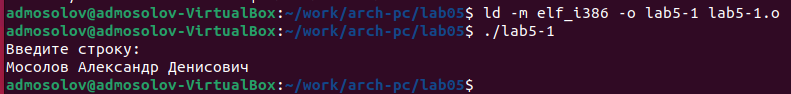


Использования функциональной клавиши F3

Транслируем текст программы *lab5-1.asm* в объектный файл для этого используем команду *nasm -f elf lab5-1.asm*. Выполняем компоновку объектного файла с помощью команды *ld -m elf\_i386 -o lab5-1 lab5-1.o* и запускаем получившийся исполняемый файл, используя команду *./lab5-1*. Программа выводит строку *‘Введите строку:’* и ожидает ввода с клавиатуры. На запрос вводим ФИО.



Объектный файл



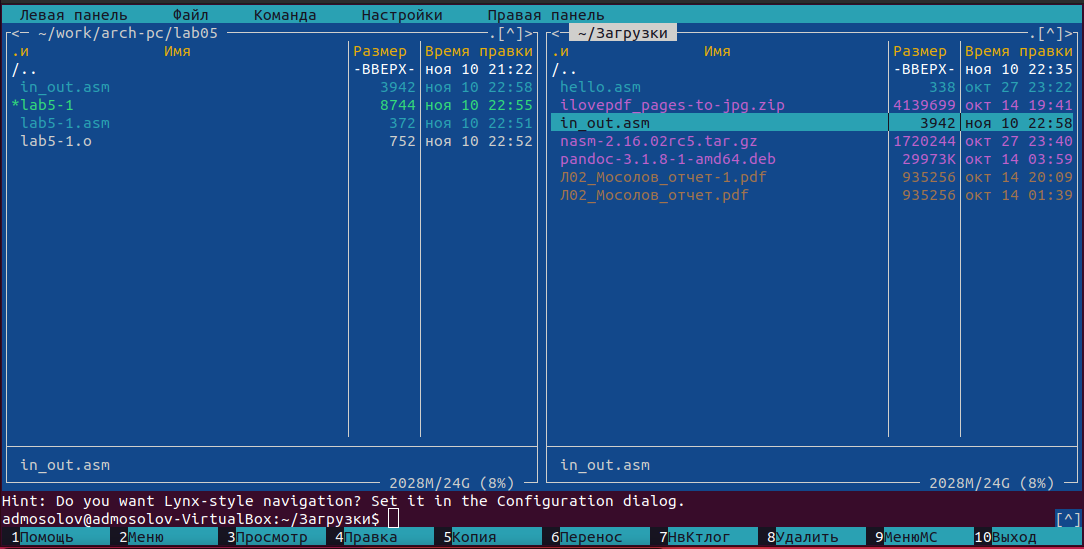
Используем команды для запуска программы

## 2.2 Подключение внешнего файла in\_out.asm

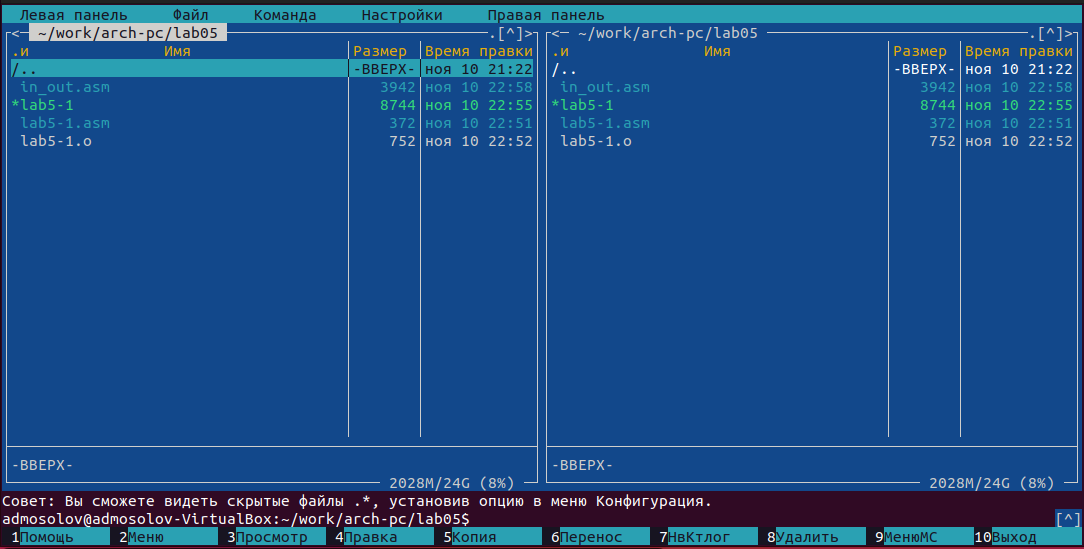
Для упрощения написания программ часто встречающиеся одинаковые участки кода (такие как, например, вывод строки на экран или выход их программы) можно оформить в виде подпрограмм и сохранить в отдельные файлы, а во всех нужных местах поставить вызов нужной подпрограммы. Это позволяет сделать основную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

Скачиваем файл in\_out.asm со страницы курса в ТУИС. Переносим его в папку с программой.

