

# Командная оболочка Midnight Commander

---

Ридой Хайатуннаби<sup>1</sup>

29 марта, 2024, Москва, Россия

<sup>1</sup>Российский Университет Дружбы Народов

# Цели и задачи работы

---

## Цель лабораторной работы

Ознакомление с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобретение практических навыков по применению команд для работы с файлами и каталогами, по управлению процессами, по проверке использования диска и обслуживанию файловой системы.

# Задачи лабораторной работы

- 1 Изучить возможности Midnight Commander
- 2 Изучить редактор Midnight Commander

# **Процесс выполнения лабораторной работы**

---

# Работа с Midnight Commander

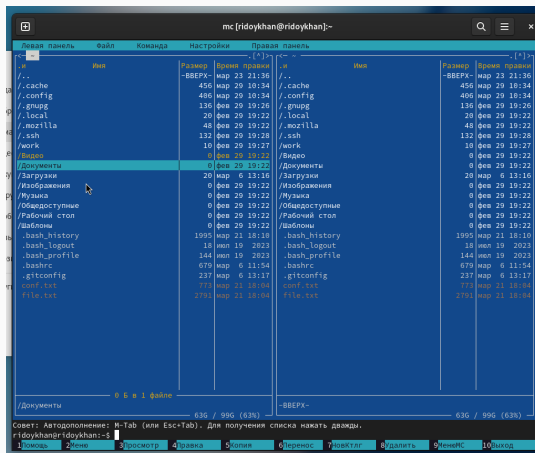


