

# **Лабораторная работа №7.**

**Командная оболочка Midnight Commander.**

**Александр Андреевич Шуплецов**

# **Содержание**

<b>1 Цель работы</b>	<b>5</b>
<b>2 Теоретическое введение</b>	<b>6</b>
<b>3 Выполнение работы</b>	<b>7</b>
<b>4 Выводы</b>	<b>18</b>
<b>Список литературы</b>	<b>19</b>

# Список иллюстраций

3.1	man mc . . . . .	7
3.2	запуск mc . . . . .	8
3.3	просмотр содержимого текстового файла . . . . .	9
3.4	редактирование содержимого текстового файла . . . . .	10
3.5	создание каталога . . . . .	10
3.6	копирование файлов в созданный каталог . . . . .	11
3.7	поиск в файловой системе . . . . .	11
3.8	выбор и повторение предыдущей команды . . . . .	11
3.9	анализ файла меню . . . . .	12
3.10	анали файл расширений . . . . .	13
3.11	вызов подменю Настройки . . . . .	14
3.12	создание текстового файла text.txt . . . . .	14
3.13	открытие текстового файла text.txt . . . . .	15
3.14	сохранение изменений в файле text.txt . . . . .	15
3.15	открытие файла cpp . . . . .	16
3.16	выключение подсветки синтаксиса в файле cpp . . . . .	17

# **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 Теоретическое введение**

Панель в тс отображает список файлов текущего каталога. Абсолютный путь к этому каталогу отображается в заголовке панели. У активной панели заголовок и одна из её строк подсвечиваются. Управление панелями осуществляется с помощью определённых комбинаций клавиш или пунктов меню тс. Панели можно поменять местами. Для этого и используется комбинация клавиш Ctrl-и или команда меню тс Переставить панели . Также можно временно убрать отображение панелей (отключить их) с помощью комбинации клавиш Ctrl-о или команды меню тс Отключить панели . Это может быть полезно, например, если необходимо увидеть вывод какой-то информации на экран после выполнения какой-либо команды shell. С помощью последовательного применения комбинации клавиш Ctrl-x d есть возможность сравнения каталогов, отображённых на двух панелях. Панели могут дополнительно быть переведены в один из двух режимов: Информация или Дерево. В режиме Информация на панель выводятся сведения о файле и текущей файловой системе, расположенных на активной панели. В режиме Дерево на одной из панелей выводится структура дерева каталогов. Управлять режимами отображения панелей можно через пункты меню тс Правая панель и Левая панель.

# 3 Выполнение работы

1. Изучим информацию о mc, вызвав в командной строке man mc.

The screenshot shows a terminal window with the command `man mc` entered. The title bar indicates the session is `aashuplecov@aashuplecov:~ — man mc`. The man page for `mc(1)` is displayed, titled "GNU Midnight Commander". The page is divided into sections: **НАИМЕНОВАНИЕ**, **СИНТАКСИС**, **ОПИСАНИЕ**, and **КЛЮЧИ**. The **ОПИСАНИЕ** section provides a brief overview of what `Midnight Commander` is and its main functions. The **КЛЮЧИ** section lists various command-line options with their descriptions. At the bottom of the page, there is a footer: `Manual page mc(1) line 1 (press h for help or q to quit)`.

Рис. 3.1: man mc

2. Запустим из командной строки mc, изучим его структуру и меню.

