

Лабораторная работа №5

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

Борисенкова София Павловна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выполнение задания для самостоятельной работы	10
4	Выводы	12

Список иллюстраций

2.1	Запуск Midnight commander	6
2.2	Интерфейс midnight commander	6
2.3	Переход в нужный каталог (~/.work/arch-pc)	6
2.4	Создание папки	6
2.5	Создание файла lab5-1.asm с помощью команды touch прямо в mc	6
2.6	Выбор текстового редактора	7
2.7	Редактирование файла lab5-1.asm	7
2.8	Проверка успешного редактирования	7
2.9	Компиляция файла с помощью nasm	7
2.10	Сборка исполняемого файла с помощью ld	7
2.11	Взаимодействие с программой	7
2.12	Открытие папки с файлом in_out.asm в правой панели	8
2.13	Копирование файла с помощью F6	8
2.14	Копирование файла с помощью F5	8
2.15	Текущий вид рабочей папки	8
2.16	Редактирование файла lab5-2.asm	8
2.17	Создание исполняемого файла	8
2.18	Запуск исполняемого файла	8
2.19	Изменение файла lab5-2.asm	9
2.20	Запуск изменённого файла	9
3.1	Создание копии файла lab5-1.asm	10
3.2	Изменение файла lab5-1-1.asm	10
3.3	Создание исполняемого файла	10
3.4	Проверка работы программы	10
3.5	Создание копии файла lab5-2.asm	11
3.6	Изменение файла lab5-2-1.asm	11
3.7	Создание исполняемого файла	11
3.8	Проверка работы программы	11

Список таблиц

1 Цель работы

Ознакомиться с программой Midnight commander и освоить написание программ на языке ассемблера с помощью инструкций `mov` и `int`

2 Выполнение лабораторной работы

Для начала выполнения лабораторной работы нам необходимо открыть Midnight commander с помощью команды `mc` (Рис. 2.1):

Запуск Midnight commander

Рис. 2.1: Запуск Midnight commander

После ввода команды мы увидим такой интерфейс (Рис. 2.2):

Интерфейс midnight commander

Рис. 2.2: Интерфейс midnight commander

С помощью стрелок и клавиши `Enter` перейдём в каталог `~/work/arch-rc` (Рис. 2.3):

Переход в нужный каталог (`~/work/arch-rc`)

Рис. 2.3: Переход в нужный каталог (`~/work/arch-rc`)

Создадим папку `lab05` с помощью клавиши `F7` (Рис. 2.4):

Создание папки

Рис. 2.4: Создание папки

Теперь с помощью команды `touch` создадим файл `lab5-1.asm` (Рис. 2.5):

Создание файла `lab5-1.asm` с помощью команды `touch` прямо в `mc`

Рис. 2.5: Создание файла `lab5-1.asm` с помощью команды `touch` прямо в `mc`

Теперь откроем только что созданный файл с помощью редактор nano (Рис. 2.6):

Выбор текстового редактора

Рис. 2.6: Выбор текстового редактора

Теперь отредактируем файл и поместим в него следующий код (Рис. 2.7):

Редактирование файла lab5-1.asm

Рис. 2.7: Редактирование файла lab5-1.asm

Теперь сохраним его (сочетанием клавиш ctrl+x и согласившись с сохранением) и с помощью F3 откроем для просмотра, чтобы убедиться, что он сохранился корректно (Рис. 2.8):

Проверка успешного редактирования

Рис. 2.8: Проверка успешного редактирования

Теперь скомпилируем его (Рис. 2.9):

Компиляция файла с помощью nasm

Рис. 2.9: Компиляция файла с помощью nasm

И соберём (Рис. 2.10):

Сборка исполняемого файла с помощью ld

Рис. 2.10: Сборка исполняемого файла с помощью ld

После этого запустим получившийся исполняемый файл и введём ФИО (Рис. 2.11):

