Отчёта по лабораторной работе 5

Основы работы с Midnight Commander (mc). Структура программы на языке ассемблера NASM

Гадаборшев Заур Закреевич НПИбд-01-23

Содержание

# 1 Цель работы

Целью работы является приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера mov и int.

# 2 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.

Программа на языке ассемблера NASM, как правило, состоит из трёх секций: секция кода программы (SECTION .text), секция инициированных (известных во время компиляции) данных (SECTION .data) и секция неинициализированных данных (тех, под которые во время компиляции только отводится память, а значение присваивается в ходе выполнения программы) (SECTION .bss).

Инструкция языка ассемблера mov предназначена для дублирования данных источника в приёмнике. В общем виде эта инструкция записывается в виде mov dst,src Здесь операнд dst — приёмник, а src — источник

Инструкция языка ассемблера intпредназначена для вызова прерывания с указанным номером. В общем виде она записывается в виде int n Здесь n — номер прерывания, принадлежащий диапазону 0–255

# 3 Выполнение лабораторной работы

## 3.1 Знакомство с Midnight Commander

Открыл Midnight Commander, с помощью клавишь со стрелками и Enter перешел в каталог ~/work/arch-pc. Далее нажал F7 и создал каталог lab05

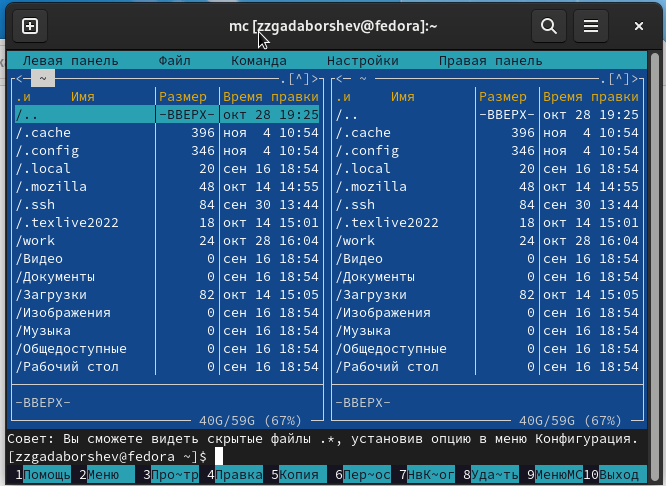


Figure 1: Запуск каталога Midnight Commander

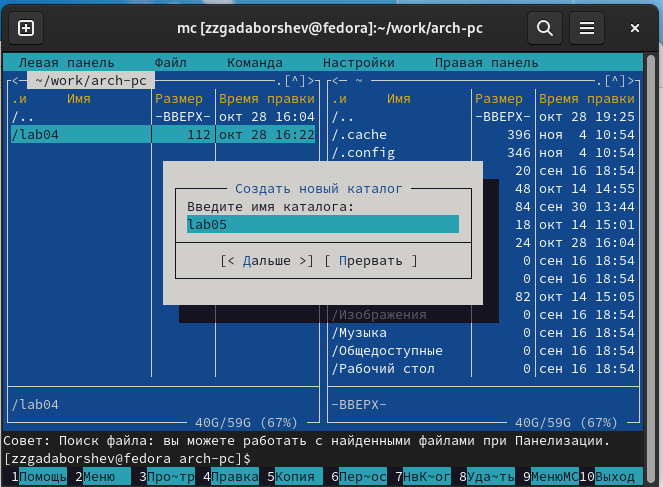


Figure 2: Создание каталога

При помощи touch создал файл lab05-1.asm

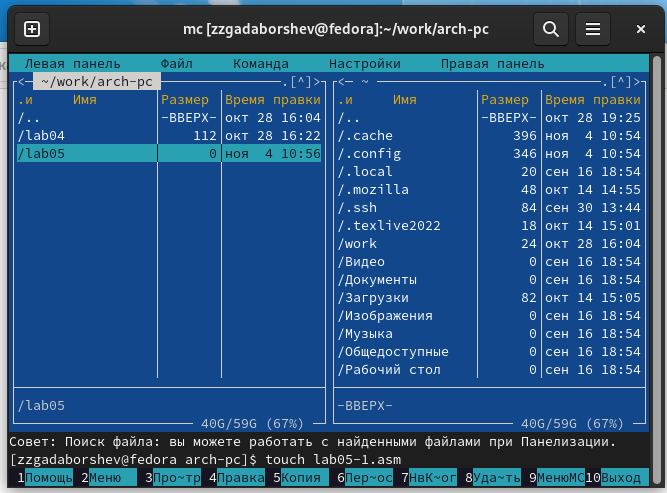


Figure 3: Создание файла lab05-1.asm

Открыл файл на редактирование клавишей F4, выбрал редактор mceditor, написал код программы из задания.