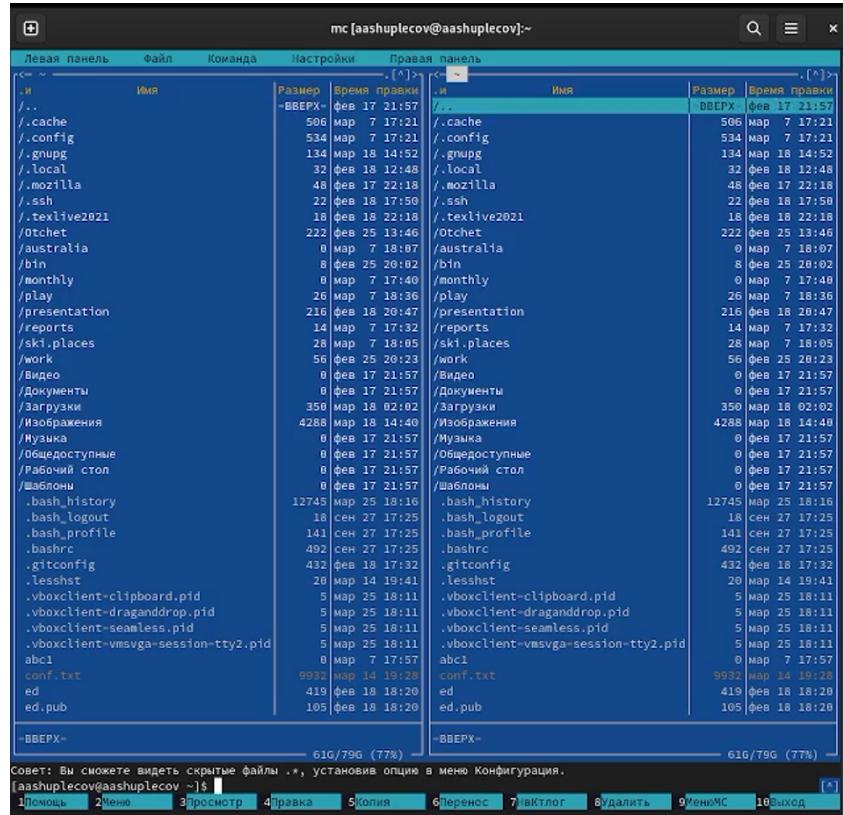


Рис. 3.2: запуск mc

### 3. Используя возможности подменю Файл , выполним:

- просмотр содержимого текстового файла

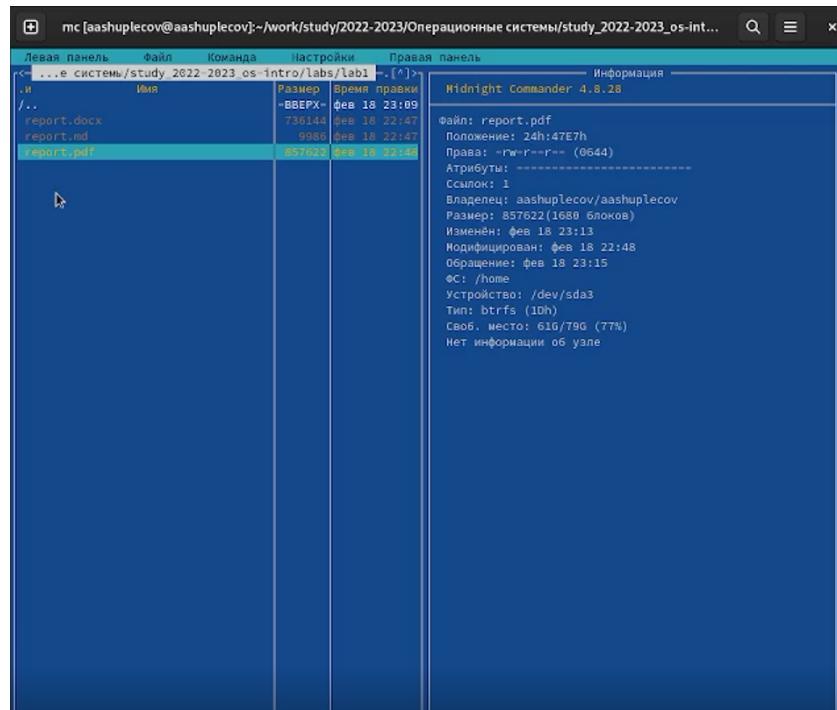


Рис. 3.3: просмотр содержимого текстового файла

– редактирование содержимого текстового файла (без сохранения результатов редактирования)

```
mc [aashuplecov@aashuplecov]:~/ski.places/equipment
equiplist2      [ 1+ 0 1/ 1] * (50 / 50b) <EOF>
Лабораторная работа номер 7
```

