

# Командная оболочка Midnight Commander

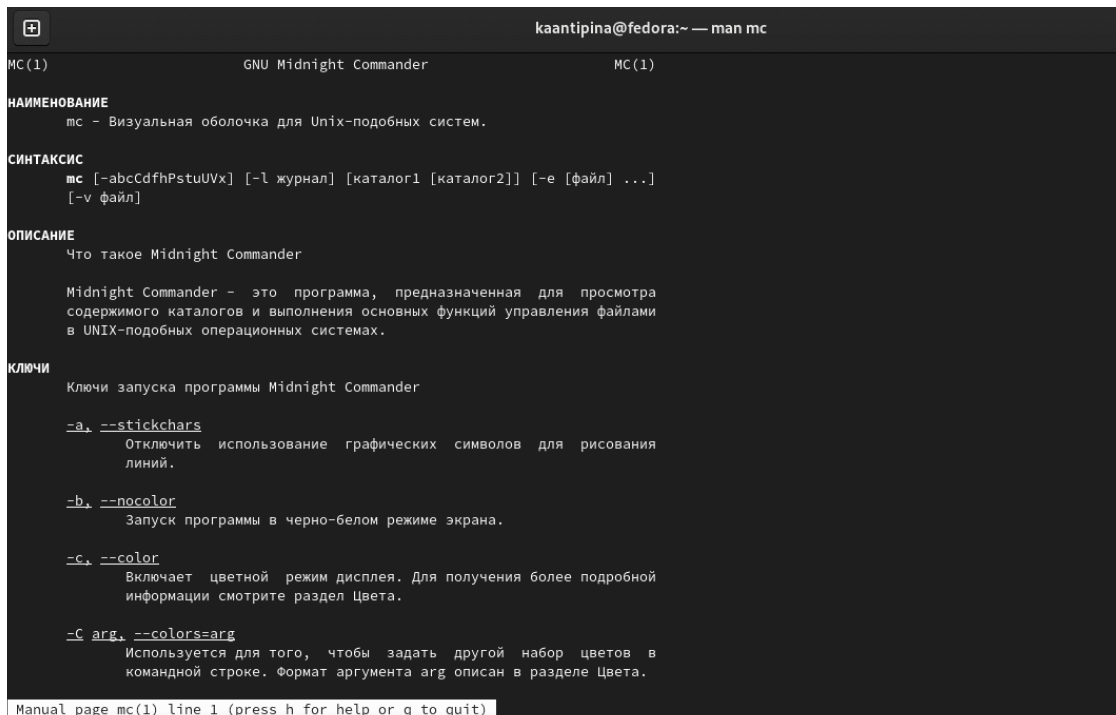
12 мая, 2022, Москва, Россия

## Цель работы

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

## Выполнение лабораторной работы

Изучим информацию о mc, вызвав в командной строке `man mc`. (рис. -@fig:001)



```
kaantipina@fedora:~ — man mc
mc(1)          GNU Midnight Commander          mc(1)

НАИМЕНОВАНИЕ
  mc - Визуальная оболочка для Unix-подобных систем.

СИНТАКСИС
  mc [-abcCdfhPstuUVx] [-l журнал] [каталог1 [каталог2]] [-e [файл] ...]
  [-v файл]

ОПИСАНИЕ
  Что такое Midnight Commander

  Midnight Commander - это программа, предназначенная для просмотра
  содержимого каталогов и выполнения основных функций управления файлами
  в UNIX-подобных операционных системах.

КЛЮЧИ
  Ключи запуска программы Midnight Commander

  -a, --stickchars
      Отключить использование графических символов для рисования
      линий.

  -b, --nocolor
      Запуск программы в черно-белом режиме экрана.

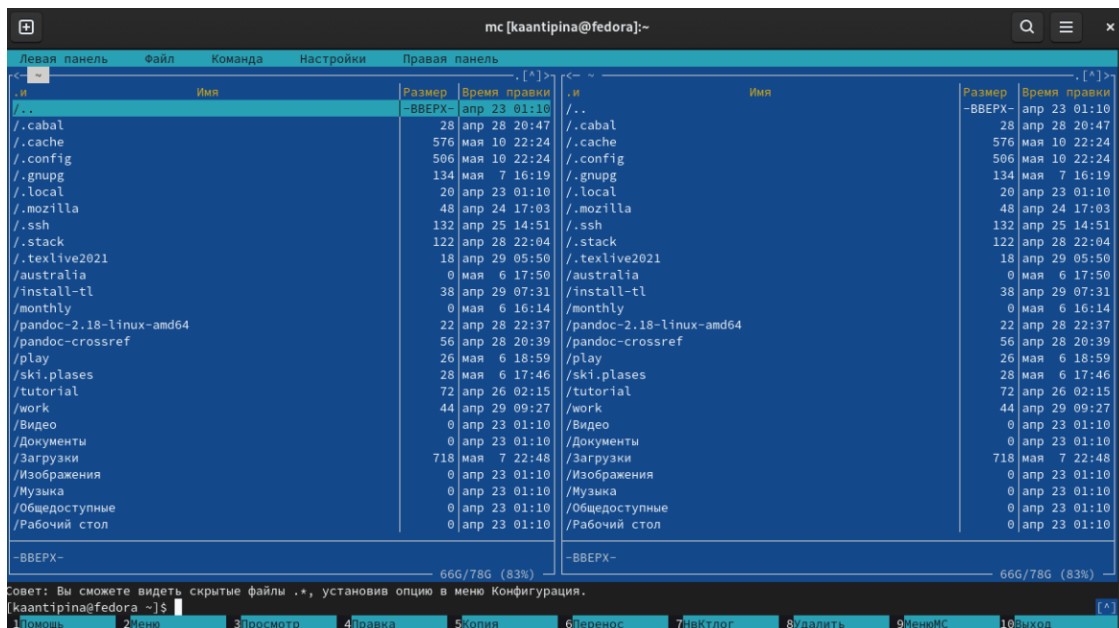
  -c, --color
      Включает цветной режим дисплея. Для получения более подробной
      информации смотрите раздел Цвета.

  -C arg, --colors=arg
      Используется для того, чтобы задать другой набор цветов в
      командной строке. Формат аргумента arg описан в разделе Цвета.

Manual page mc(1) line 1 (press h for help or q to quit)
```

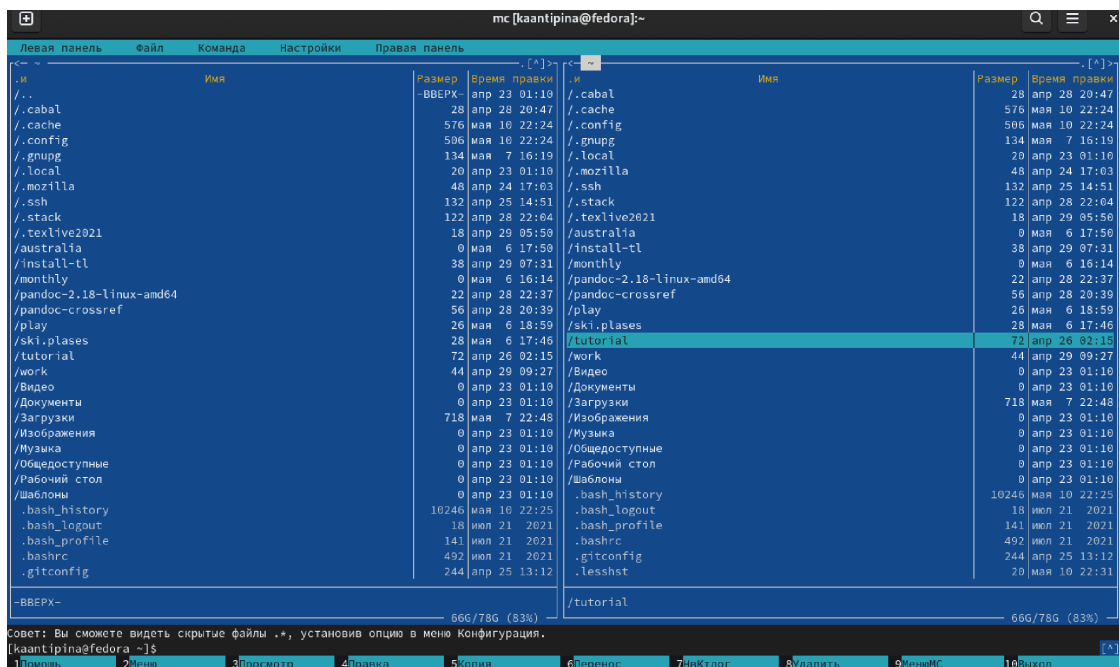
## Команда `man`

Запустим из командной строки `mc` и изучим его структуру и меню. (рис. -@fig:002)

