

# **Лабораторная работа №5**

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

Карпова Есения Алексеевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>9</b>
<b>5</b>	<b>Выводы</b>	<b>13</b>
	<b>Список литературы</b>	<b>14</b>

## Список иллюстраций

4.1	Открытие Midnight Commander . . . . .	9
4.2	Создание папки lab05 . . . . .	9
4.3	Создание файла lab5-1.asm . . . . .	9
4.4	Открытие файла lab5-1.asm . . . . .	9
4.5	Редактирую файл . . . . .	10
4.6	Компиляция файла и передача на обработку компоновку . . . . .	10
4.7	Исполнени файла . . . . .	10
4.8	Размещение файла . . . . .	10
4.9	Редактирование файла . . . . .	10
4.10	Исполнение файла . . . . .	11
4.11	Замена подпрограммы . . . . .	11
4.12	Исполнение файла с другой подпрограммой . . . . .	11
4.13	Создание копии файла . . . . .	11
4.14	Редактирование файла . . . . .	11
4.15	Исполнение файла . . . . .	12
4.16	Копирование файла . . . . .	12
4.17	Редактирование файла . . . . .	12
4.18	Исполнение файла . . . . .	12

## **Список таблиц**

# 1 Цель работы

Целью данной лабораторной работы является приобретение практических навыков работы в Midnight Commander, освоение инструкций языка ассемблера `mov` и `int`

## 2 Задание

1. Основы работы с тс
2. Структура программы на языке ассемблера NASM
3. Подключение внешнего файла
4. Выполнение заданий для самостоятельной работы

### 3 Теоретическое введение

Midnight Commander (mc) - это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управлению файловой системой, т.е. mc является файловым менеджером. MC позволяет сделать работу с файлами более удобной и наглядной. Программа на языке ассемблера NASM, как правило, состоит из трех секций: секция кода программы (SECTION.text), секция инициализированных (известных во время работы 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). Для объявления инициализированных данных в секции .data используются директивы DB, DW, DD, DQ и DT, которые резервируют память и указывают, какие значения должны храниться в этой памяти: • DB (define byte) — определяет переменную размером в 1 байт; • DW (define word) — определяет переменную размером в 2 байта (слово); • DD (define double word) — определяет переменную размером в 4 байта (двойное слово); • DQ (define quad word) — определяет переменную размером в 8 байт (учетверённое слово); • DT (define ten bytes) — определяет переменную размером в 10 байт. Директивы используются для объявления простых переменных и для объявления массивов. Для определения строк принято использовать директиву DB в связи с особенностями хранения данных в оперативной памяти. Синтаксис директив определения данных следующий: DB [ , ] [ , ] 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 — источ-

ник. В качестве операнда могут выступать регистры (register), ячейки памяти (memory) и непосредственные значения (const). Инструкция языка ассемблера `int` предназначена для вызова прерывания с указанным номером. В общем виде она записывается в виде `int n`. Здесь `n` — номер прерывания, принадлежащий диапазону 0–255. При программировании в Linux с использованием вызовов ядра `sys_calls` `n=80h` (принято задавать в шестнадцатеричной системе счисления).



## 4 Выполнение лабораторной работы

### 1. Основы работы с Midnight Commander

#### 1.1. Открываю Midnight Commander(рис. 4.1).

Открытие Midnight Commander

Рис. 4.1: Открытие Midnight Commander

1.2. Перехожу в каталог ~/work/arch-pc и с помощью функциональной клавиши F7 создаю папку lab05 (рис. 4.1).

Создание папки lab05

Рис. 4.2: Создание папки lab05

1.3. Пользуясь строкой ввода и командой touch создаю файл lab5-1.asm (рис. 4.3).

Создание файла lab5-1.asm

Рис. 4.3: Создание файла lab5-1.asm

### 2. Структура программы на языке ассемблера NASM

2.1 С помощью функциональной клавиши F4 открываю файл lab5-1.asm для редактирования во встроенном редакторе (рис. 4.4).

Открытие файла lab5-1.asm

Рис. 4.4: Открытие файла lab5-1.asm

2.2. Ввожу в файл код программы для запроса строки у пользователя (рис. 4.5).

Редактирую файл

Рис. 4.5: Редактирую файл

2.3. Оттранслирую текст программы lab5-1.asm в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл.(рис. 4.6).

Компиляция файла и передача на обработку компоновку

Рис. 4.6: Компиляция файла и передача на обработку компоновку

2.4. На запрос программы ввожу свое ФИО (рис. 4.7).

Исполнени файла

Рис. 4.7: Исполнени файла

### 3. Подключение внешнего файла

3.1. Скачиваю файл in\_out.asm со страницы курса в ТУИС. Размещаю файл в тот же каталог, где лежит файл с программой (рис. 4.8).

Размещение файла

Рис. 4.8: Размещение файла

3.2. Изменяю содержимое файла lab5-2.asm во встроенном редакторе, чтобы в программе использовались подпрограммы из внешнего файла in\_out.asm (рис. 4.9).

Редактирование файла

Рис. 4.9: Редактирование файла

3.3. Создаю исполняемый файл и проверяю его работу (рис. 4.10).

Исполнение файла

Рис. 4.10: Исполнение файла

3.4. В файле lab5-2.asm заменяю подпрограмму sprintLF на sprint (рис. 4.11).

Замена подпрограммы

Рис. 4.11: Замена подпрограммы

3.5. Создаю исполняемый файл и проверьте его работу (рис. 4.12).

Исполнение файла с другой подпрограммой

Рис. 4.12: Исполнение файла с другой подпрограммой

Разница между первым исполняемым файлом и вторым в том, что запуск первого запрашивает ввод с новой строки, а программа, которая выполняется при запуске второго, запрашивает ввод без переноса на новую строку

#### 4. Задание для самостоятельной работы

4.1. Создаю копию файла lab5-1.asm с именем lab5-1-1.asm с помощью функциональной клавиши F5 (рис. 4.13).

Создание копии файла

Рис. 4.13: Создание копии файла

4.2. С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Изменяю программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку (рис. 4.14).

Редактирование файла

Рис. 4.14: Редактирование файла

4.3. Создаю объектный файл lab5-1-1.o, отдаю его на обработку компоновщику, получаю исполняемый файл lab5-1-1, запускаю полученный исполняемый файл. Программа запрашивает ввод, ввожу свои ФИО, далее программа выводит введенные мною данные (рис. 4.15).

Исполнение файла

Рис. 4.15: Исполнение файла

4.4. Создаю копию файла lab5-2.asm с именем lab5-2-1.asm с помощью функциональной клавиши F5 (рис. 4.16).

Копирование файла

Рис. 4.16: Копирование файла

4.5. С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Изменяю программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку (рис. 4.17).

Редактирование файла

Рис. 4.17: Редактирование файла

4.6. Создаю объектный файл lab5-2-1.o, отдаю его на обработку компоновщику, получаю исполняемый файл lab5-2-1, запускаю полученный исполняемый файл. Программа запрашивает ввод без переноса на новую строку, ввожу свои ФИО, далее программа выводит введенные мною данные (рис. 4.18).

Исполнение файла

Рис. 4.18: Исполнение файла

## 5 Выводы

При выполнении лабораторной работы я приобрела практические нааыки работы с Midnight Commander, а также освоила инструкции языка ассемблера mov и int.

# Список литературы

Демидова А.В. - Лабораторная работа №5