Рис. 3.4: редактирование содержимого текстового файла

- создание каталога

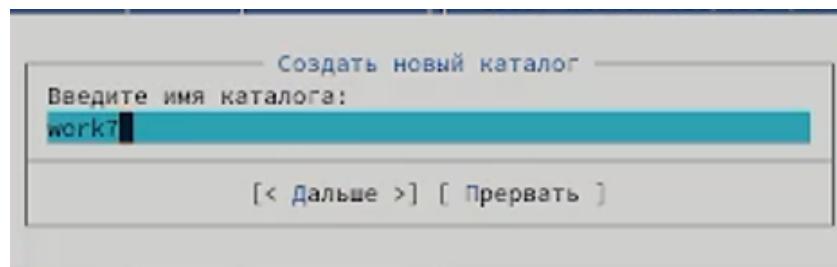


Рис. 3.5: создание каталога

- копирование файлов в созданный каталог

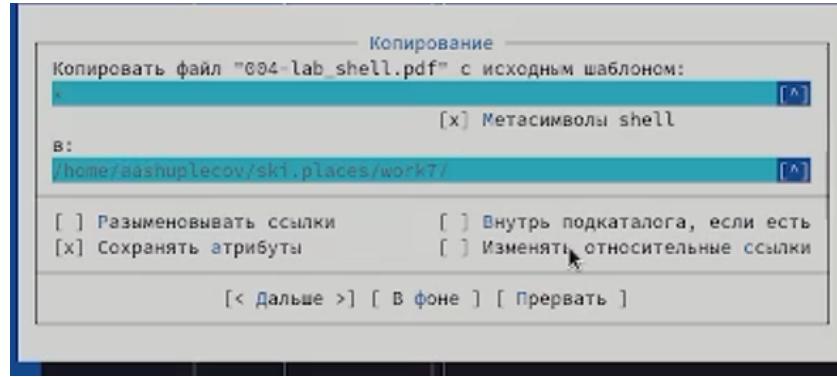


Рис. 3.6: копирование файлов в созданный каталог

4. С помощью соответствующих средств подменю Команда осуществим:

- поиск в файловой системе файла с заданными условиями

```
[aashuplecov@aashuplecov ~]$ mc
$ find -name '*.png' -print
./nitroshare=0.3.4/src/dist/icons/ubuntu-mono-light/apps/22/nitroshare-indicator.png
./nitroshare=0.3.4/src/dist/icons/ubuntu-mono-dark/apps/22/nitroshare-indicator.png
./nitroshare=0.3.4/src/dist/icons/gnome/24x24/apps/nitroshare-indicator.png
```

Рис. 3.7: поиск в файловой системе

- выбор и повторение предыдущей команды

```
[aashuplecov@aashuplecov ~]$ mc
$ find -name '*.png' -print
./nitroshare=0.3.4/src/dist/icons/ubuntu-mono-light/apps/22/nitroshare-indicator.png
./nitroshare=0.3.4/src/dist/icons/ubuntu-mono-dark/apps/22/nitroshare-indicator.png
./nitroshare=0.3.4/src/dist/icons/gnome/24x24/apps/nitroshare-indicator.png
$ find -name '*.png' -print
./nitroshare=0.3.4/src/dist/icons/ubuntu-mono-light/apps/22/nitroshare-indicator.png
./nitroshare=0.3.4/src/dist/icons/ubuntu-mono-dark/apps/22/nitroshare-indicator.png
./nitroshare=0.3.4/src/dist/icons/gnome/24x24/apps/nitroshare-indicator.png
```