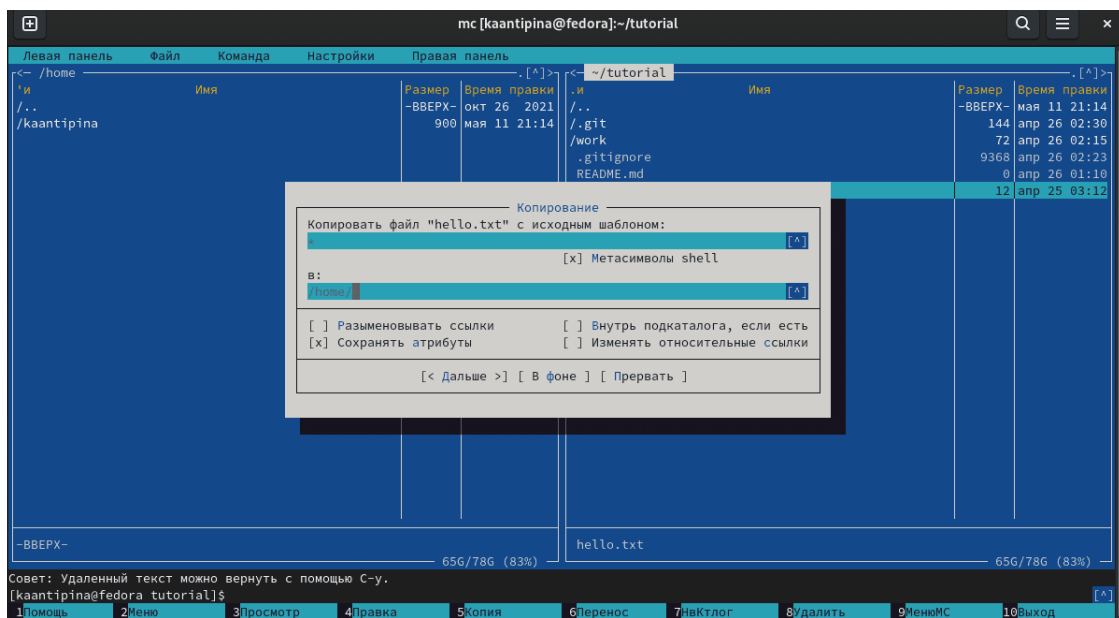


запустим *mc*

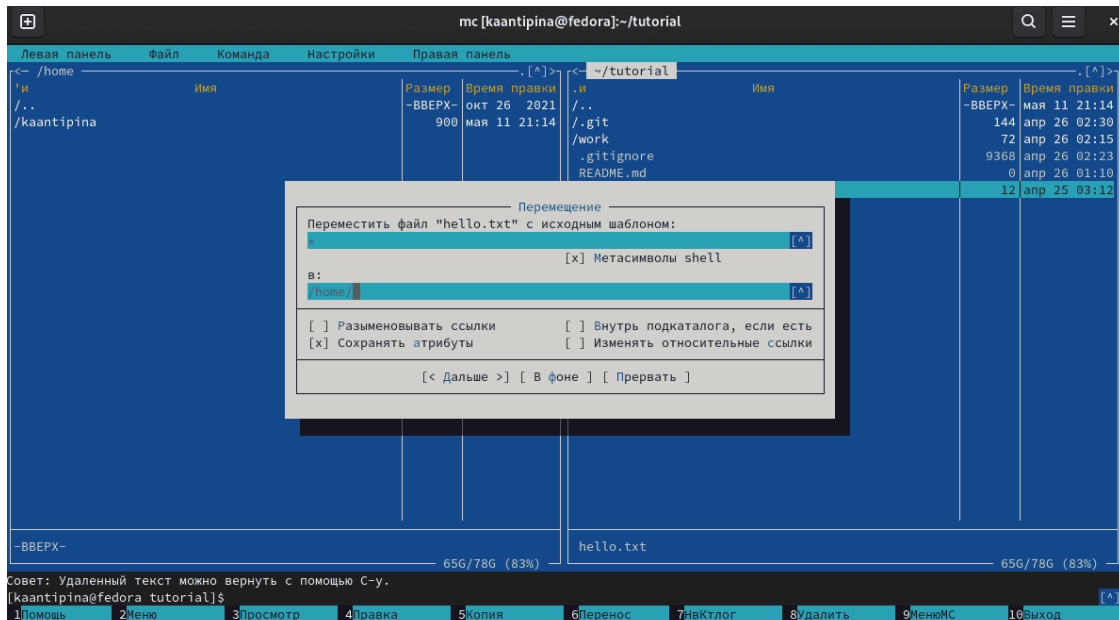
Выполним несколько операций в *mc*, используя управляющие клавиши (операции с панелями; выделение/отмена выделения файлов, копирование/перемещение файлов, получение информации о размере и правах доступа на файлы и/или каталоги и т.п.)(рис. -@fig:003)(рис. -@fig:004)(рис. -@fig:005) (рис. -@fig:006)