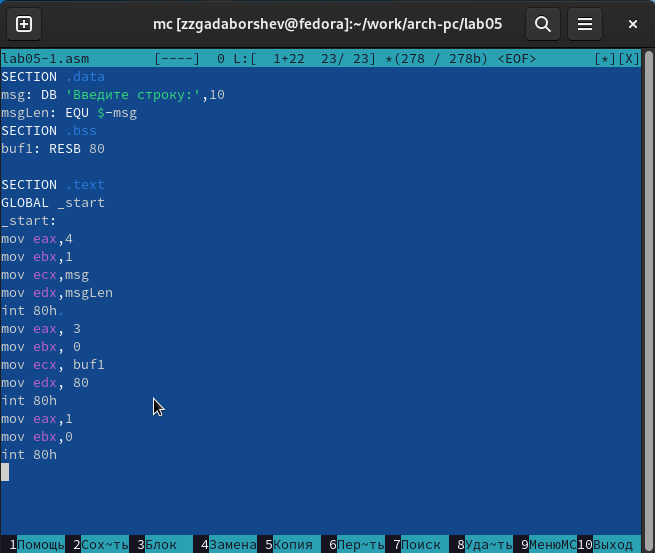


Figure 4: Программа в файле lab05-1.asm

Открыл файл на просмотр клавишей F3 и убелился, что он содержит набранный код.

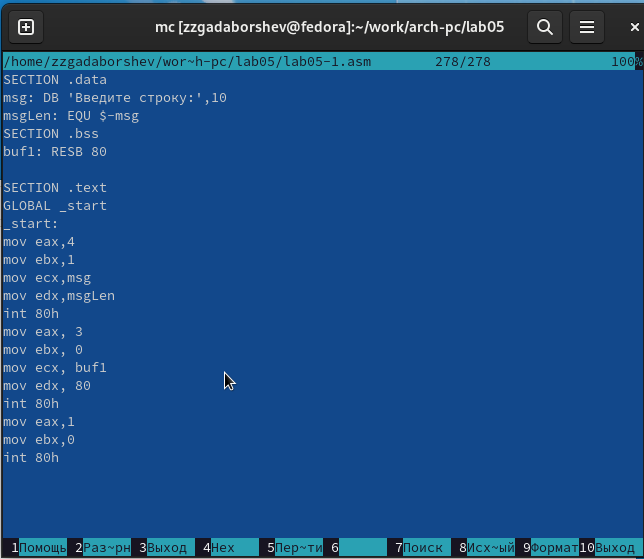


Figure 5: Просмотр файла lab05-1.asm

Транслировал файл программы в объектный файл, выполнил компановку объектного файла, получил исполняемый файл программы и провреил ее работу.

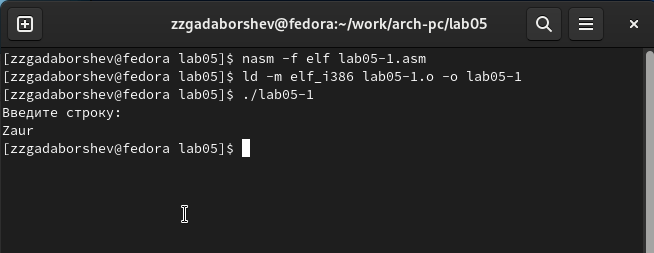


Figure 6: Запуск программы lab05-1.asm

## 3.2 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 выполнения лабораторных работ используется файл in\_out.asm, который содержит следующие подпрограммы:

* 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,);
* 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,) , в регистр ebx – длину вводимой строки (mov ebx,);
* iprint – вывод на экран чисел в формате ASCII, перед вызовом iprint в регистр eax необходимо записать выводимое число (mov eax,);
* 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,);
* quit – завершение программы.