Рис. 3.8: выбор и повторение предыдущей команды

- анализ файла меню и файла расширений

The screenshot shows a terminal window titled 'mc [aashuplecov@aashuplecov]~'. The window displays the contents of the file '.mc.menu'. The file contains various shell patterns and their descriptions. A cursor is visible near the end of the file, specifically over the command 'rm -f "\$I"'.

```
.mc.menu      [----] 48 L:[ 1+17 18/370 ] x(781 /11820b) 0x97 0x061
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
#   %!f   ditto
#   %!of  don't quote expanded macro
#####

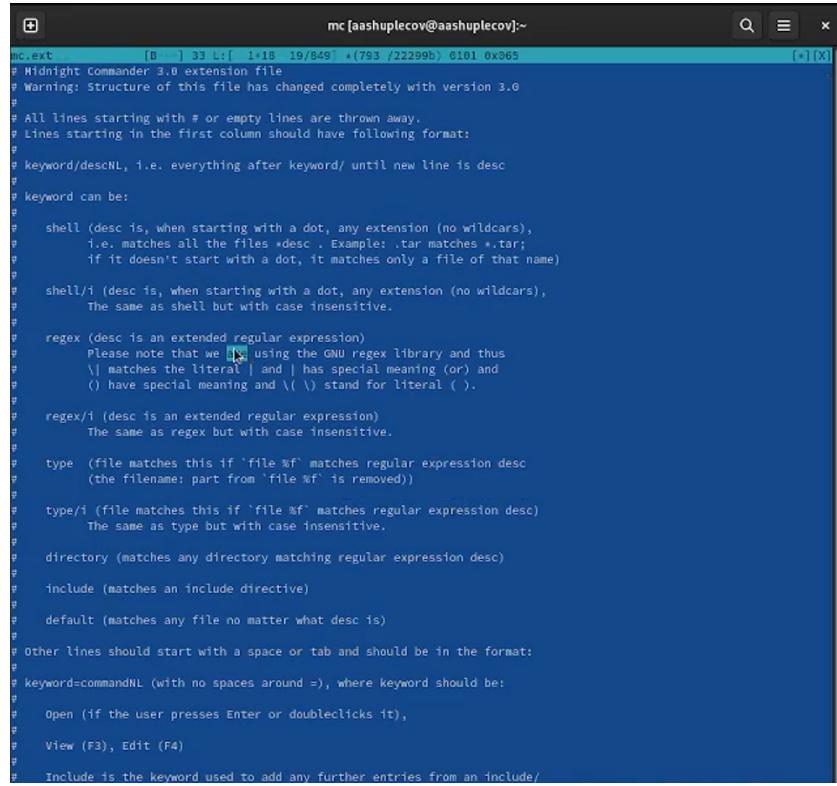
+ ! t t
@    Do something on the current file
    CMD=%{Enter command}
    $CMD %f

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

@    Edit a bug report and send it to root
I=mkttemp "$(MC_TMPDIR:=~/tmp)/mail.XXXXXX" || exit 1
${EDITOR:-vi} "$I"
test -r "$I" && mail root < "$I"
rm -f "$I"

+= f \.1$ | f \.3$ | f \.4$ | f \.5$ | f \.6$ | f \.7$ | f \.8$ | f \.man$ & t r
1    Display the file with roff -man
    %view[ascii,nroff] roff -c -Tlatin1 -mandoc %f
```

Рис. 3.9: анализ файла меню



The screenshot shows a terminal window titled 'mc [aashuplecov@aashuplecov]:' with the command 'mc .ext' entered. The window displays the contents of the 'mc .ext' file, which is a configuration file for the Midnight Commander (mc) application. The file contains various comments explaining the syntax for defining file types and descriptions. Key sections include:

- # 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 wildcards), 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 wildcards), 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 \| matches the literal | and | 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)
  - # default (matches any file no matter what desc is)

- # Other lines should start with a space or tab and should be in the format:
- # keyword=commandNL (with no spaces around =), where keyword should be:
- # Open (if the user presses Enter or doubleclicks it),
- # View (F3), Edit (F4)
- # Include is the keyword used to add any further entries from an include/

Рис. 3.10: анализ файла расширений

5. Вызовем подменю Настройки. Освоим операции, определяющие структуру экрана mc (Full screen, Double Width, Show Hidden Files и т.д.).