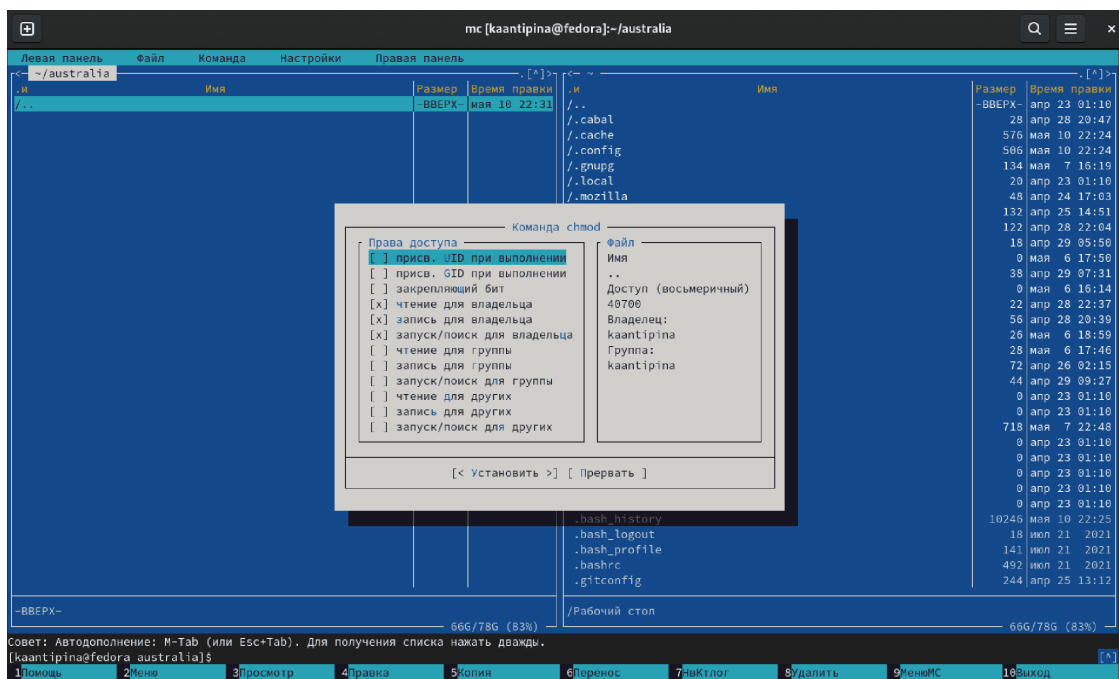
*mc*



операции в mc

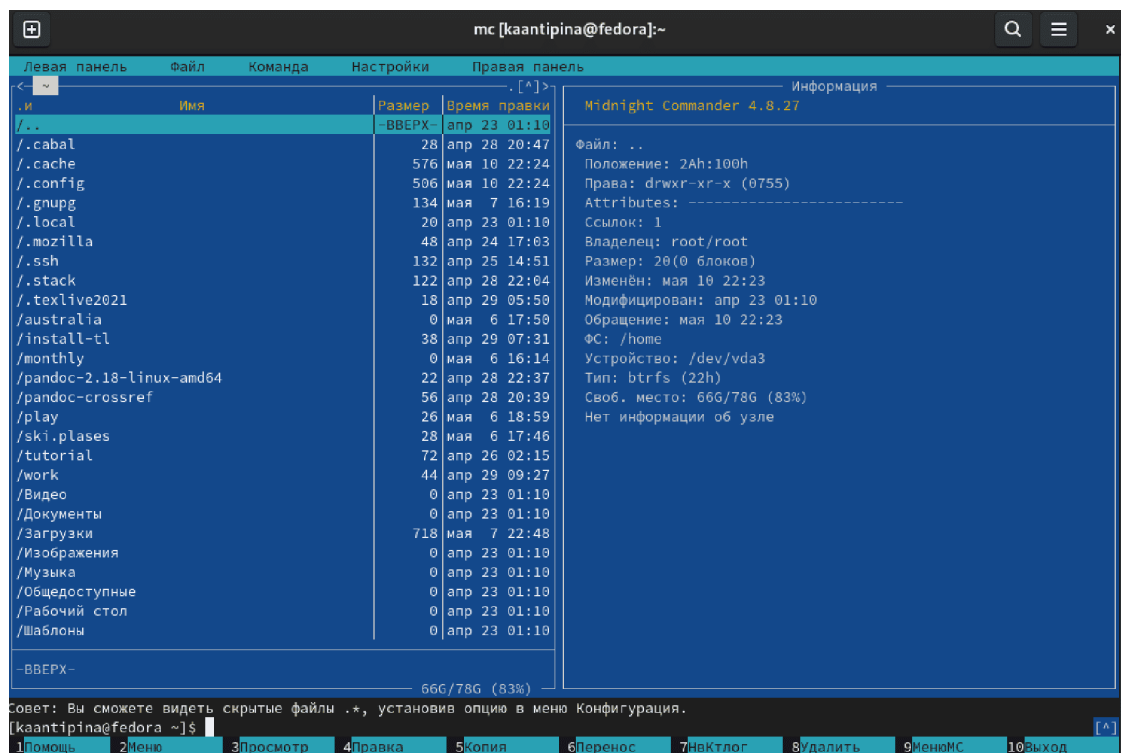


операции в mc



операции в mc

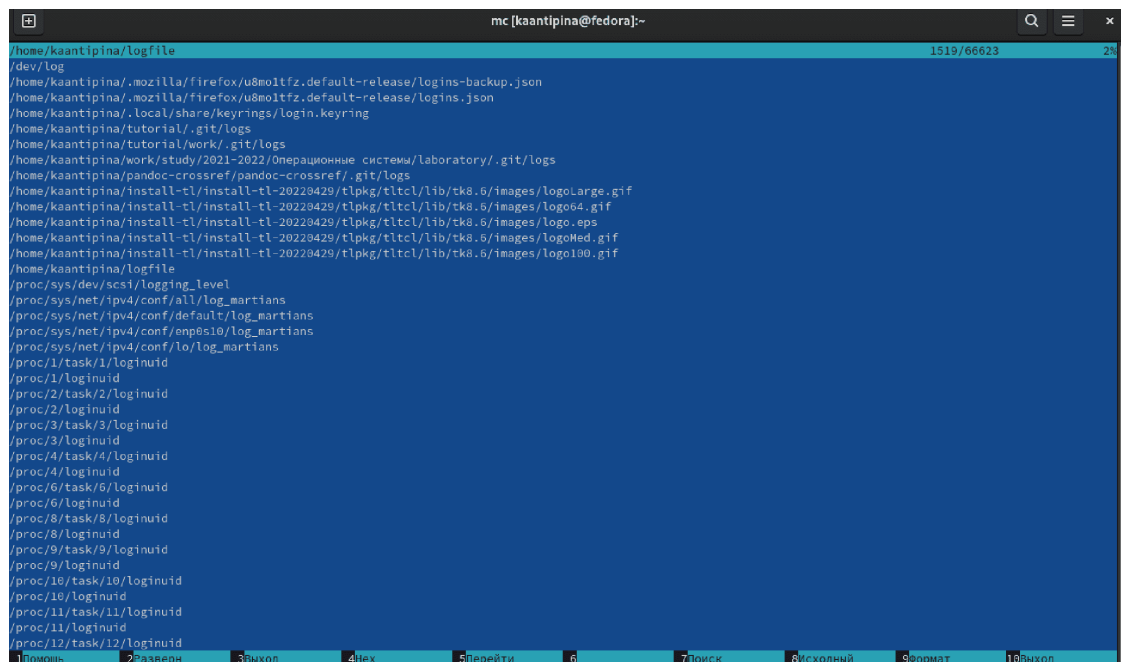
Выполним основные команды меню левой (или правой) панели. Вывод информации о файлах подробный.(рис. -@fig:007)



команды меню левой панели

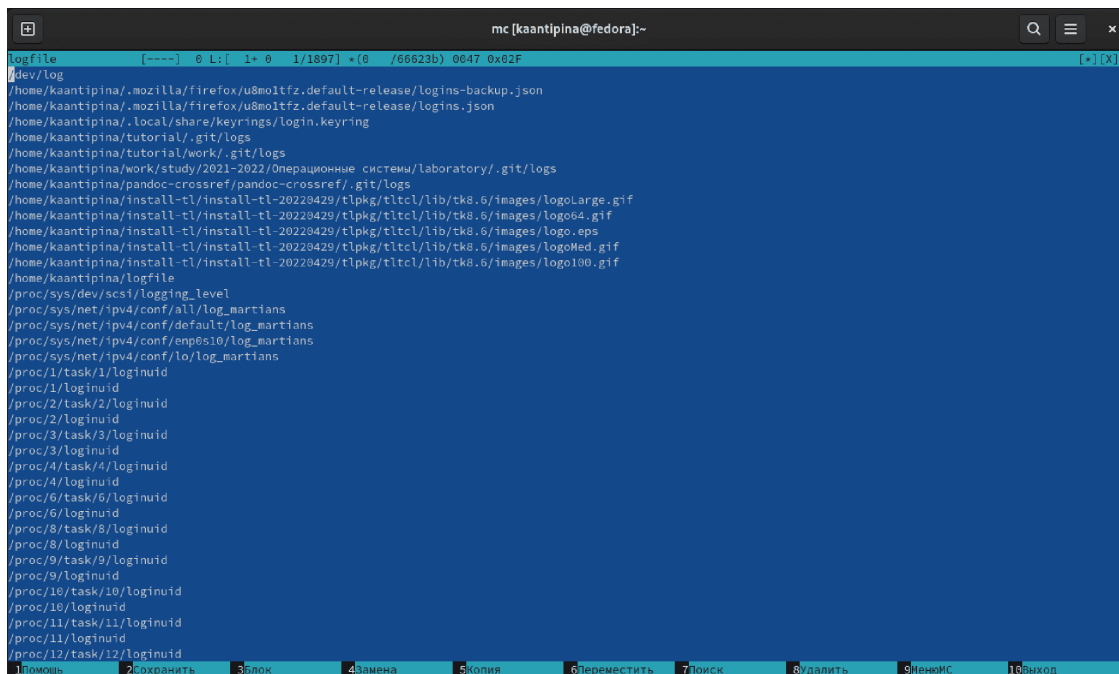
Используя возможности подменим Файл, выполним:

– просмотр содержимого текстового файла;(рис. -@fig:008)