В одной из панелей *mc* открываем каталог с файлом *lab5-1.asm*. В другой панели каталог со скаченным файлом *in\_out.asm* (для перемещения между панелями используем *Tab* ). Копируем файл *in\_out.asm* в каталог с файлом *lab5-1.asm* с помощью функциональной клавиши *F5*.

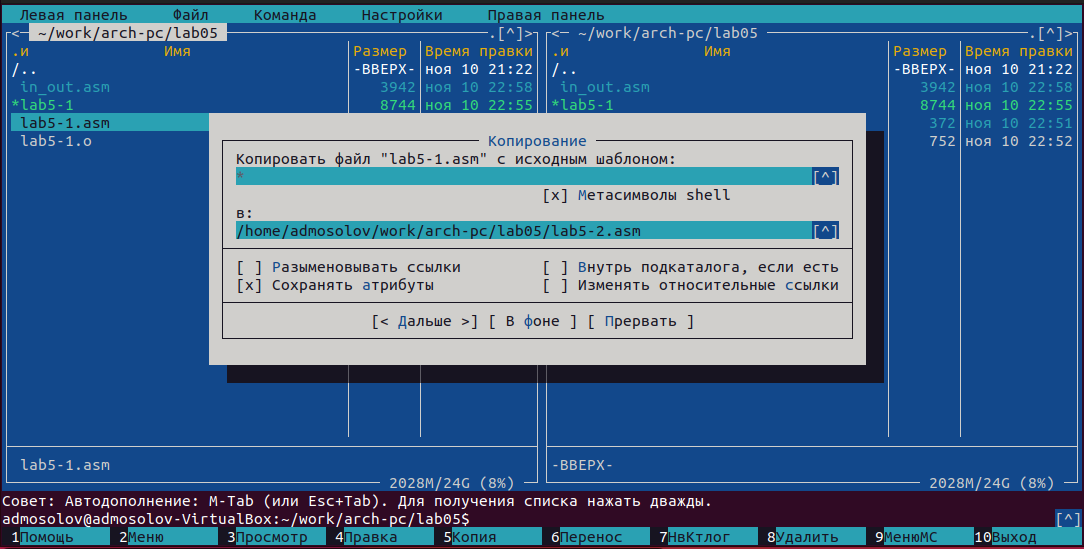


Перемещаем скачанный файл



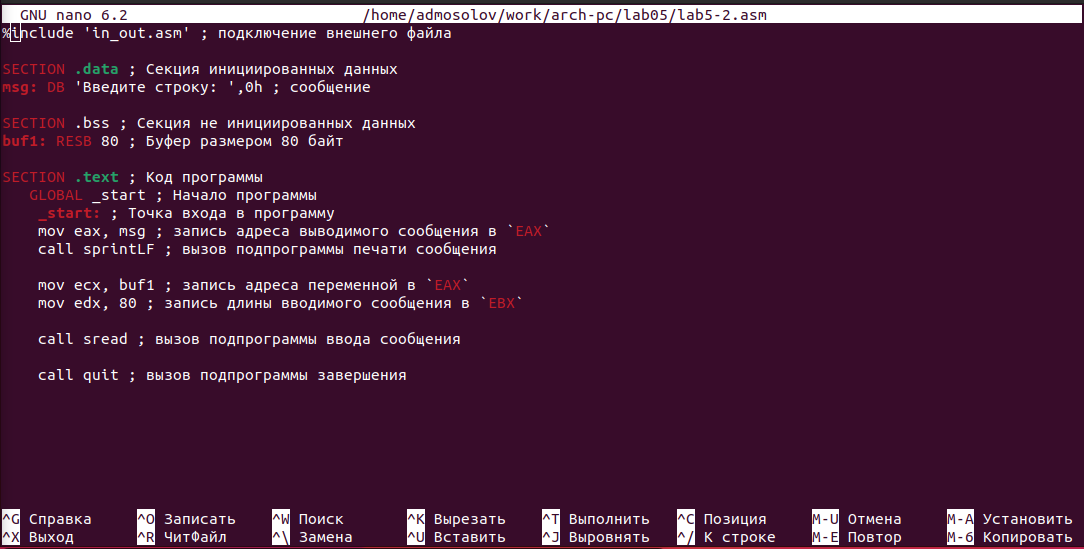
Каталог lab05

С помощью функциональной клавиши *F5* создаём копию файла *lab5-1.asm* с именем *lab5-2.asm*. Выделяем файл *lab5-1.asm*, нажимаем клавишу *F6* , вводим имя файла *lab5-2.asm* и нажимаем клавишу *Enter*.



Создаем копию файла с помощью F5

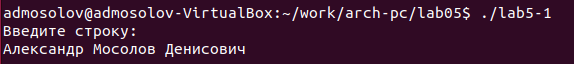
Исправляем текст программы в файле *lab5-2.asm* с использованием подпрограмм из внешнего файла *in\_out.asm* (используем подпрограммы *sprintLF*, *sread* и *quit*) в соответствии с листингом *5.2*.