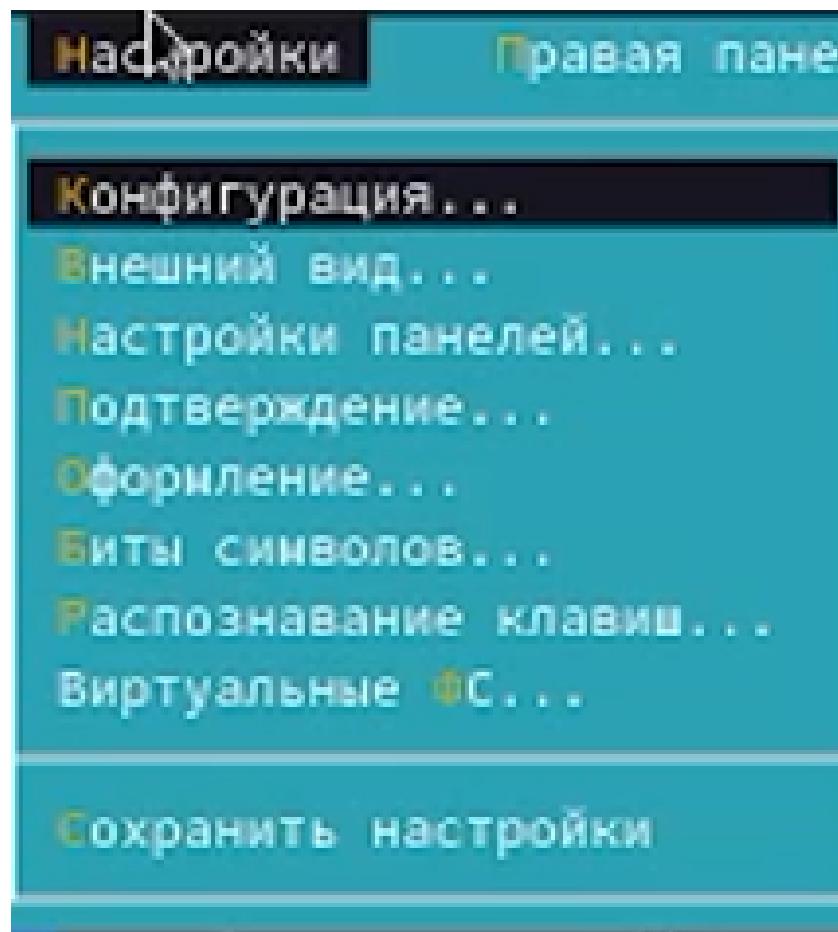


Рис. 3.11: вызов подменю Настройки

6. Создадим текстовой файл text.txt

Имя	Размер	Время правки
-ВВЕРХ-	5086	мар 25 18:38
equiplist	0	мар 7 17:51
equiplist2	0	мар 7 17:57
text.txt	0	мар 25 19:05

Рис. 3.12: создание текстового файла text.txt

7. Откроем этот файл с помощью встроенного в mc редактора, вставим в открытый файл небольшой фрагмент текста, скопированный из Интернета.