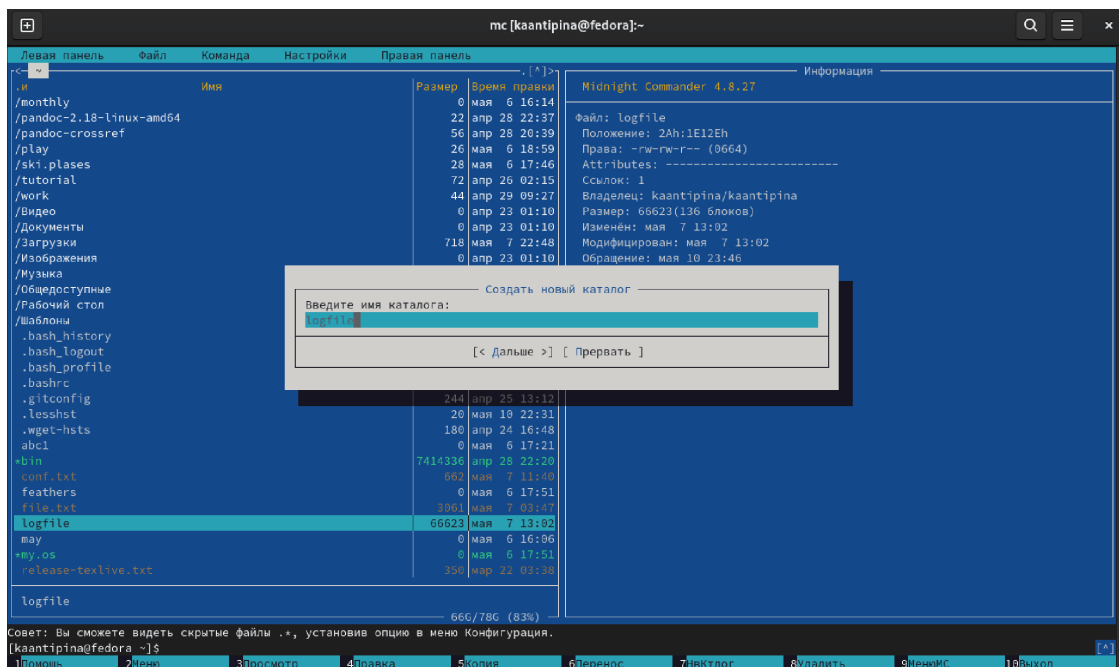


*просмотр содержимого*

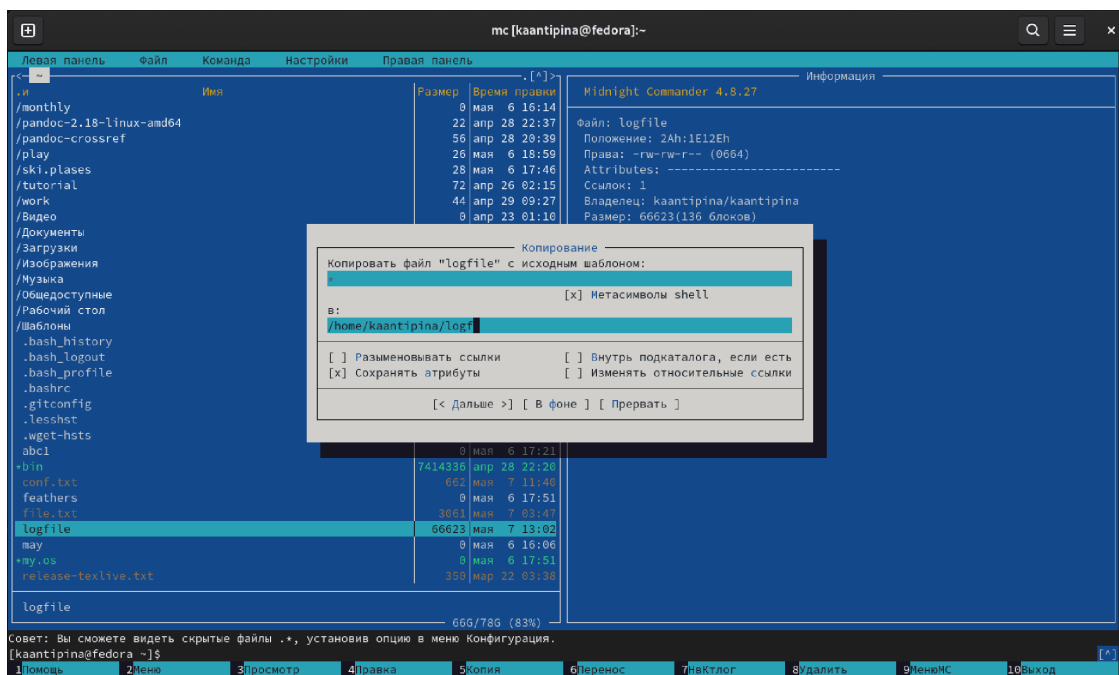
- редактирование содержимого текстового файла (без сохранения результатов редактирования);
- создание каталога;
- копирование файлов в созданный каталог.(рис. -@fig:009)(рис. -@fig:010)(рис. -@fig:011)



mc

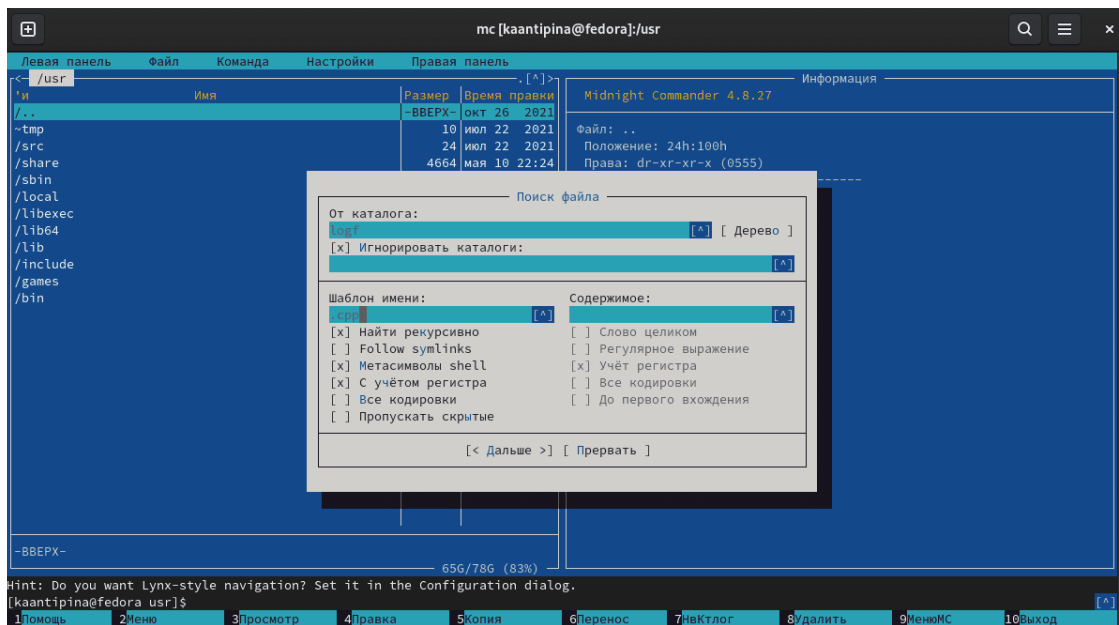


mc



*работа в mc*

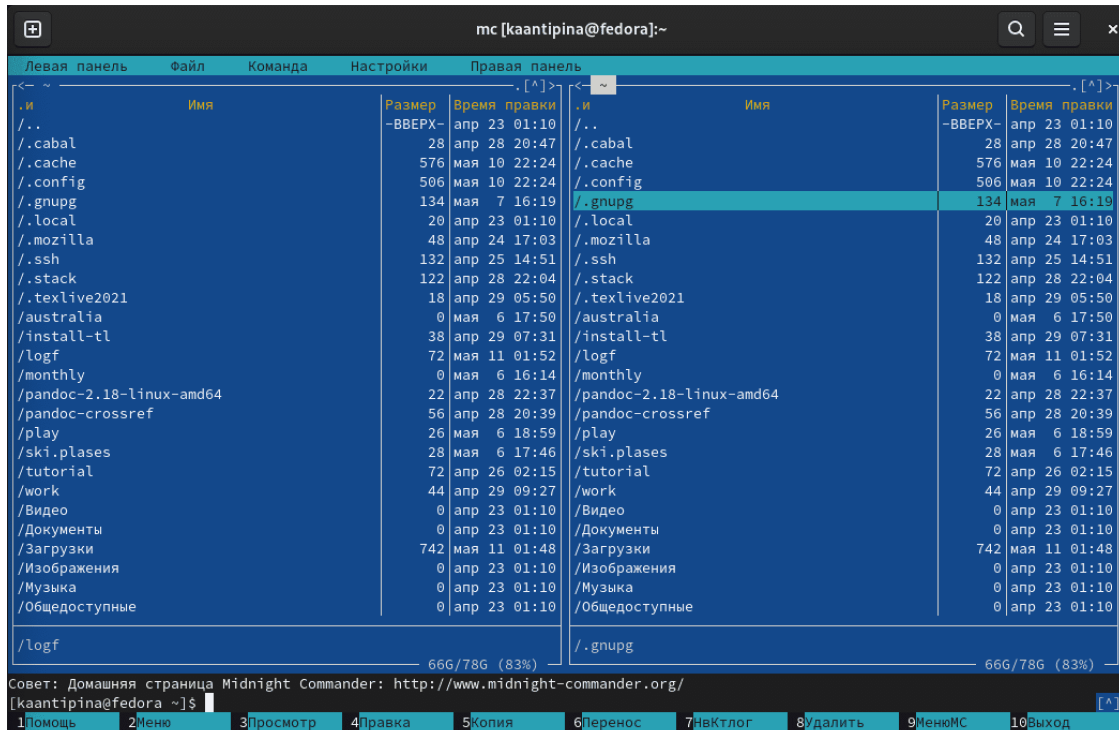
С помощью соответствующих средств подменим Команда осуществим: – поиск в файловой системе файла с заданными условиями;(рис. -@fig:012)



*поиск в файловой системе*



- выбор и повторение одной из предыдущих команд;
- переход в домашний каталог;
- анализ файла меню и файла расширений(рис. -@fig:013)(рис. -@fig:014)(рис. -@fig:015)