Скачал файл in\_out.asm и разместил его в рабочем каталоге. Для копирования используется клавиша F5. Для перемещения используется клавиша F6.

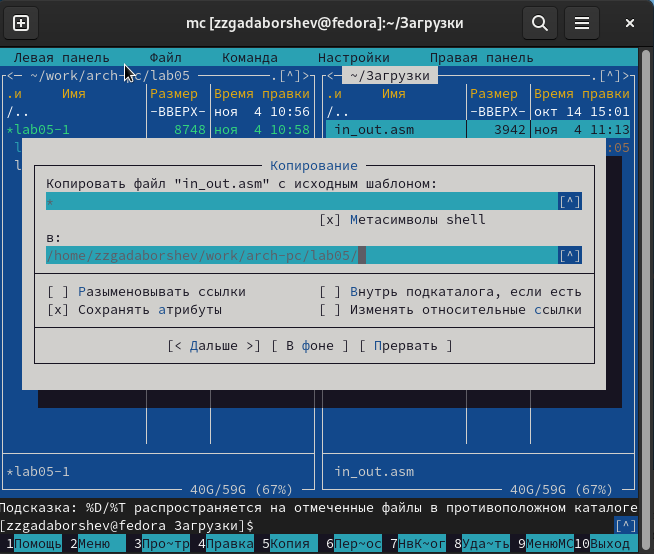


Figure 7: Копирование файла in\_out.asm

Скопировал lab05-1.asm в lab05-2.asm.

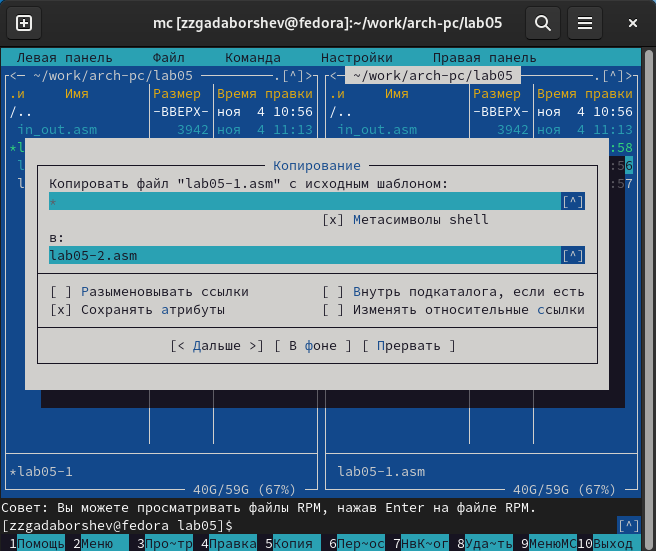


Figure 8: Копирование файла lab05-1.asm

Написал код программы lab05-2.asm с использованием подпрограмм из внешнего файла in\_out.asm . Скомпилировал программу и провреил запуск.

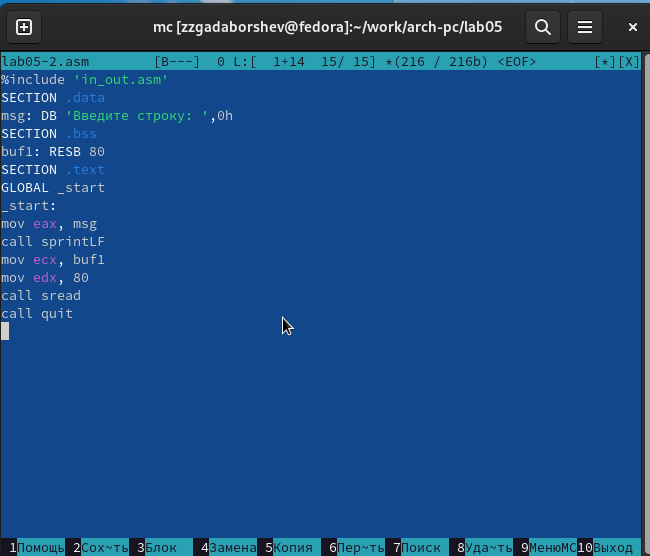


Figure 9: Программа в файле lab05-2.asm

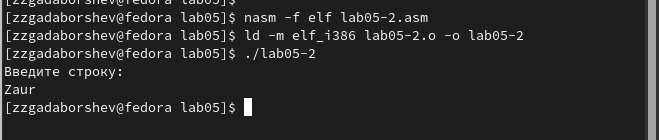


Figure 10: Запуск программы lab05-2.asm

В файле lab5-2.asm заменил подпрограмму sprintLF на sprint. Заново собрал исполняеый файл. Теперь после вывода строки она не завершается символом перехода на новую строку.

