Отчёт по лабораторной работе 5

Дисциплина: архитектура компьютера

Плугин Никита

Содержание

# 1 Цель работы

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

# 2 Задания

1. Изучить основы работы с Midnight Commander
2. Изучить инструкции ассемблера
3. Выполнить самостоятельное задание по изменению программы

# 3 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).

# 4 Выполнение лабораторной работы

## 4.1 Работа в Midnight Commander

1. Открыл Midnight Commander. Перешел в каталог ~/work/arch-pc. Создал каталог lab05 (рис. [[1](#fig:001)])

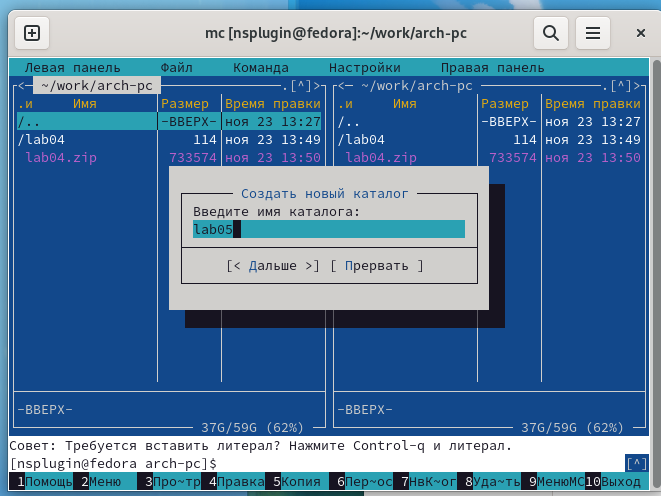


Figure 1: Создание каталога

1. Создал файл lab05-1.asm (рис. [[2](#fig:002)])

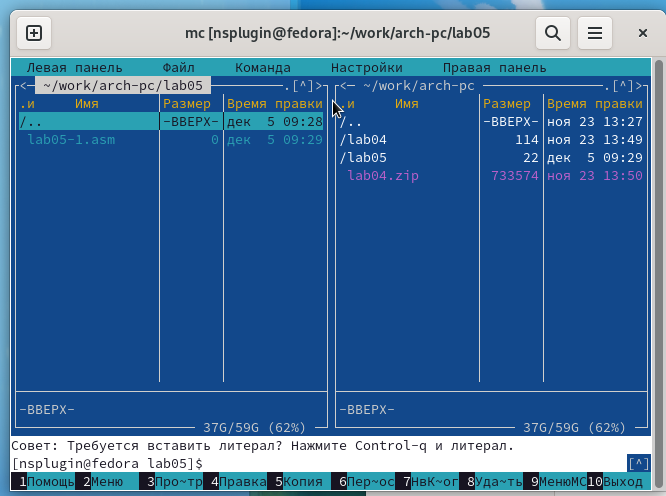


Figure 2: Создание файла lab05-1.asm

1. Открыл файл на редактирование. Написал код. (рис. [[3](#fig:003)])

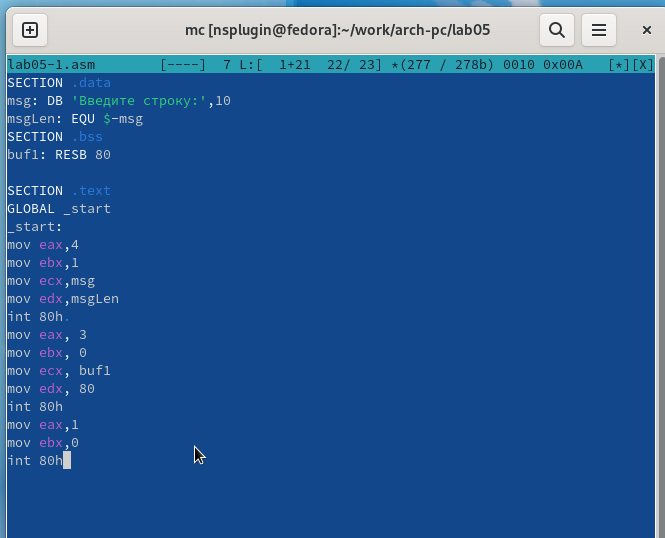


Figure 3: Программа lab05-1.asm

1. Открыл файл для просмотра и убедился, что он содержит написанный код. (рис. [[4](#fig:004)])