*работа в mc*

```
mc [kaantipina@fedora]:~
mc.ext [----] 0 L:[ 1+ 0 1/848] *(0 /22252b) 0035 0x023 [*][X]
# Midnight Commander 3.0 extension file
# Warning: Structure of this file has changed completely with version 3.0
#
# All lines starting with # or empty lines are thrown away.
# Lines starting in the first column should have following format:
#
# keyword/descNL, i.e. everything after keyword/ until new line is desc
#
# keyword can be:
#
# shell (desc is, when starting with a dot, any extension (no wildcars),
#       i.e. matches all the files +desc . Example: .tar matches *.tar;
#       if it doesn't start with a dot, it matches only a file of that name)
#
# shell/i (desc is, when starting with a dot, any extension (no wildcars),
#         The same as shell but with case insensitive.
#
# regex (desc is an extended regular expression)
#       Please note that we are using the GNU regex library and thus
#       \j matches the literal j and j has special meaning (or) and
#       () have special meaning and \( \) stand for literal ( ).
#
# regex/i (desc is an extended regular expression)
#         The same as regex but with case insensitive.
#
# type (file matches this if 'file %f' matches regular expression desc
#      (the filename: part from 'file %f' is removed))
#
# type/i (file matches this if 'file %f' matches regular expression desc)
#        The same as type but with case insensitive.
#
# directory (matches any directory matching regular expression desc)
#
# include (matches an include directive)
1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Переместить 7Поиск 8Удалить 9МенюМС 10Выход
```

*работа в mc*

```
mc [kaantipina@fedora]:~
.mc.menu [----] 0 L:[ 1+ 0 1/370] *(0 /11820b) 0115 0x073 [*][X]
shell_patterns=0

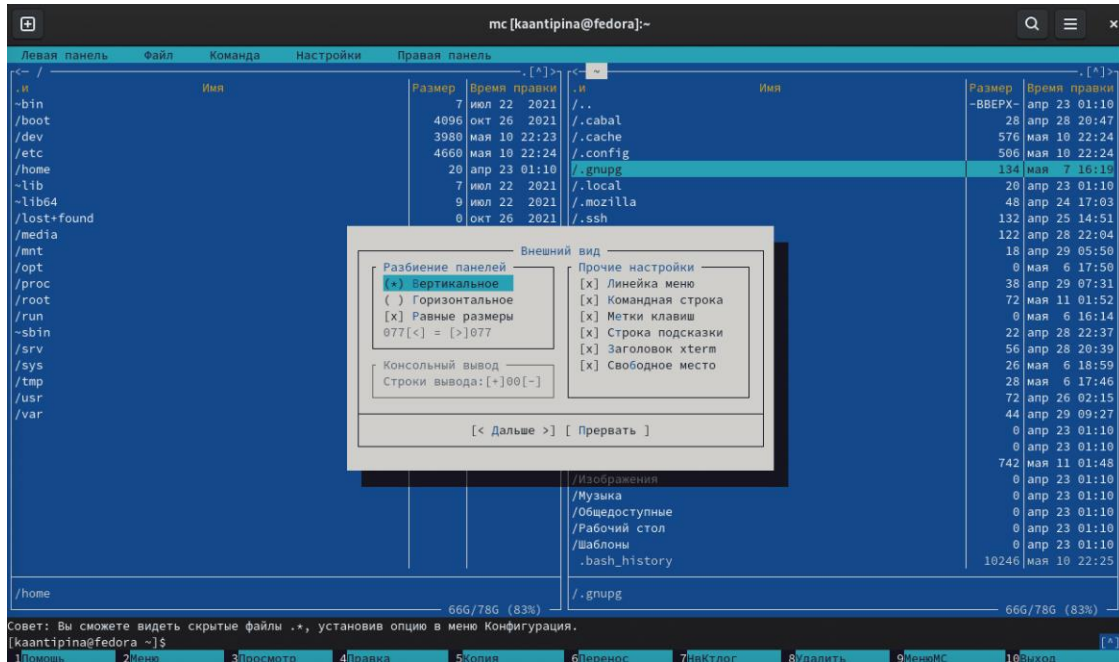
#####
# %% The % character
# %f The current file (if non-local vfs, file will be copied locally and
#   %f will be full path to it)
#
# %p The current file
# %d The current working directory
# %s "Selected files"; the tagged files if any, otherwise the current file
# %t Tagged files
# %u Tagged files (and they are untagged on return from expand_format)
# %view Runs the commands and pipes standard output to the view command
# If %view is immediately followed by '{', recognize keywords
#   ascii, hex, nroff and unform
#
# If the format letter is in uppercase, it refers to the other panel
#
# With a number followed the % character you can turn quoting on (default)
# and off. For example:
# %f quote expanded macro
# %if ditto
# %of don't quote expanded macro
#####

+ ! t t
@   Do something on the current file
    CMD=%(Enter command)
    $CMD %f

+ t t
@   Do something on the tagged files
    CMD=%(Enter command)
    for i in %t ; do
        $CMD "$i"
    done
```

*работа в mc*

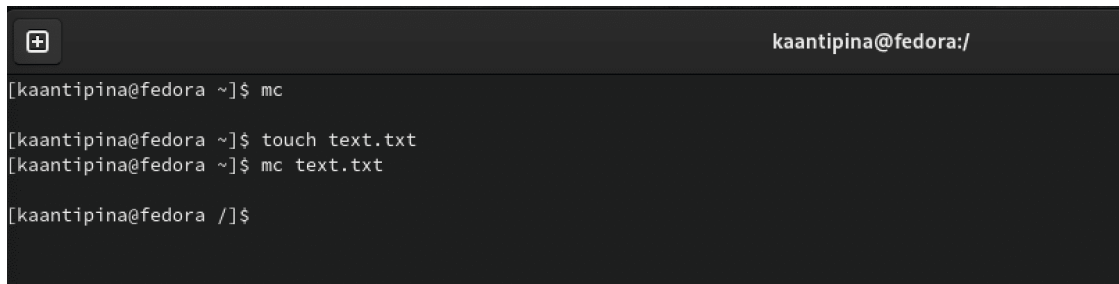
Вызываем подменю Настройки. Освоим операции, определяющие структуру экрана mc (Full screen, Double Width, Show Hidden Files и т.д.)(рис. -@fig:016)



вызываем подменю Настройки

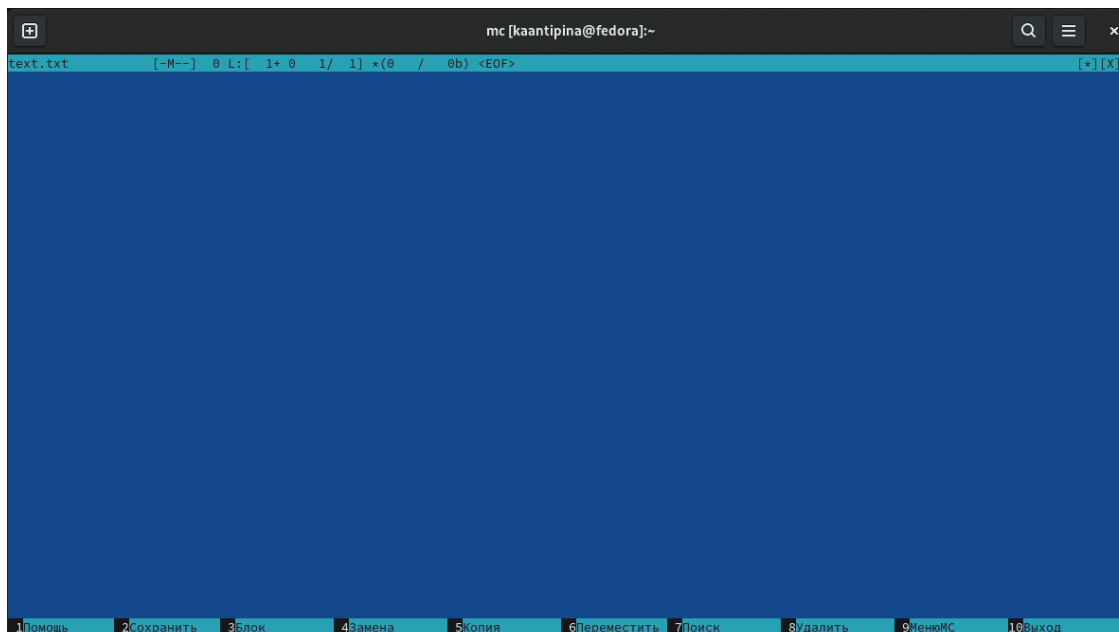
## Задание по встроенному редактору mc

Создадим текстовый файл text.txt(рис. -@fig:017)



