

# **Лабораторная работа №6**

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

Виктор Максимович Кадров

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>6</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>5</b>	<b>Задания для самостоятельной работы</b>	<b>11</b>
<b>6</b>	<b>Выводы</b>	<b>14</b>

# Список иллюстраций

4.1	Создание папки . . . . .	7
4.2	Код программы . . . . .	8
4.3	Выполнение программы . . . . .	8
4.4	Измененный код . . . . .	9
4.5	Проверка исполнения . . . . .	9
4.6	Замена вызываемой функции . . . . .	10
4.7	Результат выполнения . . . . .	10
5.1	Код программы . . . . .	11
5.2	Исполнение программы . . . . .	12
5.3	Код lab06-2 . . . . .	12
5.4	Проверка результата . . . . .	13

# 1 Цель работы

Приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера `mov` и `int`.

## 2 Задание

1. Создания программы ввода-вывода
2. Создание программы с использованием внешнего файла
3. Выполнение задания для саомстоятельной работы

### 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).

Для объявления инициированных данных в секции .data используются директивы DB, DW, DD, DQ и DT, которые резервируют память и указывают, какие значения должны храниться в этой памяти.

## 4 Выполнение лабораторной работы

Создаем папку lab06 и файл lab06-1.asm в ней (рис. 4.1). Пишем код программы в созданный файл (рис. 4.2). Создаем исполняемый файл и проверяем, что ввод и вывод происходит корректно (рис. 4.3). Изменяем код программы так, чтобы для выполнения ввода и вывода использовался код из внешнего файла (рис. 4.4). Выполняем его (рис. 4.5). Заменяем в кода программы `sprintf` на `sprint` (рис. 4.6). После выполнения (рис. 4.7) можно заметить, что ввод происходит на той же строке, что и вывод.

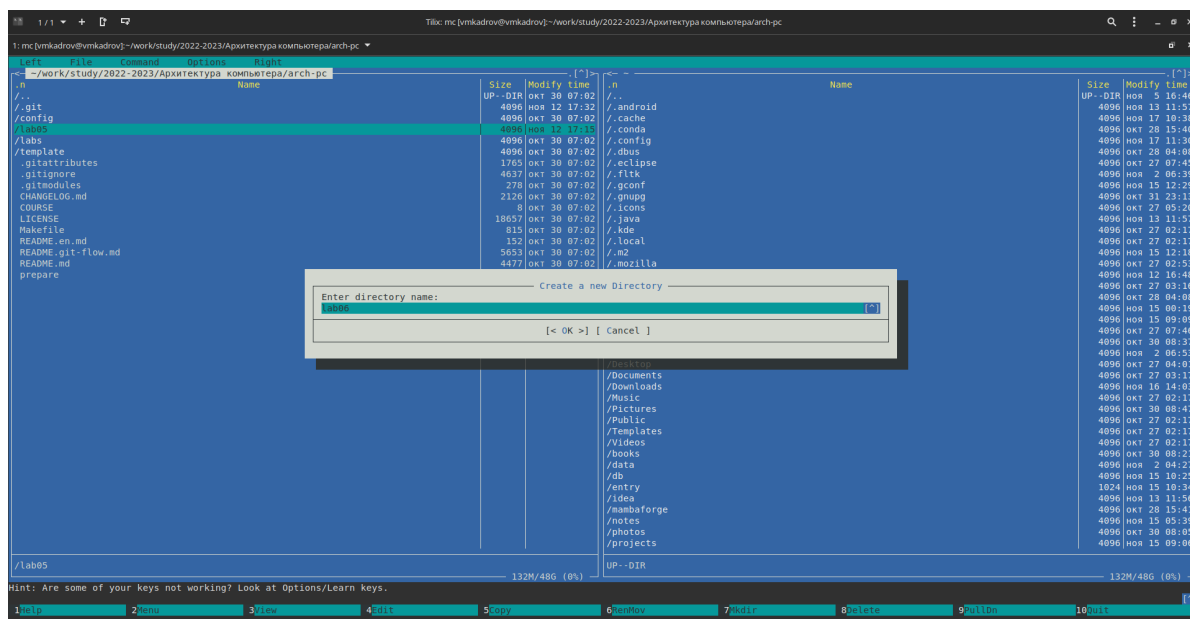


Рис. 4.1: Создание папки





```
1: mc [vmkadrov@vmkadrov] ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06
/home/vmkadrov/work/study/2022-2023/Архитектура компьютера/arch-pc [M~] 4 L| 1+19 20/ 20 |~(1199/12675) 0099 vx063 [~]A
.....
; Программа вывода сообщения на экран и ввода строки с клавиатуры
.....
#include "in_out.asm" ; подключение внешнего файла
.....
SECTION .data ; Секция инициализированных данных
    msg: DB "Введите строку" , 0h ; сообщение
.....
SECTION .bss ; Секция не инициализированных данных
    buf1: RESB 80 ; буфер размером 80 байт
.....
SECTION .text ; Код программы
GLOBAL start ; Начало программы
start: ; Точка входа в программу
    mov eax, msg ; запись адреса выводимого сообщения в EAX
    call sprintf ; вызов подпрограммы печати сообщения
    mov ecx, buf1 ; запись адреса переменной в EAX
    mov edx, 80 ; запись длины вводимого сообщения в EDI
    call read ; вызов подпрограммы ввода сообщения
    call quit ; вызов подпрограммы завершения
```