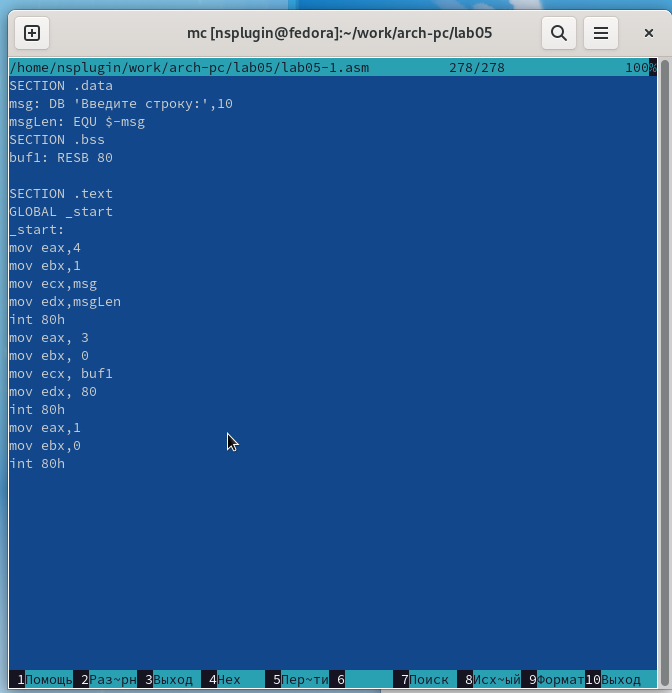


Figure 4: Просмотр файла lab05-1.asm

1. Получил исполняемый файл программы и провреил его работу.(рис. [[5](#fig:005)])

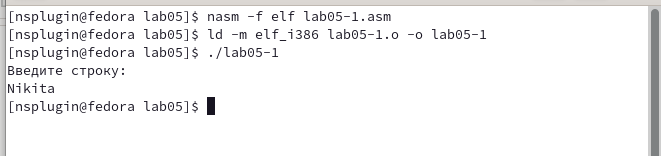


Figure 5: Проверка программы lab05-1.asm

## 4.2 Подключение внешнего файла in\_out.asm

1. Скачал файл in\_out.asm. Добавил файл in\_out.asm в рабочий каталог. Скопировал lab05-1.asm в lab05-2.asm. (рис. [[6](#fig:006)])

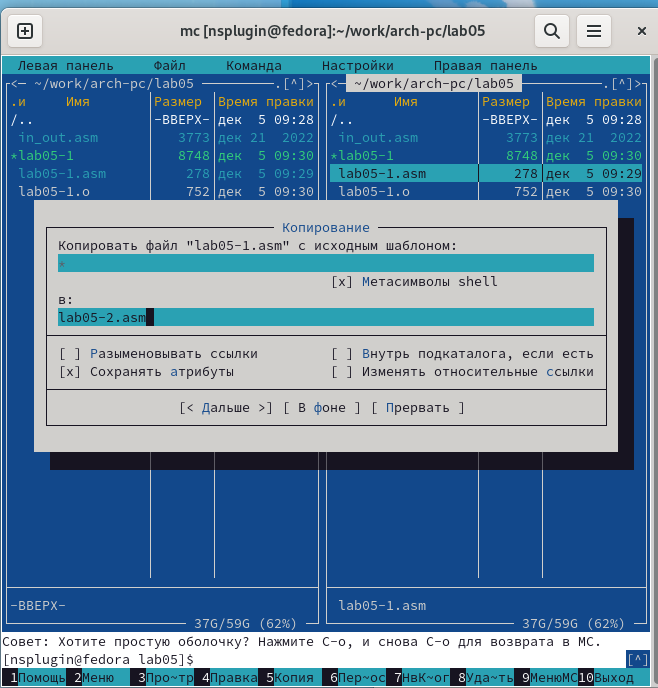


Figure 6: Копирование файла

1. Написал код программы lab05-2.asm. (рис. [[7](#fig:007)]) Скомпилировал программу и провреил запуск. (рис. [[8](#fig:008)])