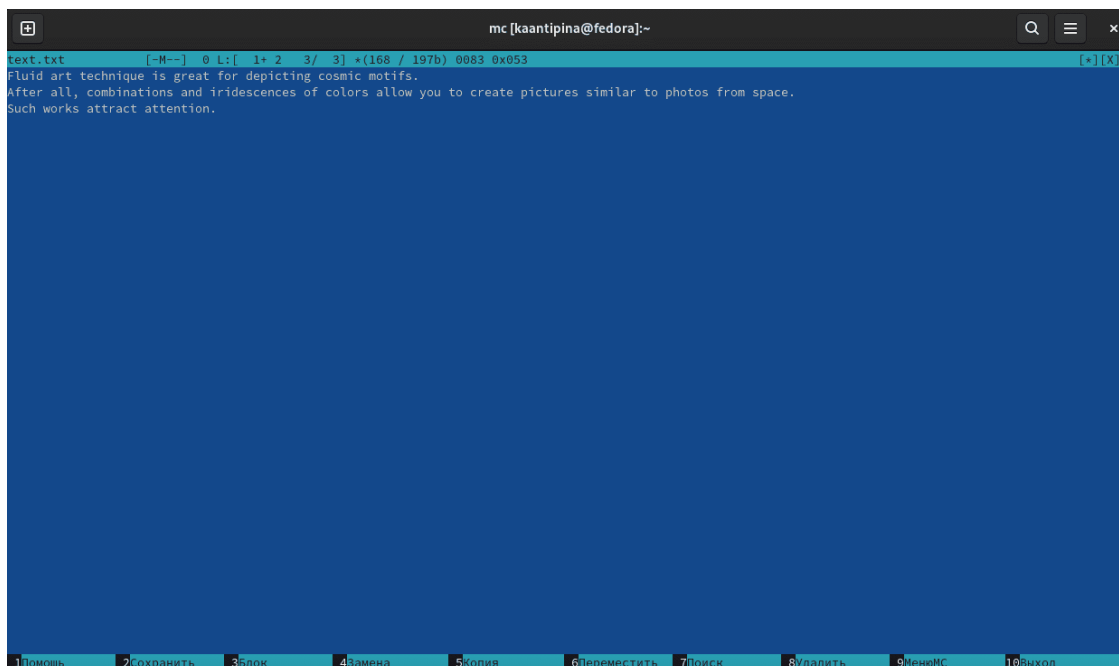
создаем файл

Открываем этот файл с помощью встроенного в mc редактора(рис. -@fig:018)



*открыли файл*

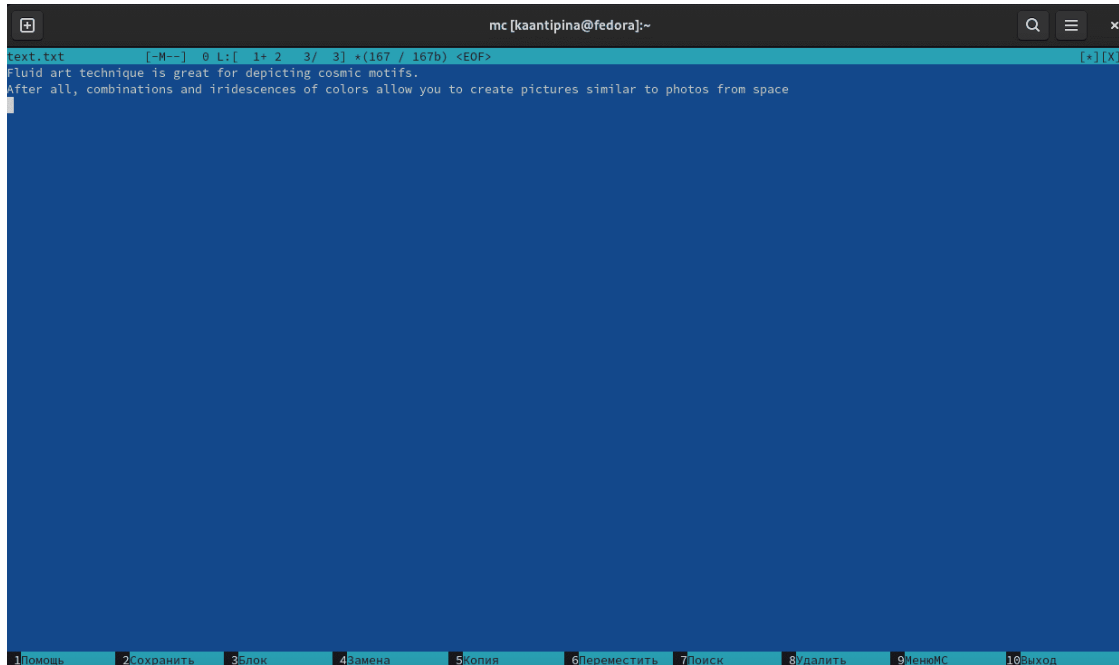
Вставляем в открытый файл небольшой текст, скопированный из любого источника.(рис. -@fig:019)



*вставляем в файл текст*

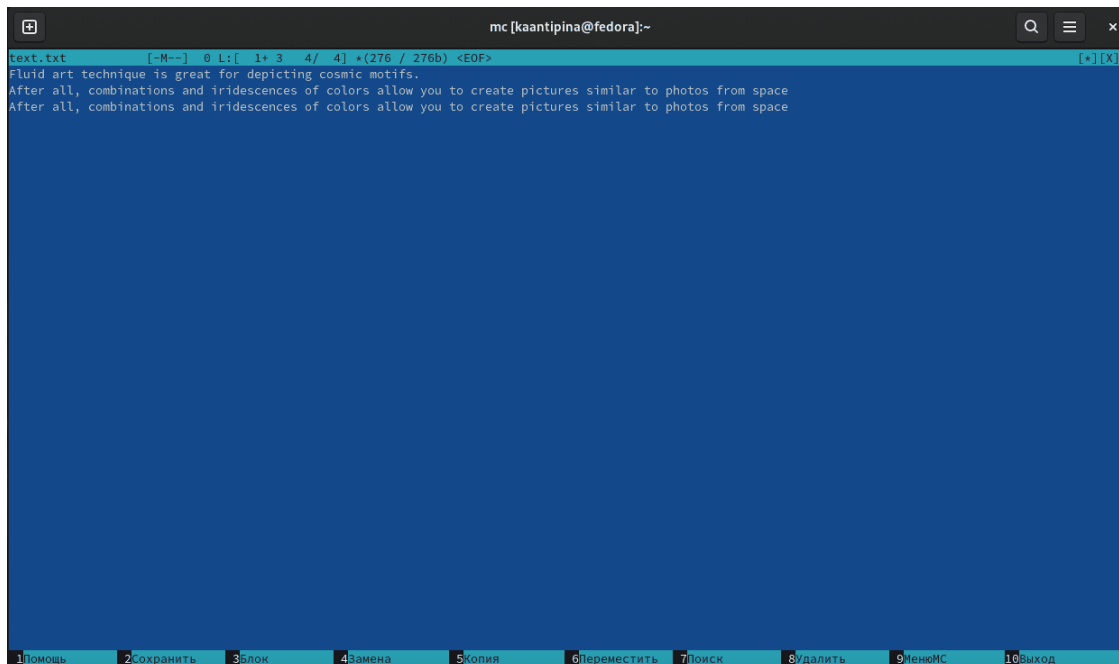
Проделаем с текстом следующие манипуляции, используя горячие клавиши:

Удалим строку текста. - F8 (рис. -@fig:020)



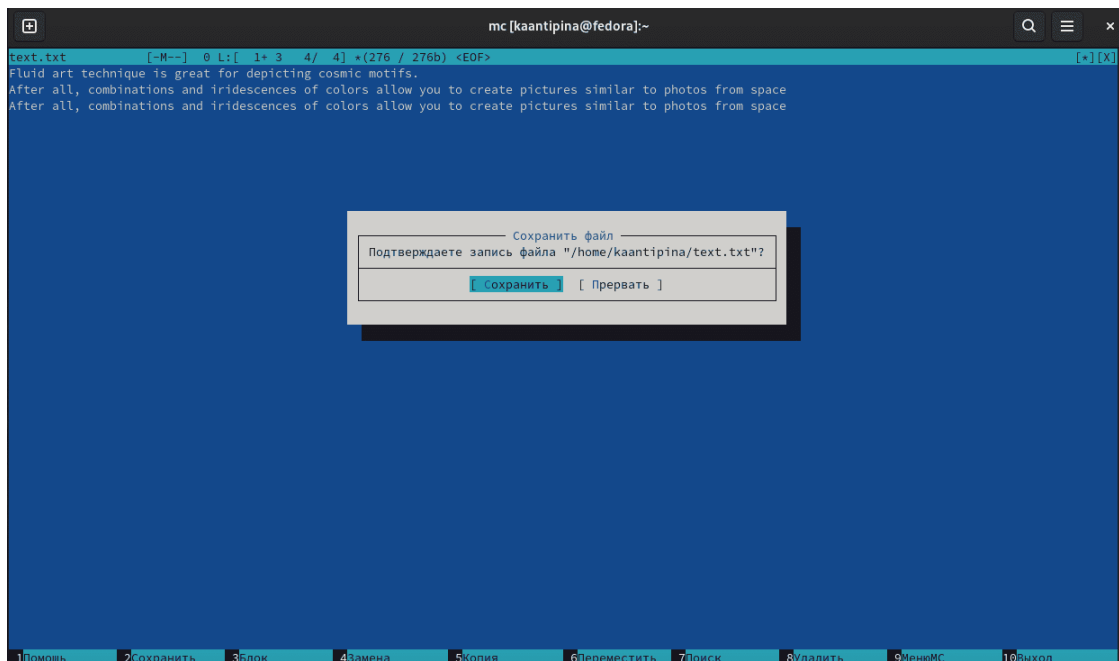
*удалим строку*

Выделим фрагмент текста и скопируем его на новую строку. - F5 (рис. -@fig:021)