Взаимодействие с программой

Рис. 2.11: Взаимодействие с программой

После нажатия Enter программа завершится и ничего не произойдёт. Теперь скачаем файл in_out.asm и откроем папку с ним в правой панели (Рис. 2.12):

Открытие папки с файлом in_out.asm в правой панели

Рис. 2.12: Открытие папки с файлом in_out.asm в правой панели

Скопируем его в нашу рабочую папку с помощью F6 (Рис. 2.13):

Копирование файла с помощью F6

Рис. 2.13: Копирование файла с помощью F6

Теперь сделаем копию файла lab5-1.asm с помощью команды F5. Назовём копию lab5-2.asm (Рис. 2.14):

Копирование файла с помощью F5

Рис. 2.14: Копирование файла с помощью F5

Теперь наша папка выглядит следующим образом (Рис. 2.15):

Текущий вид рабочей папки

Рис. 2.15: Текущий вид рабочей папки

Откроем в текстовом редакторе файл lab5-2.asm и напишем туда следующий код (Рис. 2.16):

Редактирование файла lab5-2.asm

Рис. 2.16: Редактирование файла lab5-2.asm

После чего создадим исполняемый файл с помощью nasm и ld (Рис. 2.17):

Создание исполняемого файла

Рис. 2.17: Создание исполняемого файла

Запустим созданный файл (Рис. 2.18):

Запуск исполняемого файла

Рис. 2.18: Запуск исполняемого файла

Он работает также, как и файл lab5-1, но использует для работы сторонний файл. Попробуем теперь вместо команды `sprintLF` использовать просто команду `sprint` (Рис. 2.19):

Изменение файла lab5-2.asm

Рис. 2.19: Изменение файла lab5-2.asm

Точно также соберём исполняемый файл и запустим его (Рис. 2.20):

Запуск изменённого файла

Рис. 2.20: Запуск изменённого файла

Как мы видим, теперь нет переноса на следующую строку. Этим и отличаются команды `sprintLF` от `sprint`. Первая добавляет перенос после текста, а вторая нет

3 Выполнение задания для самостоятельной работы

Теперь создадим с помощью F5 копию файла lab5-1.asm (Рис. 3.1):

Создание копии файла lab5-1.asm

Рис. 3.1: Создание копии файла lab5-1.asm

Изменим копию так, чтобы она выводила тот текст, который получила на ввод. Для этого перед системным вызовом `exit` вставим текст с системным вызовом `write`. Он очень похож на системный вызов `write`, который уже был в коде, но есть несколько отличий. Так, мы перемещаем адрес строки `buf1` в `ecx` и размер строки `buf1` (80) в `edx` (Рис. 3.2):

Изменение файла lab5-1-1.asm

Рис. 3.2: Изменение файла lab5-1-1.asm

Сохраним изменения и создадим исполняемый файл (Рис. 3.3):

Создание исполняемого файла

Рис. 3.3: Создание исполняемого файла

Запустим его и проверим, что всё работает (Рис. 3.4):

Проверка работы программы

Рис. 3.4: Проверка работы программы

Теперь создадим с помощью F5 копию файла lab5-2.asm (Рис. 3.5):

Создание копии файла lab5-2.asm

Рис. 3.5: Создание копии файла lab5-2.asm

теперь сделаем так, чтобы этот код также выводил тот текст, что получит на ввод. Для этого перед последней строкой добавим строчку, которая записывает в `eax` адрес `buf1`, а также строчку, которая вызывает подпрограмму `sprintLF` (Рис. 3.6):

Изменение файла lab5-2-1.asm

Рис. 3.6: Изменение файла lab5-2-1.asm

Теперь создадим исполняемый файл (Рис. 3.7):

Создание исполняемого файла

Рис. 3.7: Создание исполняемого файла

Теперь запустим программу и убедимся, что она работает (Рис. 3.8):

Проверка работы программы

Рис. 3.8: Проверка работы программы

4 Выводы

В результате выполнения работы были получены навыки работы с Midnight commander, а также навыки написания простых программ ввода-вывода на языке ассемблера