Рис. 1: Запуск mc

# Работа с Midnight Commander

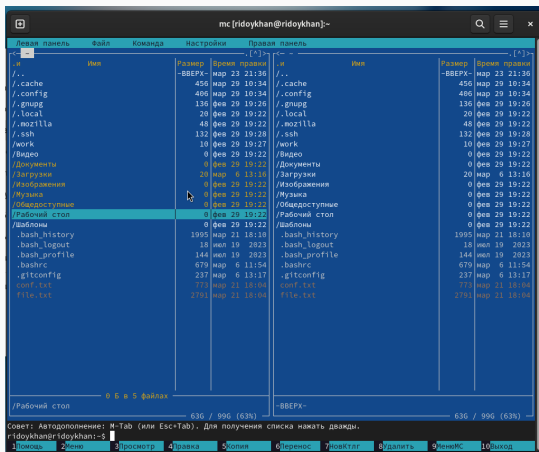


Рис. 2: Выделение

# Работа с Midnight Commander

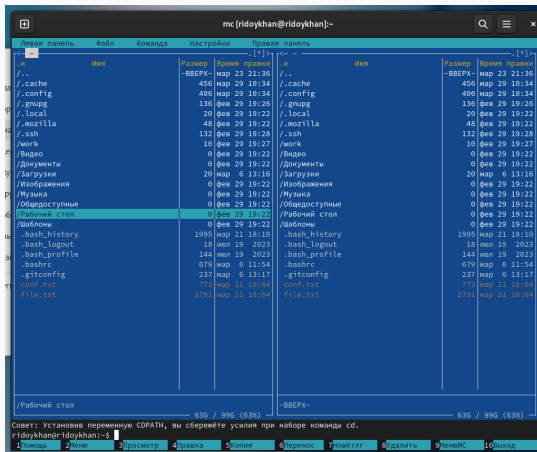


Рис. 3: Отмена



# Работа с Midnight Commander

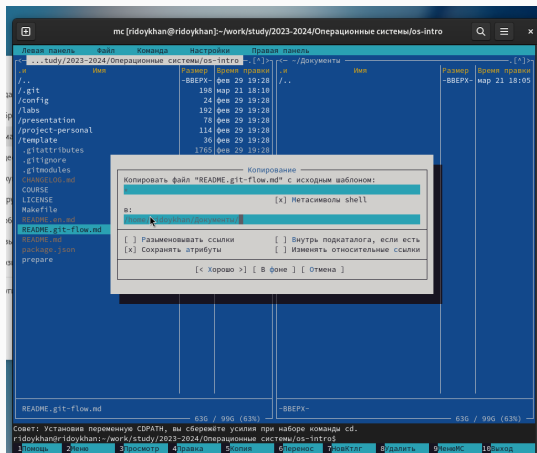


Рис. 4: Копирование

# Работа с Midnight Commander

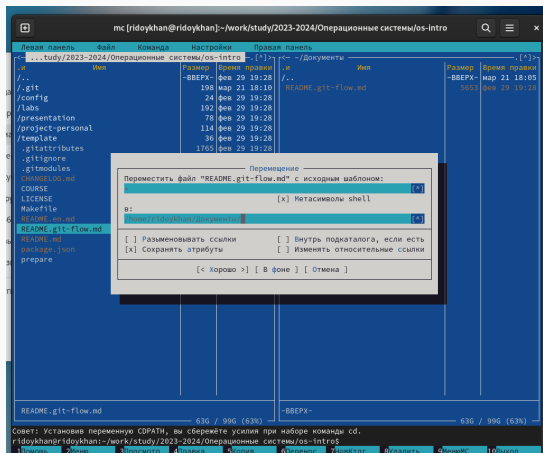


Рис. 5: Перемещение

# Работа с Midnight Commander

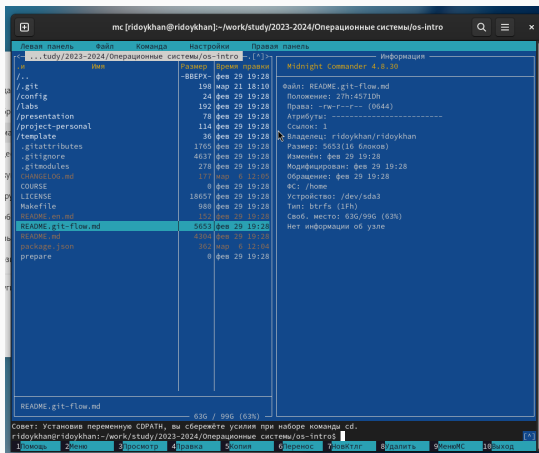


Рис. 6: Информация

# Работа с Midnight Commander

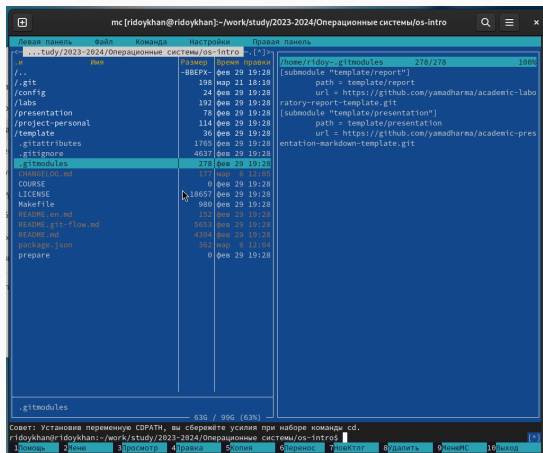


Рис. 7: Быстрый просмотр

# Работа с Midnight Commander

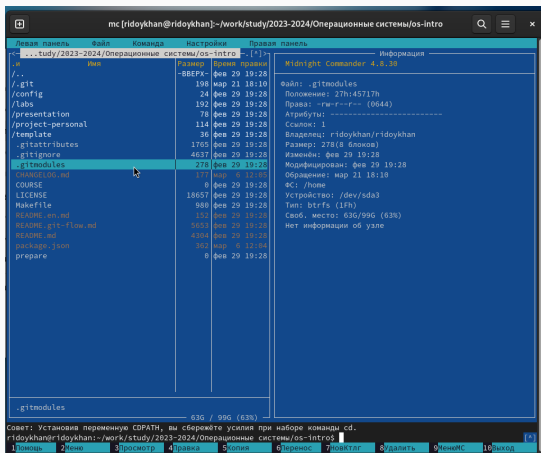
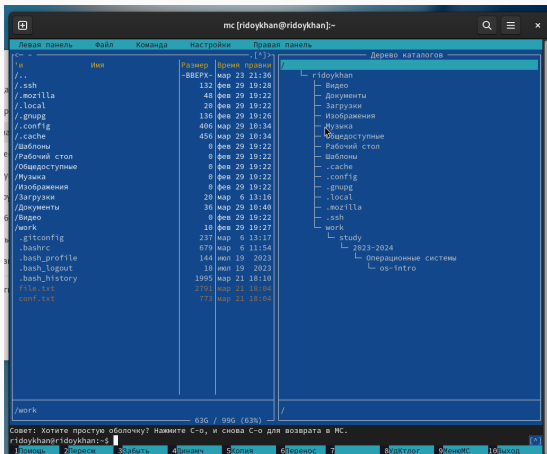