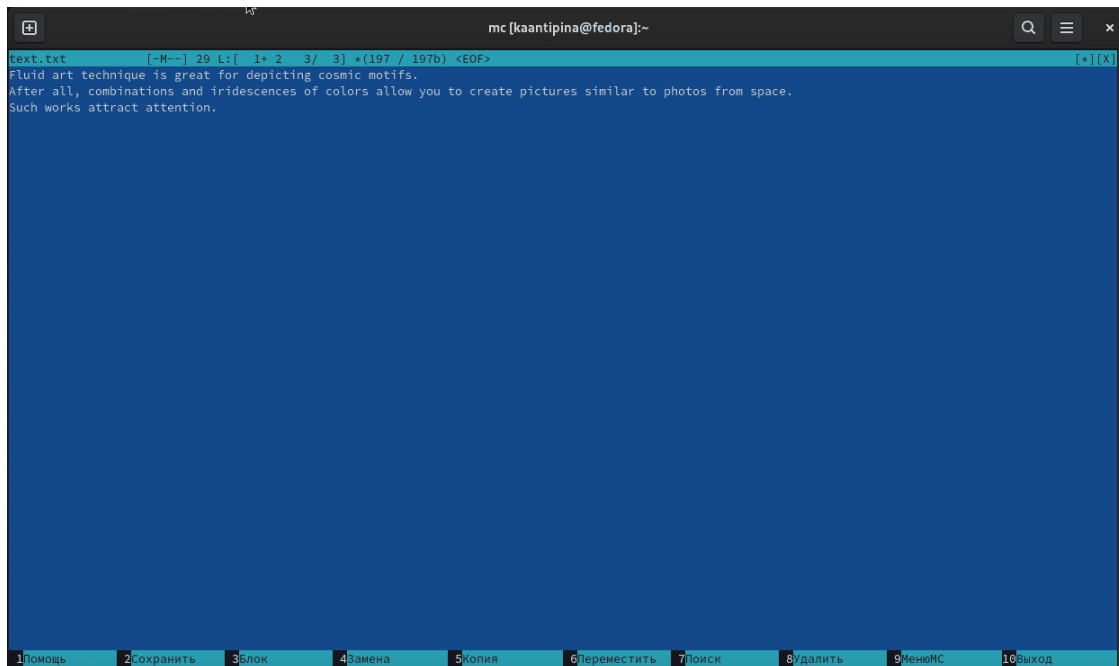
*Копирование фрагмента*

Сохраним файл. - F2 (рис. -@fig:022)



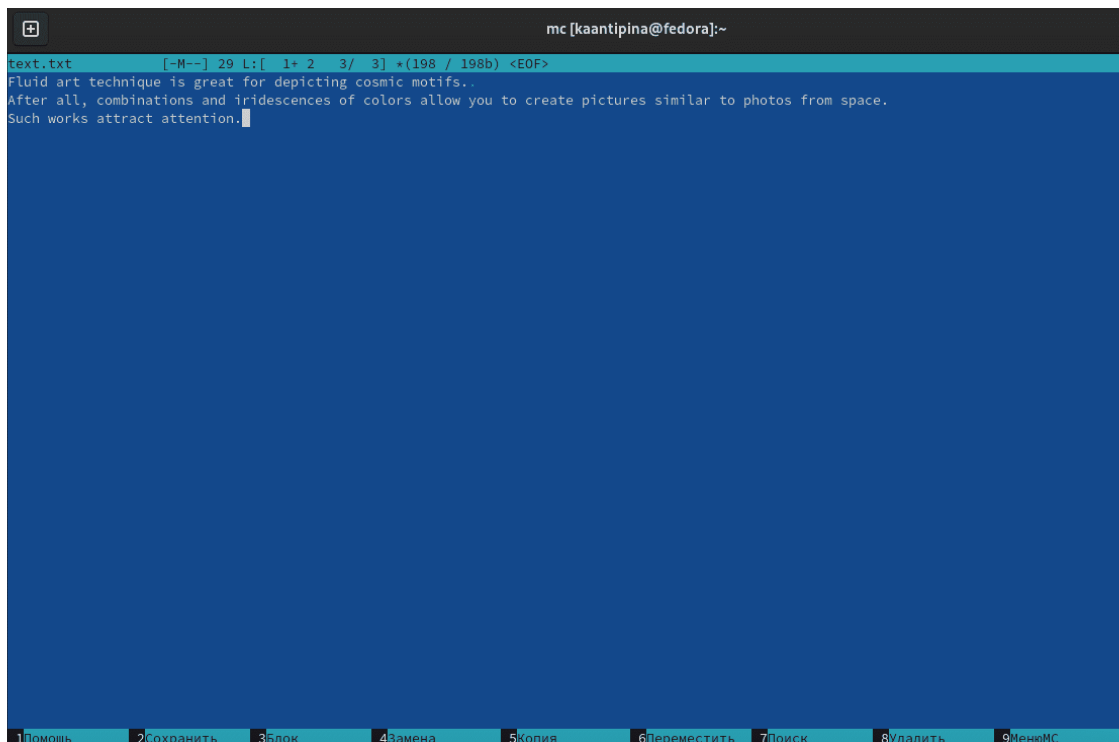
*Сохранение*

Отменим последнее действие. - Ctrl+U (рис. -@fig:023)



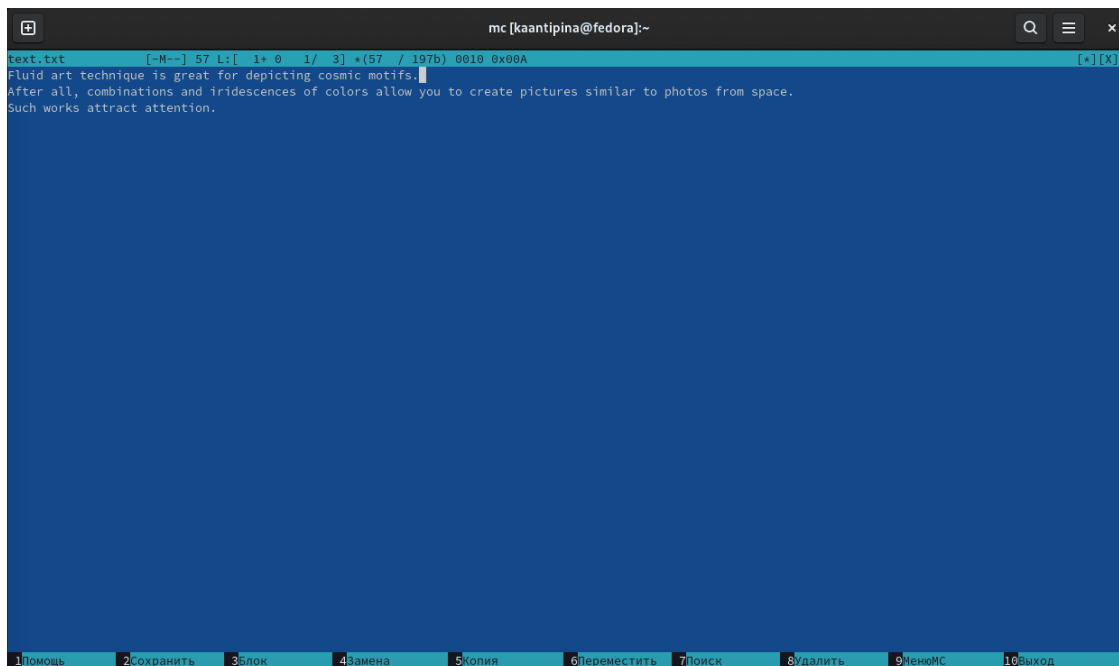
*Отмена*

Перейдем в конец файла - PageDown или Ctrl+X (рис. -@fig:024)



