

Отчёт по лабораторной работе № 6

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

Давит Оганнисян Багратович

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	7
3.1	Открываем Midnight Commander	7
3.2	Переходим в каталог ~/work/arch-pc созданный при выполнении лабораторной работы No5 и с помощью функциональной клавиши F7 создайте папку lab06	8
3.3	Пользуясь строкой ввода и командой touch создаем файл lab6-1.asm	9
3.4	С помощью функциональной клавиши F4 открываем файл lab6-1.asm для редактирования во встроенном редакторе.	10
3.5	Вводим текст программы из листинга 6.1 (можно без комментариев), сохраняем изменения и закрывем файл.	11
3.6	Оттранслируем текст программы lab6-1.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл. Программа выводит строку 'Введите строку:' и ожидает ввода с клавиатуры. На запрос вводим наши ФИО.	11
3.7	Скачиваем файл in_out.asm со страницы курса в ТУИС и копируем в наш каталог.	12
3.8	С помощью функциональной клавиши F6 создаем копию файла lab6-1.asm с именем lab6-2.asm. Выделяем файл lab6-1.asm, нажимаем клавишу F6 , вводим имя файла lab6-2.asm и нажимаем клавишу Enter	13
3.9	Исправим текст программы в файле lab6-2.asm с использованием подпрограмм из внешнего файла in_out.asm (используем подпрограммы sprintLF, sread и quit) в соответствии с листингом 6.2. Создаем исполняемый файл и проверяем его работу.	14
3.10	В файле lab6-2.asm заменим подпрограмму sprintLF на sprint. Создадим исполняемый файл и проверим его работу.	14
3.11	Задание для самостоятельной работы	15
4	Выводы	16

Список иллюстраций

3.1	Запуск Midnight Commander	7
3.2	Создание папки lab06	8
3.3	Создание файла lab06-1.asm	9
3.4	Открывание файла lab06-1.asm во встроенном редакторе	10
3.5	Текст программы	11
3.6	Запуск исполняемого файла lab6-1	11
3.7	Копирование файла in_out.asm	12
3.8	Перемещение файла lab6-1 с именем lab6-2	13
3.9	sprintLF lab6-2	14
3.10	sprint lab6-2	14
3.11	Измененный lab6-1	15
3.12	Измененный lab6-2	15

Список таблиц

1 Цель работы

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.

3 Выполнение лабораторной работы

3.1 Открываем Midnight Commander

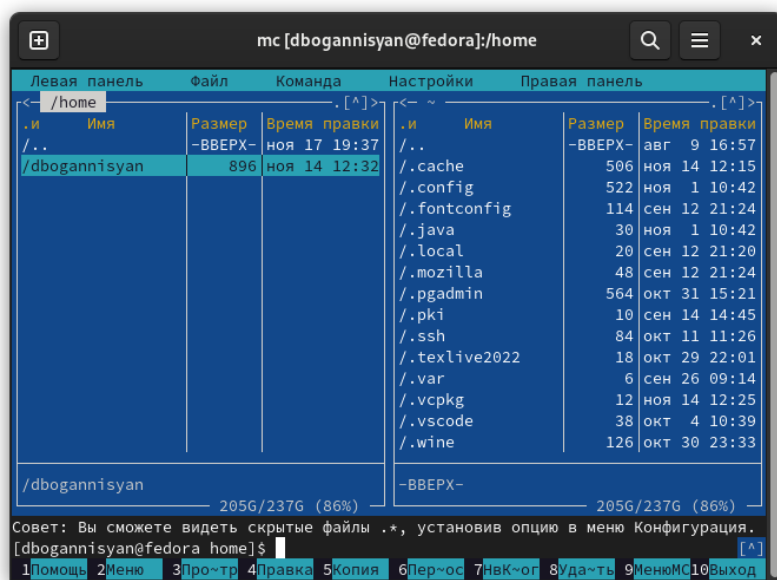


Рис. 3.1: Запуск Midnight Commander

3.2 Переходим в каталог ~/work/arch-pc созданный при выполнении лабораторной работы No5 и с помощью функциональной клавиши F7 создайте папку lab06