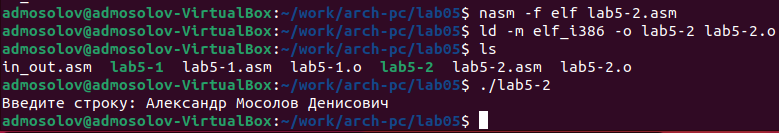
Текст программы lab5-2

В файле *lab5-2.asm* замените подпрограмму *sprintLF* на *sprint*. Создайте исполняемый файл и проверьте его работу. В чем разница?

Разница заключается в том, что при использовании подпрограммы *sprintLF* при печати строк программа автоматически переводила курсор на следующую строку, а при *sprint* курсор смещается вправо и не переходит на следующую строку.



Используем подпрограмму sprintLF



Используем подпрограмму sprint

## 2.3 Задание для самостоятельной работы

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.

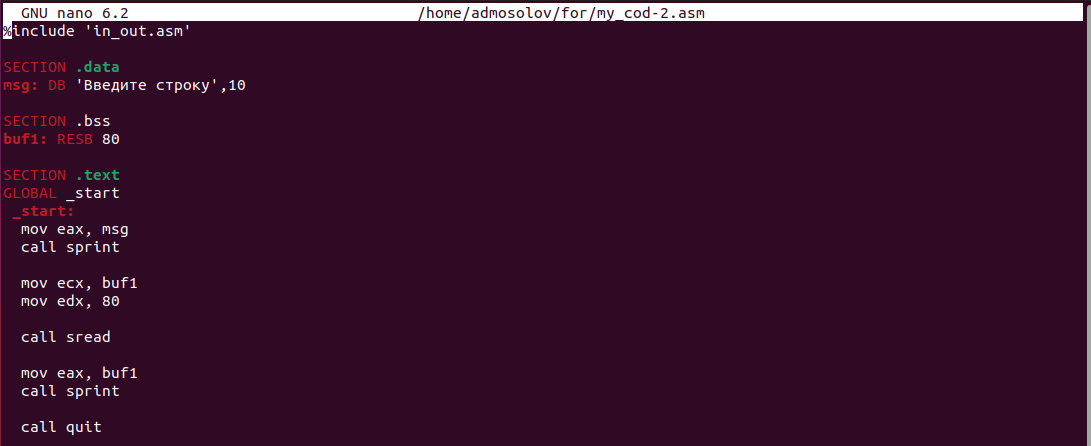
Создадим в домашнем каталоге папку *for* для выполнения этого задания. Пернесём туда файл *in\_out.asm*. Назовём файл с программой *my\_cod.asm*. Напишем текст программы.



Текст программы my\_cod.asm

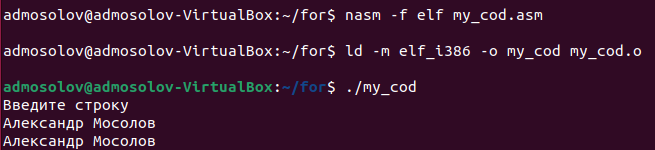
Создайте копию файла *lab5-1.asm*. Внесите изменения в программу с использованием внешнего файла *in\_out.asm*, так чтобы она работала по следующему алгоритму: \* вывести приглашение типа “Введите строку:”; \* ввести строку с клавиатуры; \* вывести введённую строку на экран.

Назовём файл с программой *my\_cod-2.asm*. Напишем текст программы.



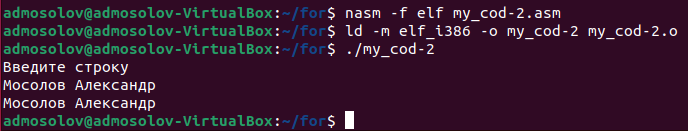
Текст программы my\_cod-2.asm

Транслируем текст программы (*my\_cod.asm*). Для трансляции используем: *nasm -f elf my\_cod.asm*. Выполняем компоновку с помощью команды *ld -m elf\_i386 -o my\_cod my\_cod.o* и запускаем получившийся исполняемый файл, используя команду *./my\_cod*.



Запускаем программу my\_cod

Аналогичные команды используем для запуска программы *my\_cod-2*.



Запускаем программу my\_cod-2

Переименновываем все файлы с именем *my\_cod* на *lab5-1*, а с именем *my\_cod-2* на *lab5-2* и переносим все файлы из каталога *for* в каталог *~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab05*. Удостоверимся в том, что правильно перенесли файлы с помощью команды *ls*.

Просмотр каталога lab05

Просмотр каталога lab05

Загружаем файлы на *github*.

Загружаем файлы на github

Загружаем файлы на github

# 3 Выводы

В ходе выполнения лабораторной работы были приобретены практические навыки работы в *Midnight Commander*, помимо этого были освоены инструкции языка ассемблера *mov* и *int*.