*Переход в конец файла*

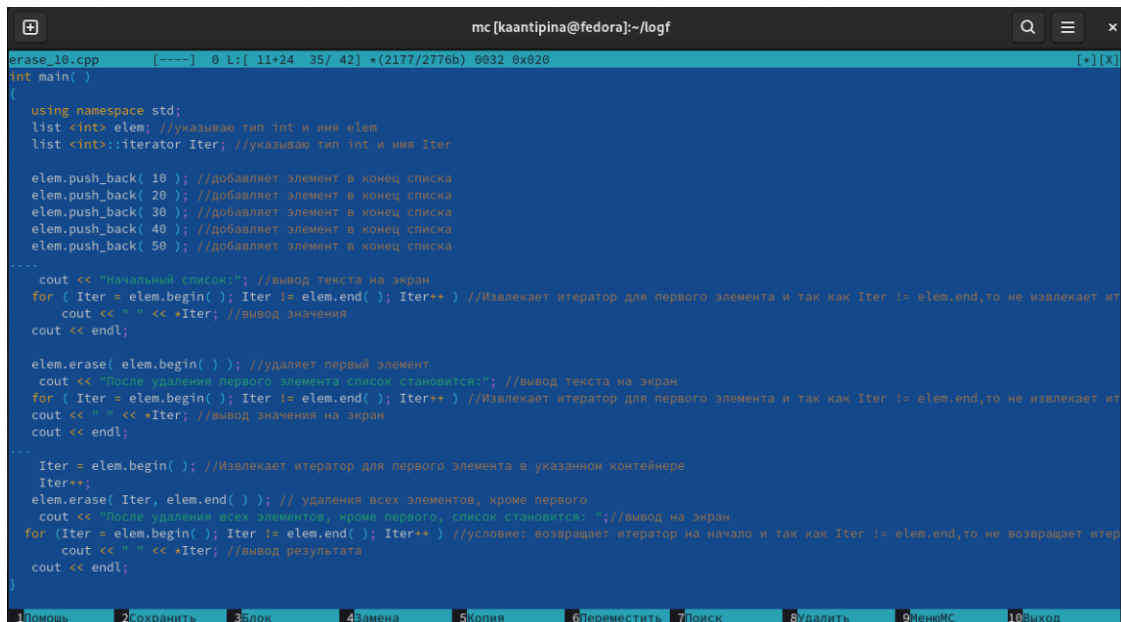
Перейдем в начало файла - PageUp или Ctrl+Z (рис. -@fig:025)



*Переход в начало файла*



Откроем файл с исходным текстом на языке программирования C++ (рис. -@fig:026)



```
erase_10.cpp [----] 0 L: [ 11+24 35/ 42] *(2177/2776b) 0032 0x020
int main( )
{
    using namespace std;
    list<int> elem; //указываю тип int и имя elem
    list<int>::iterator Iter; //указываю тип int и имя Iter

    elem.push_back( 10 ); //добавляет элемент в конец списка
    elem.push_back( 20 ); //добавляет элемент в конец списка
    elem.push_back( 30 ); //добавляет элемент в конец списка
    elem.push_back( 40 ); //добавляет элемент в конец списка
    elem.push_back( 50 ); //добавляет элемент в конец списка

    ....
    cout << "Начальный список!"; //вывод текста на экран
    for ( Iter = elem.begin( ); Iter != elem.end( ); Iter++ ) //Извлекает итератор для первого элемента и так как Iter != elem.end, то не извлекает итератор
        cout << " " << *Iter; //вывод значения
    cout << endl;

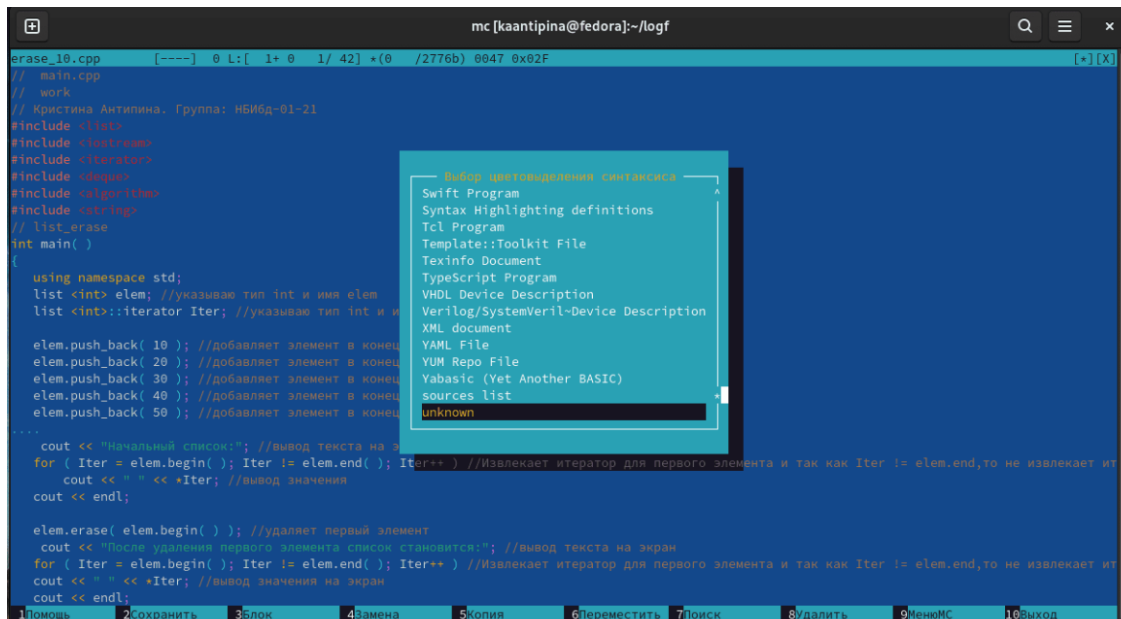
    elem.erase( elem.begin( ) ); //удаляет первый элемент
    cout << "После удаления первого элемента список становится:"; //вывод текста на экран
    for ( Iter = elem.begin( ); Iter != elem.end( ); Iter++ ) //Извлекает итератор для первого элемента и так как Iter != elem.end, то не извлекает итератор
        cout << " " << *Iter; //вывод значения на экран
    cout << endl;

    ....
    Iter = elem.begin( ); //Извлекает итератор для первого элемента в указанном контейнере
    Iter++;
    elem.erase( Iter, elem.end( ) ); // удаление всех элементов, кроме первого
    cout << "После удаления всех элементов, кроме первого, список становится:"; //вывод на экран
    for ( Iter = elem.begin( ); Iter != elem.end( ); Iter++ ) //условие: возвращает итератор на начало и так как Iter != elem.end, то не возвращает итератор
        cout << " " << *Iter; //вывод результата
    cout << endl;
}
```

1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Переместить 7Поиск 8Удалить 9МенюМС 10Выход

Файл с программой

Используя меню редактора, выключим подсветку синтаксиса. (рис. -@fig:027)



```
erase_10.cpp [----] 0 L: [ 1+ 0 1/ 42] *(0 /2776b) 0047 0x02F
// main.cpp
// work
// Кристина Антипина. Группа: НБИИд-01-21
#include <iostream>
#include <iostream>
#include <iostream>
#include <iostream>
#include <iostream>
#include <iostream>
#include <iostream>
// list_erase
int main( )
{
    using namespace std;
    list<int> elem; //указываю тип int и имя elem
    list<int>::iterator Iter; //указываю тип int и имя Iter

    elem.push_back( 10 ); //добавляет элемент в конец списка
    elem.push_back( 20 ); //добавляет элемент в конец списка
    elem.push_back( 30 ); //добавляет элемент в конец списка
    elem.push_back( 40 ); //добавляет элемент в конец списка
    elem.push_back( 50 ); //добавляет элемент в конец списка

    ....
    cout << "Начальный список!"; //вывод текста на экран
    for ( Iter = elem.begin( ); Iter != elem.end( ); Iter++ ) //Извлекает итератор для первого элемента и так как Iter != elem.end, то не извлекает итератор
        cout << " " << *Iter; //вывод значения
    cout << endl;

    elem.erase( elem.begin( ) ); //удаляет первый элемент
    cout << "После удаления первого элемента список становится:"; //вывод текста на экран
    for ( Iter = elem.begin( ); Iter != elem.end( ); Iter++ ) //Извлекает итератор для первого элемента и так как Iter != elem.end, то не извлекает итератор
        cout << " " << *Iter; //вывод значения на экран
    cout << endl;
}
```

1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Переместить 7Поиск 8Удалить 9МенюМС 10Выход

Выбор цветовой подсветки синтаксиса

- Swift Program
- Syntax Highlighting definitions
- Tcl Program
- Template::Toolkit File
- Texinfo Document
- TypeScript Program
- VHDL Device Description
- Verilog/SystemVerilog-Device Description
- XML document
- YAML File
- YUM Repo File
- Yabasic (Yet Another BASIC)
- sources list
- unknown

Цветовыделение синтаксиса

## Вывод

В данной работе я ознакомилась с инструментами командной оболочки Midnight Commander. Приобрела навыки практической работы по просмотру каталогов и файлов.

—