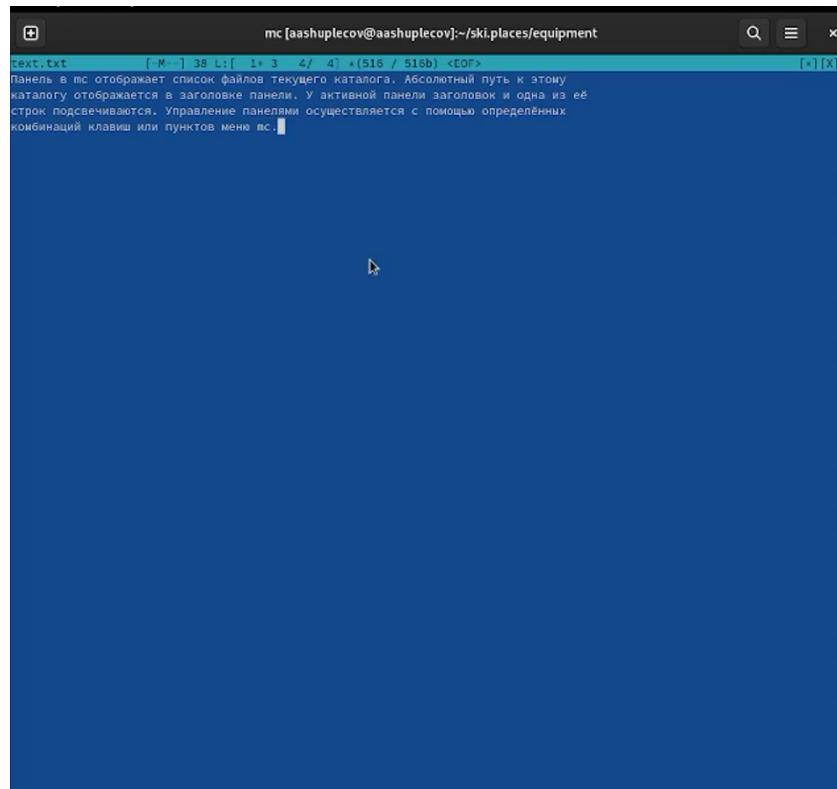


Рис. 3.13: открытие текстового файла text.txt

8. Поработаем с текстовым файлом text.txt и сохраним изменения в нем.

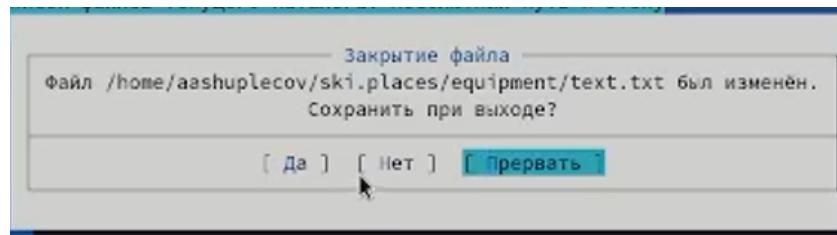
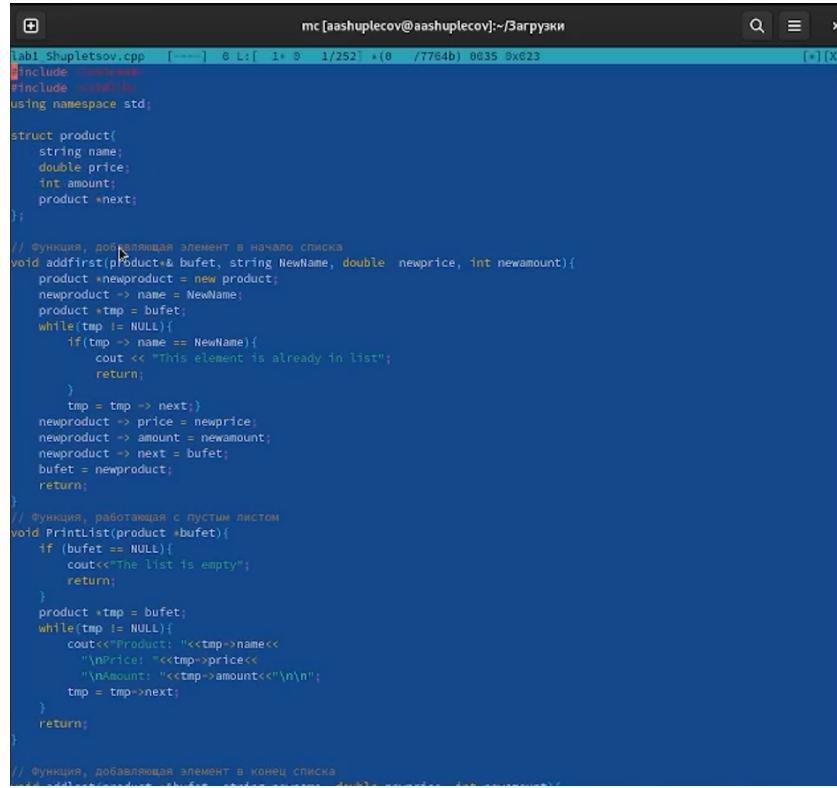


