

# **Лабораторная работа №7**

**Midnight Commander (mc).**

Вершинина Ангелина 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>8</b>
<b>5</b>	<b>Выводы</b>	<b>13</b>
	<b>Список литературы</b>	<b>14</b>

## Список иллюстраций

4.1	Открытие Midnight Commander . . . . .	8
4.2	Переход в каталог . . . . .	9
4.3	Создание каталога . . . . .	9
4.4	Создание файла . . . . .	10
4.5	Открытие файла . . . . .	10
4.6	Ввод текста программы . . . . .	11
4.7	Сохранение текста программы . . . . .	11
4.8	Редактирование файла . . . . .	12

## Список таблиц

# 1 Цель работы

Освоение основных возможностей командной оболочки Midnight Commander.  
Приобретение навыков практической работы по просмотру каталогов и файлов;  
манипуляций с ними

## 2 Задание

Работа с файлом в тс

### **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.

# 4 Выполнение лабораторной работы

Открою Midnight Commander (рис. [4.1])

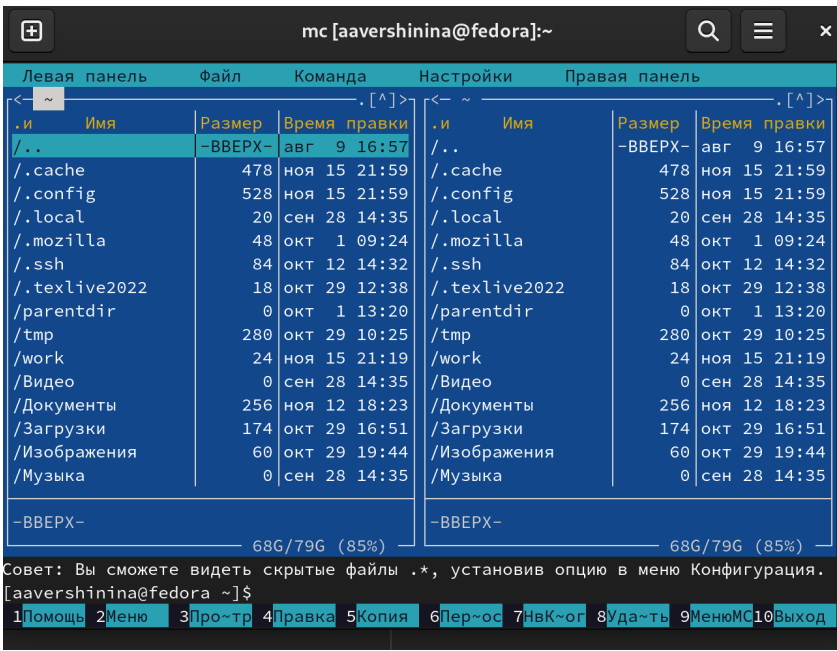


Рис. 4.1: Открытие Midnight Commander

Пользуясь клавишами **⌘**, **⌘** и **Enter** перейду в каталог `~/work/arch-pc` (рис. [4.2])