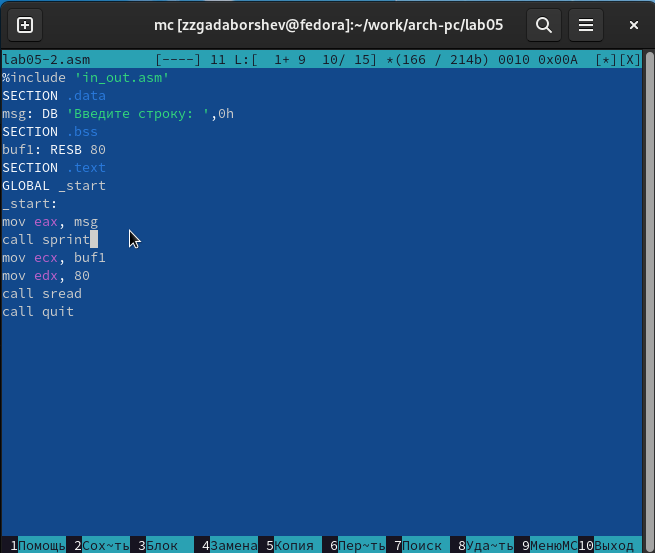


Figure 11: Программа в файле lab05-2.asm

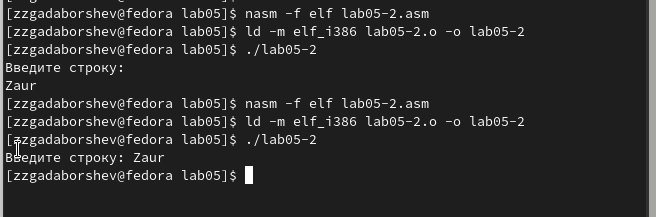


Figure 12: Запуск программы lab05-2.asm

## 3.3 Задание для самостоятельной работы

Скопировал программу lab05-1.asm и измении код, так чтобы она работала по следующему алгоритму:

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

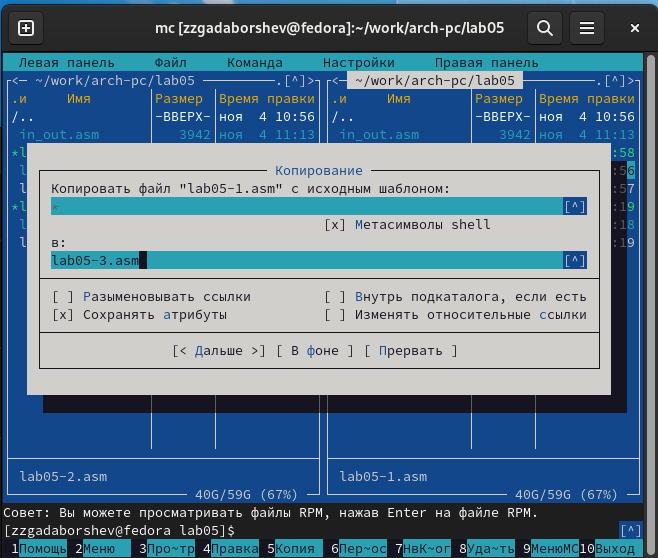


Figure 13: Копирование файла lab05-1.asm

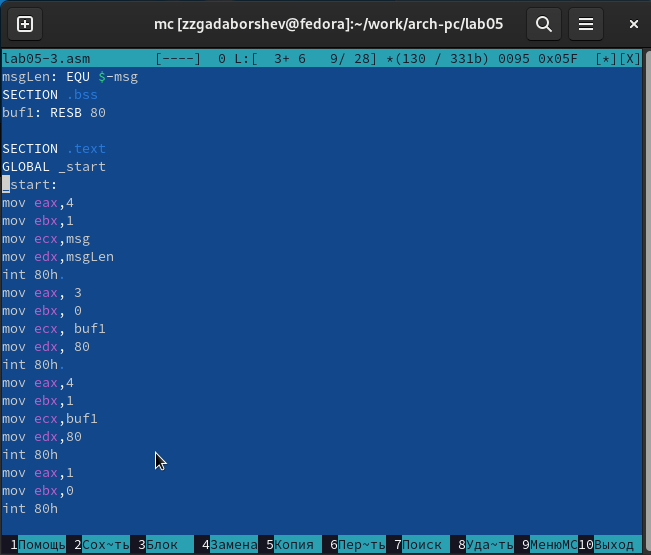


Figure 14: Программа в файле lab05-3.asm

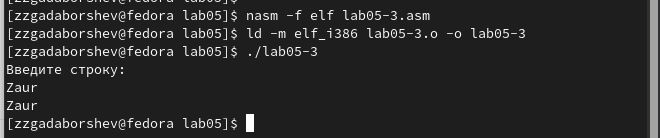


Figure 15: Запуск программы lab05-3.asm

Аналогично скопировал программу lab05-2.asm и изменил код, но теперь использовал подпрограммы из файла in\_out.asm.

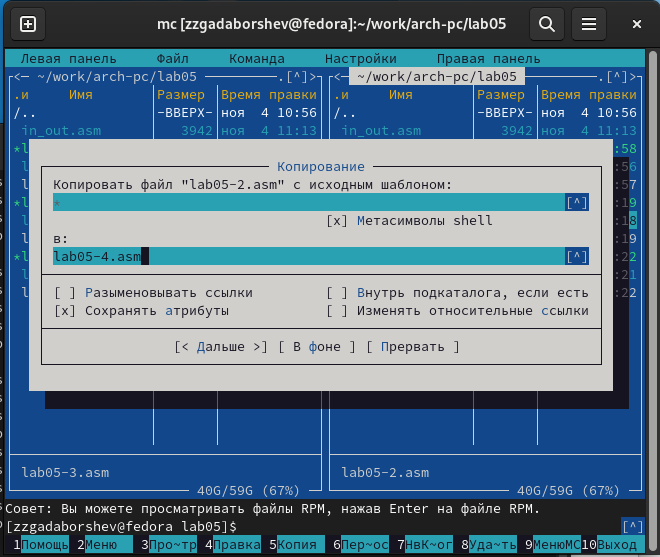


Figure 16: Копирование файла lab05-2.asm

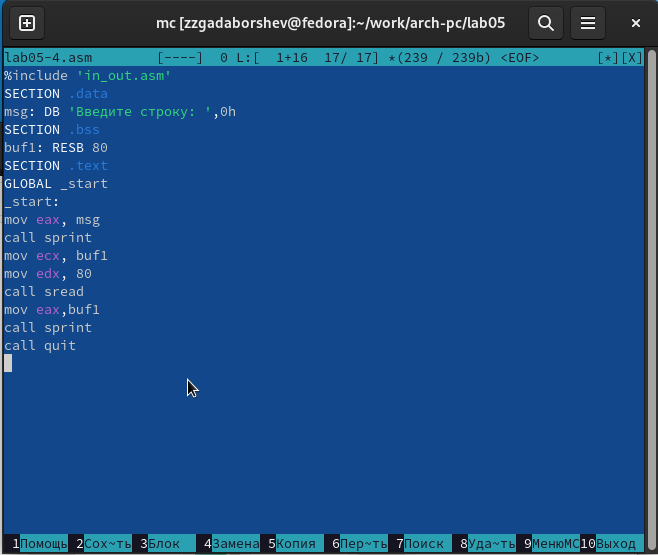


Figure 17: Программа в файле lab05-4.asm

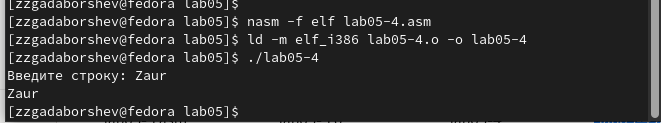


Figure 18: Запуск программы lab05-4.asm

# 4 Выводы

Научились писать базовые ассемблерные программы. Освоили ассемблерные инструкции mov и int.