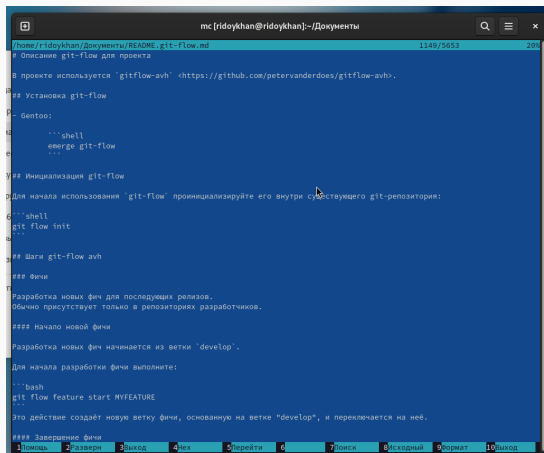
Рис. 8: Информация

## Работа с Midnight Commander



### Рис. 9: Дерево каталогов

# Работа с Midnight Commander



The screenshot shows the Midnight Commander interface. The title bar indicates the current directory is `mc [ridoykhan@ridoykhan]: ~/Документы`. The file list at the top shows `/home/ridoykhan/Документы/README.git-flow.md` with a size of 1149/5653 and a progress bar at 20%. The main pane displays the content of this file, which is a README for `git-flow`. The text includes instructions on how to use `git-flow`, how to install it using `emerge` or `pip`, and how to initialize it within a repository. It also describes the workflow for creating and managing features. At the bottom, there is a navigation bar with buttons for `Помощь`, `Разверн`, `Выход`, `Нек`, `Перейти`, `6`, `Поиск`, `Всходный`, `Формат`, and `Выход`.

```
mc [ridoykhan@ridoykhan]: ~/Документы
/home/ridoykhan/Документы/README.git-flow.md 1149/5653 20%
# Описание git-flow для проекта

В проекте используется 'gitflow-avh' <https://github.com/petervanderdoes/gitflow-avh>.

## Установка git-flow

- Gentoo:

    ``shell
    emerge git-flow
    ...

## Инициализация git-flow

Для начала использования 'git-flow' проинициализируйте его внутри существующего git-репозитория:

    ``shell
    git flow init
    ...

## Шаги git-flow avh

### Фичи

Разработка новых фич для последующих релизов.
Обычно присутствует только в репозиториях разработчиков.

#### Начало новой фичи

Разработка новых фич начинается из ветки 'develop'.

Для начала разработки фичи выполните:

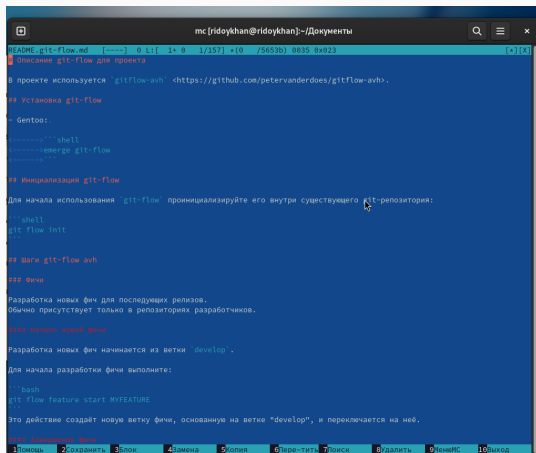
    ``bash
    git flow feature start MYFEATURE
    ...

Это действие создаёт новую ветку фичи, основанную на ветке "develop", и переключается на неё.

### Завершение фичи
```

Рис. 10: Просмотр содержимого текстового файла

# Работа с Midnight Commander



```
mc [ridoykhan@ridoykhan]:~/Документы
README_git-flow.md [-----] 0 Ls: 1+0 1/157 *0 /5653b 0035 0x023 [*][X]
Описание git-flow для проекта

В проекте используется 'gitflow-avh' <https://github.com/petervanderdoes/gitflow-avh>.

## Установка git-flow

- Gentoo:

-----" shell
----->emerge git-flow
-----"

## Инициализация git-flow

Для начала использования 'git-flow' проинициализируйте его внутри существующего git-репозитория:

" shell
git flow init
"

## Шаг git-flow avh

## Ветки

Разработка новых фич для последующих релизов.
Обычно присутствует только в репозиториях разработчиков.

### Начало новой фичи

Разработка новых фич начинается из ветки 'develop'.

Для начала разработки фичи выполните:

" bash
git flow feature start MYFEATURE
"

Это действие создаст новую ветку фичи, основанную на ветке "develop", и переключается на неё.

## Конец ветки фичи
```