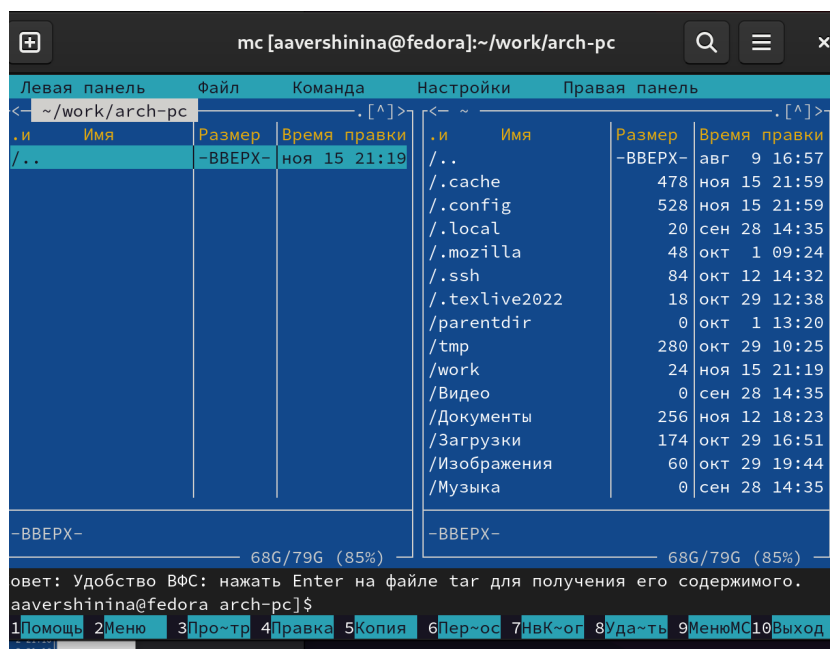


Рис. 4.2: Переход в каталог

С помощью функциональной клавиши F7 создам папку lab06 (рис. [4.3]) и перейду в созданный каталог.

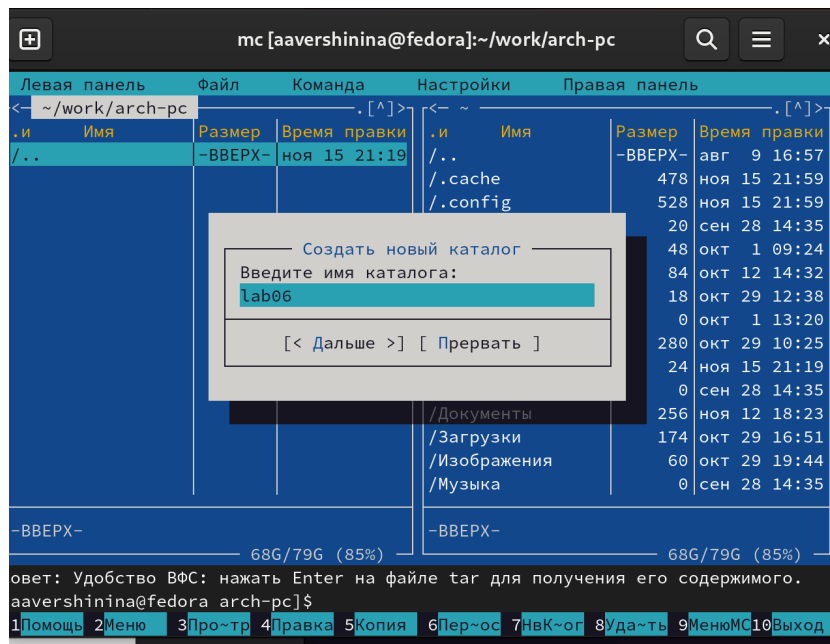


Рис. 4.3: Создание каталога

Пользуясь строкой ввода и командой touch создам файл lab6-1.asm (рис. [4.4])

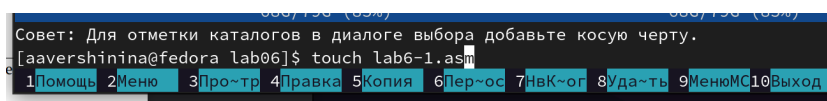


Рис. 4.4: Создание файла

С помощью функциональной клавиши F4 открою файл lab6-1.asm для редактирования во встроенном редакторе (рис. [4.5])