Рис. 4.4: Измененный код

```
1: vmkadrov@vmkadrov:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06
(base) + lab06 git:(master) # nasm -f elf lab06-2.asm
(base) + lab06 git:(master) # ld -m elf_i386 lab06-2.o -o lab06-2
(base) + lab06 git:(master) # ./lab06-2
Введите строку:
Карпов Виктор Максимович
(base) + lab06 git:(master) #
```

Рис. 4.5: Проверка исполнения

```
1: mc [vmkadrov@vmkadrov:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06]
/home/vmkadrov/work/study/2022-2023/Архитектура компьютера/arch-pc[----] 15 L: [ 1+15 16/ 28] *(674 /12656) 6832 0x020 [*]1X
; Программа вывода сообщения на экран и ввода строки с клавиатуры
#include "in_out.asm" ; подключение внешнего файла

SECTION .data ; Секция инициализированных данных
msg: DB "Введите строку: ",0h ; сообщение

SECTION .bss ; Секция не инициализированных данных
buf1: RESB 80 ; буфер размером 80 байт

SECTION .text ; Код программы
GLOBAL start ; объявление программы
start: ; точка входа в программу
mov eax, msg ; запись адреса выводимого сообщения в EAX
call sprintf ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса буфера в EAX
mov edx, 80 ; запись длины выводимого сообщения в EDI
call fwrite ; вызов подпрограммы вывода сообщения
call quit ; вызов подпрограммы завершения
```

Рис. 4.6: Замена вызываемой функции

```
1: vmkadrov@vmkadrov:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06
- lab06 git:(master) * name: f elf: lab06-2.asm 66 \
ld -m elf_i386 lab06-2.o -o lab06-2 66 \
./lab06-2
Введите строку: Кадров Виктор Максимович
- lab06 git:(master) *
```

Рис. 4.7: Результат выполнения

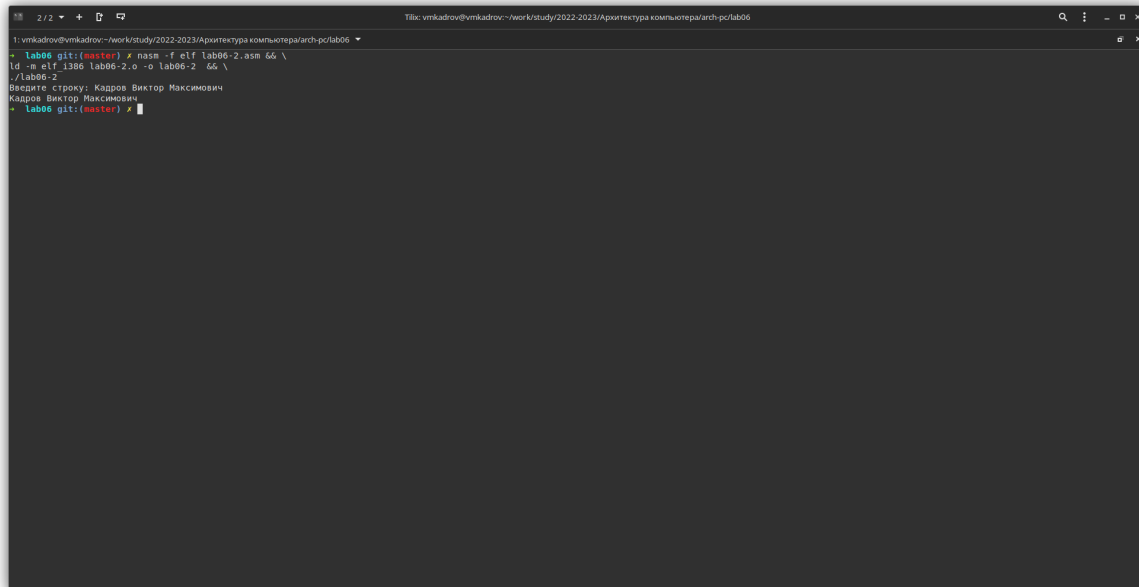
## 5 Задания для самостоятельной работы

Изменяем код lab06-1 так, чтобы введенная строка выводилась на экран (рис. 5.1). Проверяем, что все выполняется корректно (рис. 5.2). Аналогично меняем код lab06-2 (рис. 5.3) и проверяем результат (рис. 5.4).

[illegible]

Рис. 5.1: Код программы





```
1: vmkadrov@vmkadrov:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06
+ lab06 git:(master) x nasm -f elf lab06-2.asm 66 \
ld -m elf_i386 lab06-2.o -o lab06-2 66 \
./lab06-2
Введите строку: Кадров Виктор Максимович
Кадров Виктор Максимович
+ lab06 git:(master) x
```

Рис. 5.4: Проверка результата

## 6 Выводы

В ходе выполнения лабораторной работы были приобретены практические навыки работы в Midnight Commander. Также были освоены инструкции языка ассемблера `mov` и `int`.