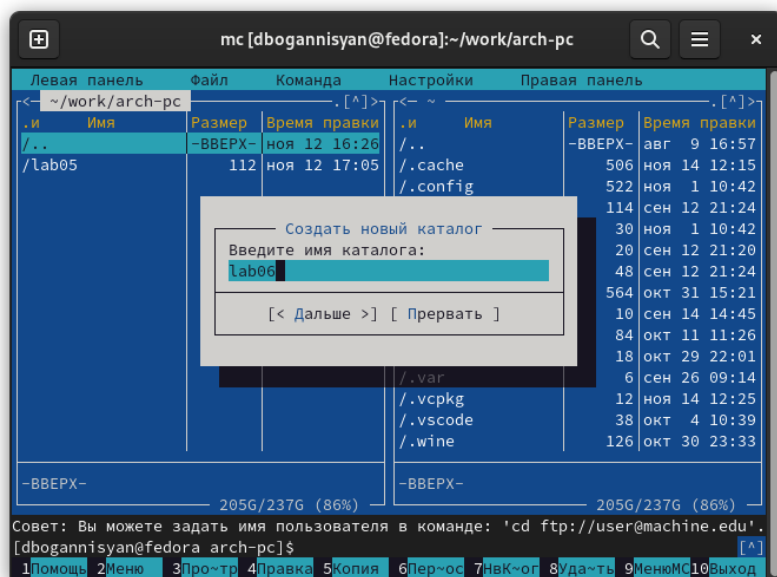


Рис. 3.2: Создание папки lab06

3.3 Пользуясь строкой ввода и командой touch создадим файл lab6-1.asm

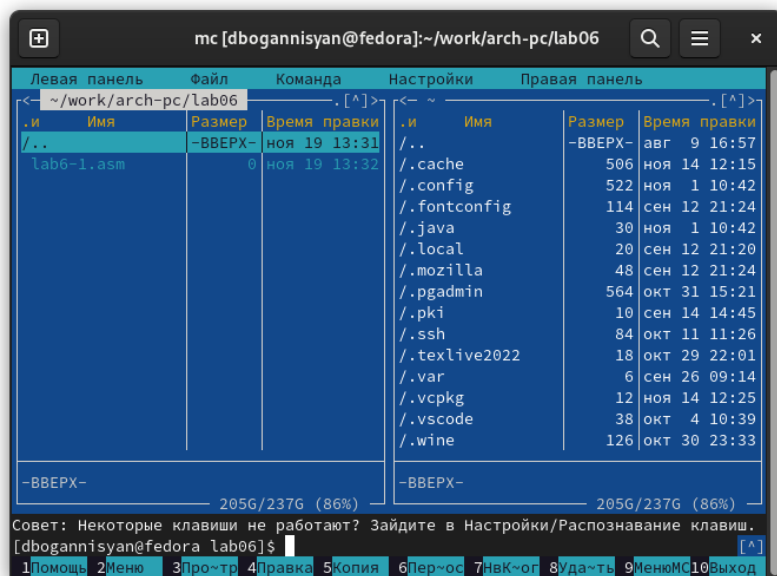
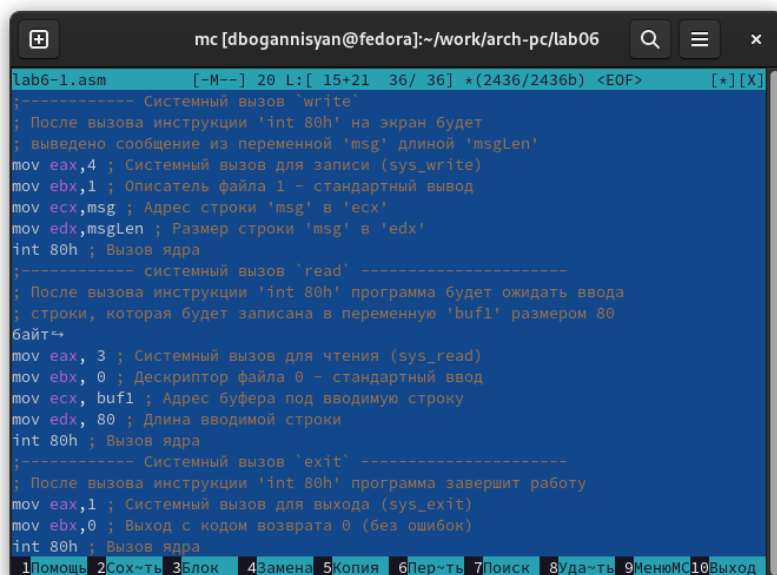