**Рис. 11:** Отредактируем содержимое текстового файла без сохранения результатов



# Работа с Midnight Commander

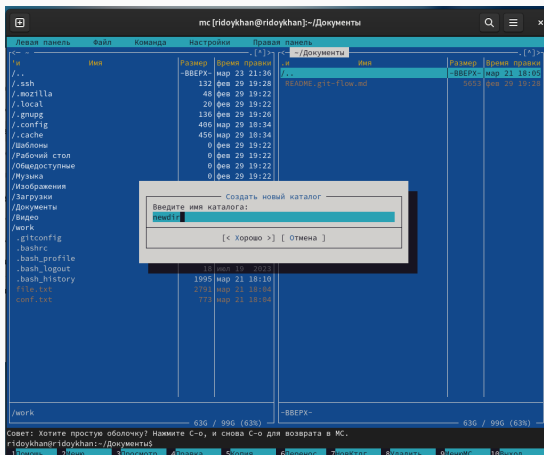


Рис. 12: Создание каталога

# Работа с Midnight Commander

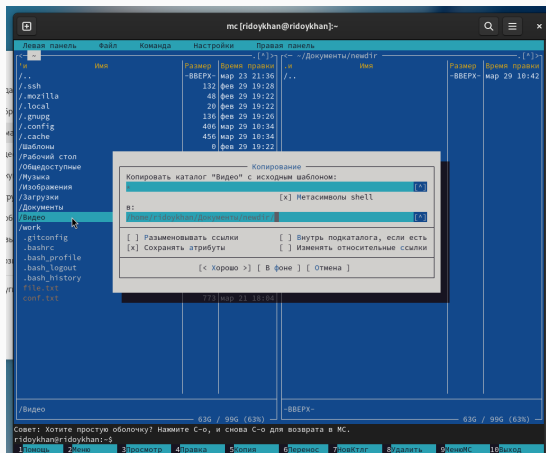


Рис. 13: Копирование в файлов в созданный каталог

# Работа с Midnight Commander

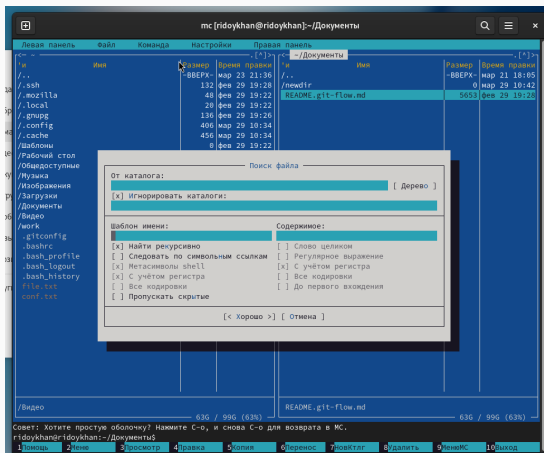


Рис. 14: Поиск файлов

# Работа с Midnight Commander

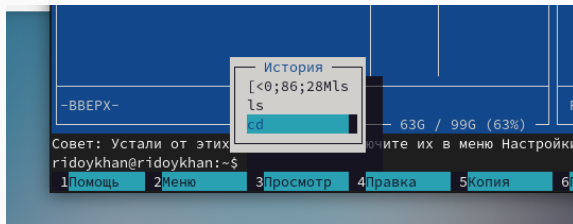


Рис. 15: История команд

# Работа с Midnight Commander

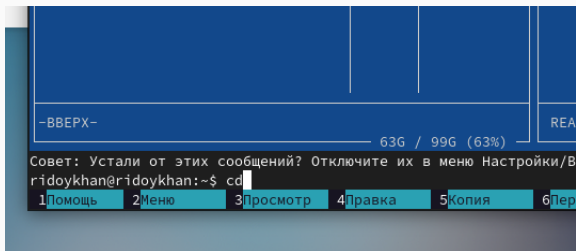


Рис. 16: Переход в домашний каталог

# Работа с Midnight Commander



```
mc.ext.ini [-----] 0 L: [ 1+ 0 1/1136] *0 /26909b/0035 0x023 [*][X]
# Midnight Commander 4.8 extension file
#
# Warning: The structure of this file has been completely changed with the version 4.0!
#
# All lines starting with # or empty lines are ignored.
#
# IMPORTANT: mc scans this file only upon first use or after editing it using the
# mc "Edit extension file" command (F8-c-e). If you edit this file in any other way
# while mc is running, you will need to press F8-c-e and exit the editor for your
# changes to take effect, or exit mc and start it again.
#
# Section name can be anything with following exceptions:
#   there are two reserved section names:
#       mc.ext.ini
#       Default
#   special name pattern:
#       include/xxxx
#   See below for more details.
#
# Section [mc.ext.ini] is mandatory. It contains file metadata.
# "Version" parameter is mandatory. It contains the file format version.
#
# Section [Default] is optional. It is applied only if no other match was found.
#
# Sections like [include/xxxx] can be referenced as "include=xxxx" from other sections.
# Section [include/xxxx] can be located as before as after sections that point to it.
#
# Sections are processed from top to bottom, thus the order is important.
# If there are more than one sections with the same name in this file, the first
# section will be used.
#
# [Default] should be a catch-all action and come last.
#
```

Рис. 17: Просмотр файла расширений