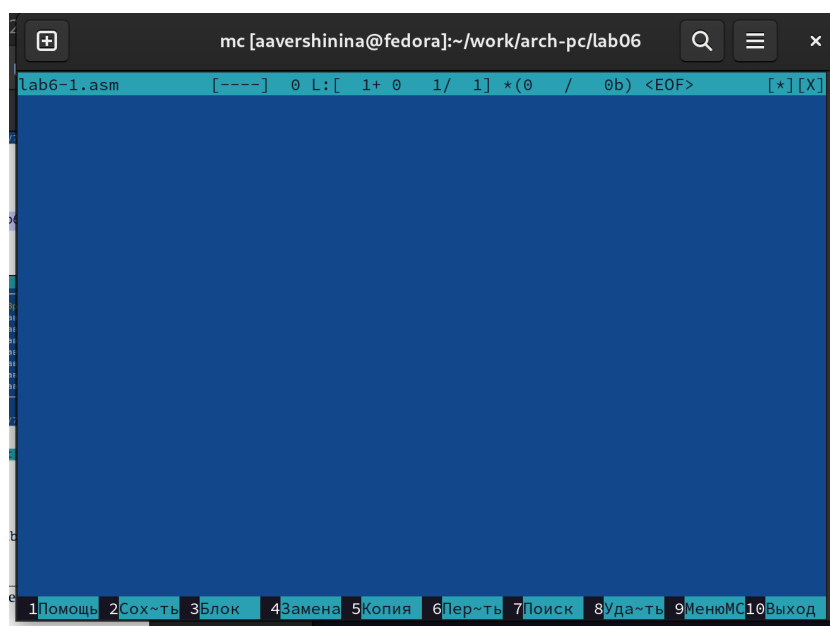
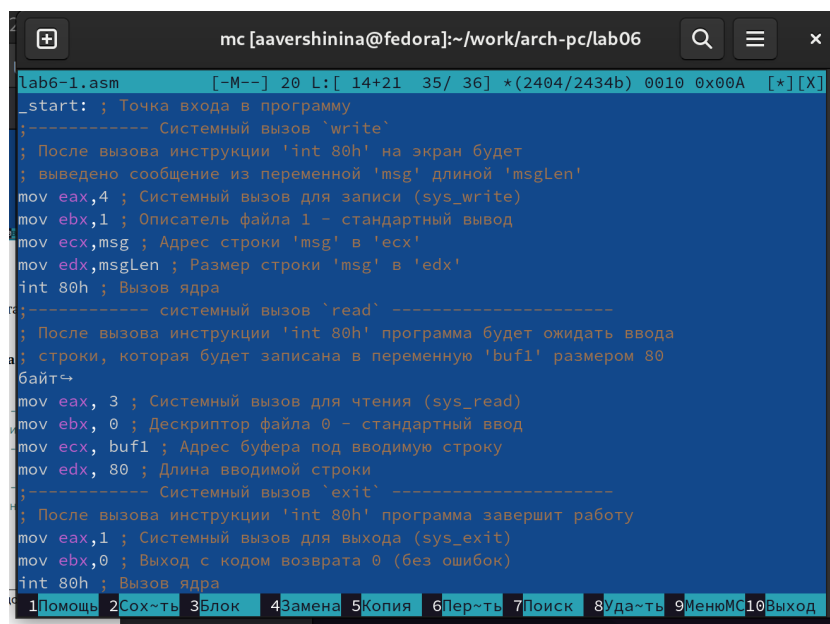