Рис. 3.3: Создание файла lab06-1.asm

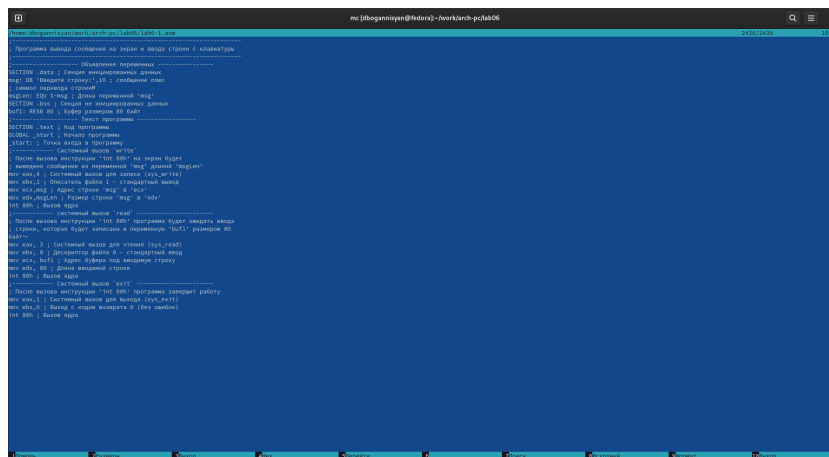
3.4 С помощью функциональной клавиши F4 открываем файл lab6-1.asm для редактирования во встроенном редакторе.



```
lab6-1.asm  [-M--] 20 L:[ 15+21 36/ 36] *(2436/2436b) <EOF>  [*] [X]
;----- Системный вызов 'write'
; После вызова инструкции 'int 80h' на экран будет
; выведено сообщение из переменной 'msg' длиной 'msgLen'
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла 1 - стандартный вывод
mov ecx,msg ; Адрес строки 'msg' в 'ecx'
mov edx,msgLen ; Размер строки 'msg' в 'edx'
int 80h ; Вызов ядра
;----- системный вызов 'read' -----
; После вызова инструкции 'int 80h' программа будет ожидать ввода
; строки, которая будет записана в переменную 'buf1' размером 80
байт
mov eax, 3 ; Системный вызов для чтения (sys_read)
mov ebx, 0 ; Дескриптор файла 0 - стандартный ввод
mov ecx, buf1 ; Адрес буфера под вводимую строку
mov edx, 80 ; Длина вводимой строки
int 80h ; Вызов ядра
;----- Системный вызов 'exit' -----
; После вызова инструкции 'int 80h' программа завершит работу
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра
1Помощь 2Сох-ть 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9Меню 10Выход
```

Рис. 3.4: Открывание файла lab06-1.asm во встроенном редакторе

3.5 Вводим текст программы из листинга 6.1 (можно без комментариев), сохраняем изменения и закрывем файл.



```

; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
SECTION .data ; Объявление переменных
msg DB "Введите строку:", 10 ; Сообщение "msg"
; конец переменных
SECTION .bss ; Зарезервированная память
buf1 RESB 100 ; Буфер размером 100 байт
;-----
SECTION .text ; Код программы
global _start ; Начало программы
_start: ; Точка входа в программу
;-----
; Системный вызов "write"
; После вызова инструкции "int 0x80" на экран будет
; выведено сообщение из переменной "msg" длиной "len"
mov edi, 4 ; Системный вызов для вывода (sys_write)
mov ebx, 1 ; Описание файла 1 - стандартный вывод
mov ecx, msg ; Адрес строки "msg" в "text"
mov edx, len ; Длина строки "msg" в "text"
int 0x80 ; Вызов ядра

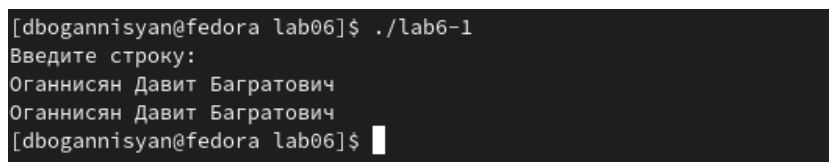
;-----
; Системный вызов "read"
; После вызова инструкции "int 0x80" программа будет ожидать ввода
; строки, которую будет заносить в переменную "buf1" размером 100
; байт.
mov ebx, 3 ; Системный вызов для чтения (sys_read)
mov ebx, 0 ; Дескриптор файла 0 - стандартный ввод
mov ecx, buf1 ; Адрес буфера под введенную строку
mov edx, 100 ; Длина введенной строки
int 0x80 ; Вызов ядра

;-----
; Системный вызов "exit"
; После вызова инструкции "int 0x80" программа завершит работу
mov ebx, 1 ; Системный вызов для выхода (sys_exit)
mov ecx, 0 ; Выход с кодом возврата 0 (без ошибок)
int 0x80 ; Вызов ядра

```

Рис. 3.5: Текст программы

3.6 Оттранслируем текст программы lab6-1.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл. Программа выводит строку 'Введите строку:' и ожидает ввода с клавиатуры. На запрос вводим наши ФИО.



```

[dbogannisyan@fedora lab06]$ ./lab6-1
Введите строку:
Оганнисян Давит Багратович
Оганнисян Давит Багратович
[dbogannisyan@fedora lab06]$

```

Рис. 3.6: Запуск исполняемого файла lab6-1

3.7 Скачиваем файл in_out.asm со страницы курса в ТУИС и копируем в наш каталог.