Рис. 3.14: сохранение изменений в файле text.txt

9. Откроем файл с исходным текстом на языке программирования C++.



```
labb1 Shupletsov.cpp [----] 0 L: [ 1< 0 1/252 ] * (0 /7764b) 0035 0x023
mc [aashuplecov@aashuplecov:~/Загрузки] [x] [x]

#include <iostream>
#include <vector>
using namespace std;

struct product{
    string name;
    double price;
    int amount;
    product *next;
};

// Функция, добавляющая элемент в начало списка
void addfirst(product*& bufet, string NewName, double newprice, int newamount){
    product *newproduct = new product;
    newproduct->name = NewName;
    newproduct->price = newprice;
    newproduct->amount = newamount;
    newproduct->next = bufet;
    bufet = newproduct;
    return;
}

// Функция, работавшая с пустым листом
void PrintList(product *bufet){
    if (bufet == NULL){
        cout<<"The list is empty";
        return;
    }
    product *tmp = bufet;
    while(tmp != NULL){
        cout<<"Product: "<<tmp->name<<
            "\nPrice: "<<tmp->price<<
            "\nAmount: "<<tmp->amount<<"\n\n";
        tmp = tmp->next;
    }
    return;
}

// Функция, добавляющая элемент в конец списка
void addlast(product*& bufet, string NewName, double newprice, int newamount){
```

Рис. 3.15: открытие файла срр

10. Используя меню редактора, выключим подсветку синтаксиса.

```
Lab1_Shupletsov.cpp [ -- ] 0 L:[ 1+ 0 1/252 ] *(0 /7764b) 0035 0x023 [ * ] [ X ]
#include <iostream>
#include <cstdlib>
using namespace std;

struct product{
    string name;
    double price;
    int amount;
    product *next;
};

// Функция, добавляющая элемент в начало списка
void addFirst(product*& bufet, string NewName, double newprice, int newamount){
    product *newproduct = new product;
    newproduct->name = NewName;
    product *tmp = bufet;
    while(tmp != NULL){
        if(tmp->name == NewName){
            cout << "This element is already in list";
            return;
        }
        tmp = tmp->next;
    }
    newproduct->price = newprice;
    newproduct->amount = newamount;
    newproduct->next = bufet;
    bufet = newproduct;
    return;
}

// Функция, работающая с пустым листом
void PrintList(product *bufet){
    if (bufet == NULL){
        cout << "The list is empty";
        return;
    }
    product *tmp = bufet;
    while(tmp != NULL){
        cout << "Product: "<<tmp->name<<
        "\nPrice: "<<tmp->price<<
        "\nAmount: "<<tmp->amount<<"\n\n";
        tmp = tmp->next;
    }
    return;
}

// Функция, добавляющая элемент в конец списка
void addLast(product *&bufet, string newname, double newprice, int newamount){
```

Рис. 3.16: выключение подсветки синтаксиса в файле cpp

## **4 Выводы**

Я освоил основные возможности командной оболочки Midnight Commander, приобрел навыки практической работы по просмотру каталогов и файлов, манипуляций с ними.

# **Список литературы**

Кулябов Д.С. “Материалы к лабораторным работам”