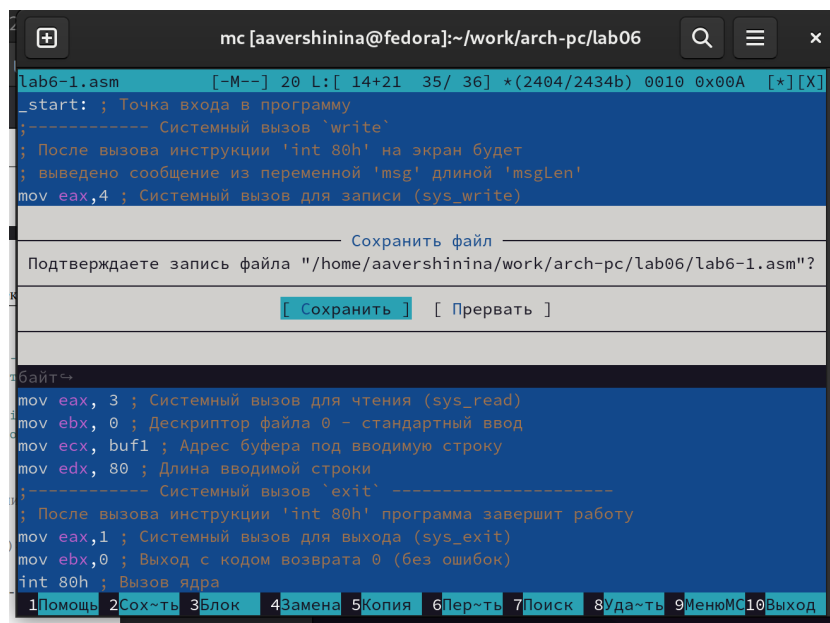
Рис. 4.5: Открытие файла

Введу текст программы из листинга 6.1 (рис. [4.6]), со- храню изменения (рис. [4.7]) и закрою файл.



```
lab6-1.asm [-M--] 20 L:[ 14+21 35/ 36] *(2404/2434b) 0010 0x00A [*][X]
_start: ; Точка входа в программу
;----- 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 ;Descriptor файла 0 - стандартный ввод
mov ecx, buf1 ; Адрес буфера под вводимую строку
mov edx, 80 ; Длина вводимой строки
;----- 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 9МенюMC10Выход
```

Рис. 4.6: Ввод текста программы



```
lab6-1.asm [-M--] 20 L:[ 14+21 35/ 36] *(2404/2434b) 0010 0x00A [*][X]
_start: ; Точка входа в программу
;----- 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)
;----- 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 ;Descriptor файла 0 - стандартный ввод
mov ecx, buf1 ; Адрес буфера под вводимую строку
mov edx, 80 ; Длина вводимой строки
;----- 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 9МенюMC10Выход
```

Сохранить файл

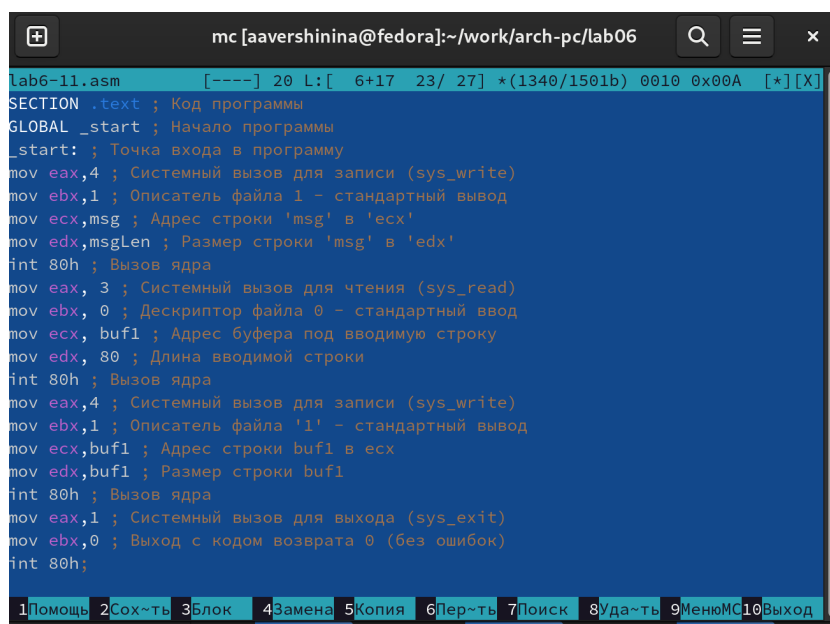
Подтверждаете запись файла "/home/aaavershinina/work/arch-pc/lab06/lab6-1.asm"?

[ Сохранить ] [ Прервать ]

Рис. 4.7: Сохранение текста программы

Создам копию файла lab6-1.asm с именем lab6-11.asm с помощью функциональной клавиши F5. С помощью функциональной клавиши F4 открываю созданный

файл для редактирования. Изменяю программу так, чтобы кроме вывода приглашения и запроса ввода, производился вывод вводимой пользователем строки (рис. [4.8])



```
lab6-11.asm [----] 20 L:[ 6+17 23/ 27] *(1340/1501b) 0010 0x00A [*][X]
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
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
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
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла '1' - стандартный вывод
mov ecx,buf1 ; Адрес строки buf1 в ecx
mov edx,buf1 ; Размер строки buf1
int 80h ; Вызов ядра
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h;
```

1Помощь 2Сох-ть 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюMC10Выход

Рис. 4.8: Редактирование файла

## 5 Выводы

При выполнении данной лабораторной работы я освоила основные возможности командной оболочки Midnight Commander. Приобрела навыки практической работы по просмотру каталогов и файлов; манипуляций с ними

## **Список литературы**