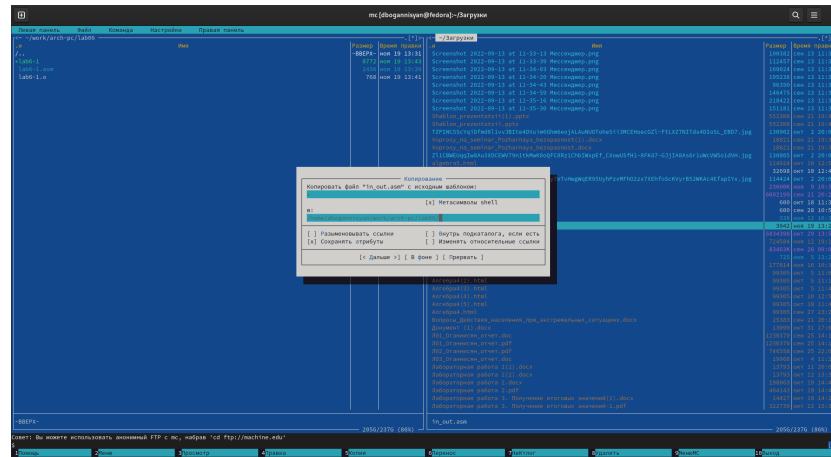


Рис. 3.7: Копирование файла in_out.asm

3.8 С помощью функциональной клавиши F6 создаем копию файла lab6-1.asm с именем lab6-2.asm.

Выделяем файл lab6-1.asm, нажимаем клавишу F6 ,

вводим имя файла lab6-2.asm и нажимаем клавишу

Enter

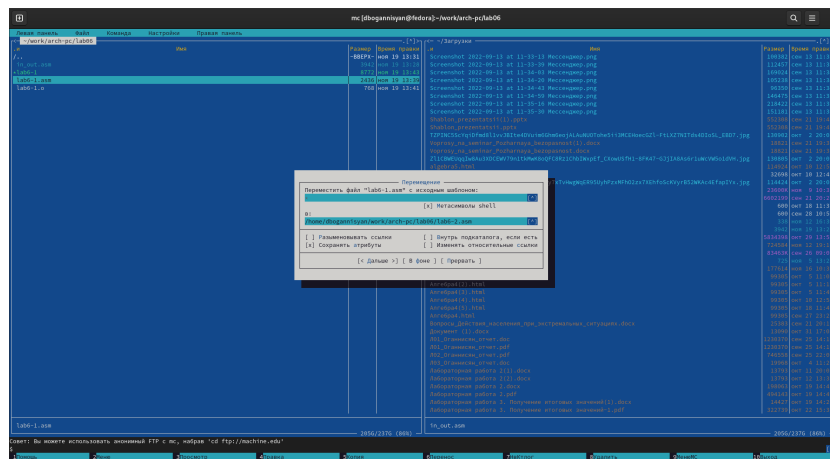


Рис. 3.8: Перемещение файла lab6-1 с именем lab6-2

3.9 Исправим текст программы в файле lab6-2.asm с использованием подпрограмм из внешнего файла in_out.asm (используем подпрограммы sprintLF, sread и quit) в соответствии с листингом 6.2. Создаем исполняемый файл и проверяем его работу.

```
[dbogannisyan@fedora lab06]$ nasm -f elf lab6-2.asm
[dbogannisyan@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[dbogannisyan@fedora lab06]$ ./lab6-2
Введите строку:
gs
```

Рис. 3.9: sprintLF lab6-2

3.10 В файле lab6-2.asm заменим подпрограмму sprintLF на sprint. Создадим исполняемый файл и проверим его работу.

```
[dbogannisyan@fedora lab06]$ nasm -f elf lab6-22.asm
[dbogannisyan@fedora lab06]$ ld -m elf_i386 -o lab6-22 lab6-22.o
[dbogannisyan@fedora lab06]$ ./lab6-22
Введите строку: gs
[dbogannisyan@fedora lab06]$
```

Рис. 3.10: sprint lab6-2

Разница только в том, что в этом случае ввод осуществляется в одну строчку

3.11 Задание для самостоятельной работы

```
$ nasm -f elf lab6-11.asm
$ ld -m elf_i386 -o lab6-11 lab6-11.o
$ ./lab6-11
Введите строку:
Оганнисян
Оганнисян
```

Рис. 3.11: Измененный lab6-1

```
[dbogannisyan@fedora lab06]$ nasm -f elf lab6-222.asm
[dbogannisyan@fedora lab06]$ ld -m elf_i386 -o lab6-222 lab6-222.o
[dbogannisyan@fedora lab06]$ ./lab6-222
Введите строку:
Оганнисян
Оганнисян
MMMMMM[dbogannisyan@fedora lab06]$
```

Рис. 3.12: Измененный lab6-2

4 Выводы

Я приобрел практические навыки работы в Midnight Commander и освоил инструкции языка ассемблера `mov` и `int`.