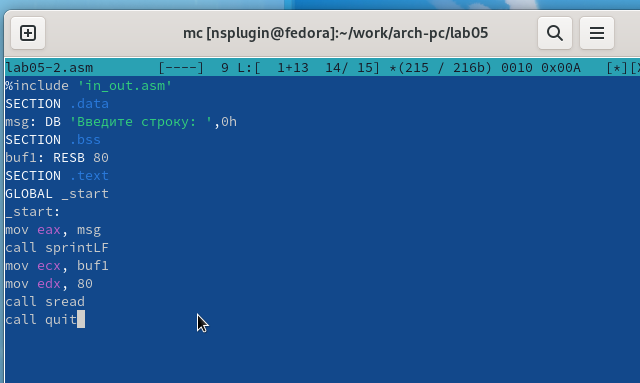


Figure 7: Программа lab05-2.asm

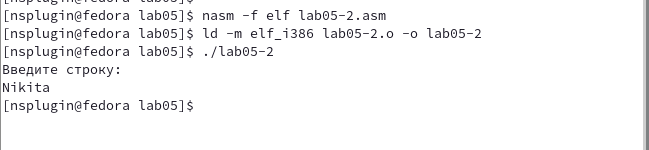


Figure 8: Проверка программы lab05-2.asm

1. В файле lab5-2.asm я заменил подпрограмму sprintLF на sprint (рис. [[9](#fig:009)]). Затем я снова собрал исполняемый файл (рис. [[10](#fig:010)]). Теперь после вывода строки она не завершается символом перехода на новую строку.

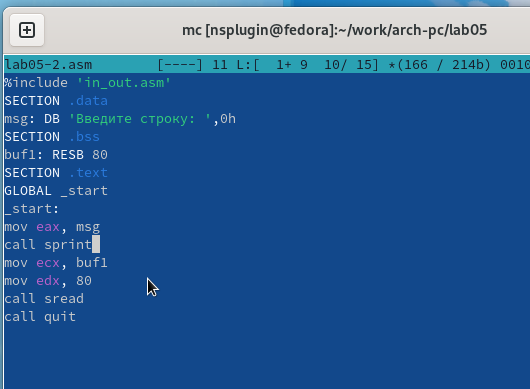


Figure 9: Программа lab05-2.asm

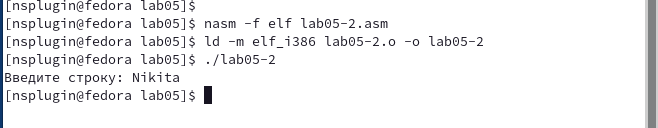


Figure 10: Проверка программы lab05-2.asm

## 4.3 Задание для самостоятельной работы

1. Скопировал программу lab05-1.asm и изменил код, чтобы программа выводила приглашение типа “Введите строку:”, затем считывала строку с клавиатуры и выводила введенную строку на экран. (рис. [[11](#fig:011)], рис. [[12](#fig:012)])



Figure 11: Программа lab05-3.asm

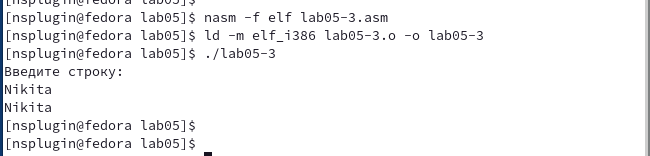


Figure 12: Проверка программы lab05-3.asm

1. Также я скопировал программу lab05-2.asm и внес соответствующие изменения в код, чтобы программа выводила приглашение типа “Введите строку:”, затем считывала строку с клавиатуры и выводила введенную строку на экран.(рис. [[13](#fig:013)], рис. [[14](#fig:014)])

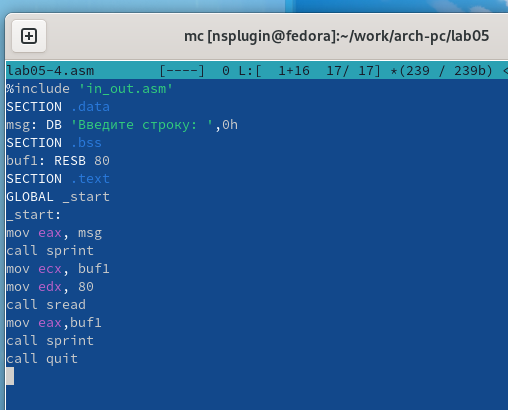


Figure 13: Программа lab05-4.asm

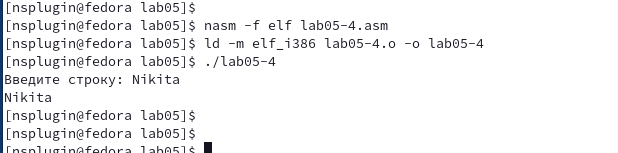


Figure 14: Проверка программы lab05-4.asm

Отличие этих двух реализаций заключается в том, что файл in\_out.asm содержит уже готовые подпрограммы для обеспечения ввода/вывода. Таким образом, нам остается только разместить данные в нужных регистрах и вызвать желаемую подпрограмму с помощью инструкции call.

# 5 Выводы

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

# 6 Источники

1. [Архитектура ЭВМ](https://esystem.rudn.ru/course/view.php?id=108)