A screenshot of the Midnight Commander file menu. The window title is "menu" with a status bar at the bottom showing "1: Помощь 2: Сохранить 3: Поиск 4: Имена 5: Скопия 6: Перенести 7: Повтор 8: Добавить 9: НайтиС 10: Выход". The menu content includes options like "shell\_patterns=0", "#####", "F The % character", "F The current file (if non-local vfs, file will be copied locally and %f will be full path to it)", "F The current file", "F The current working directory", "F 'Selected files'; the tagged files if any, otherwise the current file", "F Tagged files", "F Tagged files (and they are untagged on return from expand\_format)", "F %view Runs the commands and pipes standard output to the view command", "F If %view is immediately followed by '!', recognize keywords", "F ascii, hex, nroff and unform", "F If the format letter is in uppercase, it refers to the other panel", "F With a number followed the % character you can turn quoting on (default) and off. For example:", "F %f quote expanded macro", "F %if ditto", "F %of don't quote expanded macro", "#####", "t Do something on the current file", "CMD=%(Enter command)", "SCMD %f", "t Do something on the tagged files", "CMD=%(Enter command)", "for i in %t ; do".

```
menu [-----] 1 t: 1*16 17/378 *(732 /11821b) 0010 0x00A [x]
shell_patterns=0

#####
# %F 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)
| SCMD %f

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

Рис. 18: Просмотр файла меню

# Работа с Midnight Commander



Рис. 19: Конфигурация



# Работа с Midnight Commander



Рис. 20: Внешний вид

# Работа с Midnight Commander



Рис. 21: Настройки панелей



Рис. 22: Подтверждение



**Рис. 23:** Оформление



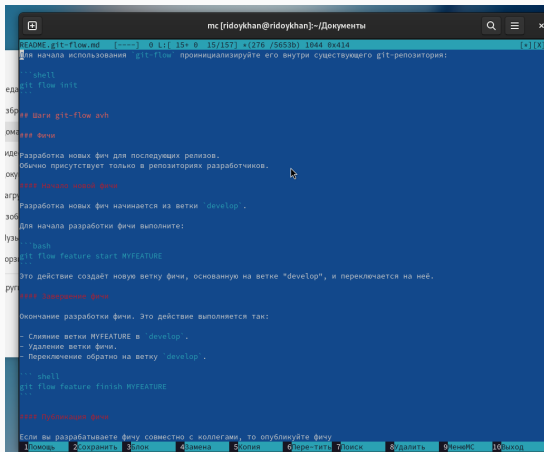
**Рис. 24:** Кодировка символов

# Работа с Midnight Commander



Рис. 25: Распознавание клавиш

# Работа с редактором Midnight Commander



The screenshot shows the Midnight Commander (mc) file editor interface. The title bar indicates the user is 'ridoykhan@ridoykhan' in the directory '~/Документы'. The editor is displaying a file named 'README.git-flow.md'. The text within the editor is as follows:

```
README.git-flow.md [----] 0 L: [ 15+ 0 15/157] + (276 /5653b) 1044 8x414 [•][X]
для начала использования git-flow проинициализируйте его внутри существующего git-репозитория:

''' shell
git flow init
'''

Забр
ом: ## Фича

Иде: Разработка новых фич для последующих релизов.
Ом: Обычно присутствует только в репозиториях разработчиков.
ру:
разр:
агр:
зоб:
луз:
ораз:
Это действие создаёт новую ветку фичи, основанную на ветке "develop", и переключается на неё.
ру:
разр:
Окончание разработки фичи. Это действие выполняется так:
- Слияние ветки MYFEATURE в 'develop'.
- Удаление ветки фичи.
- Переключение обратно на ветку 'develop'.

''' shell
git flow feature finish MYFEATURE
'''

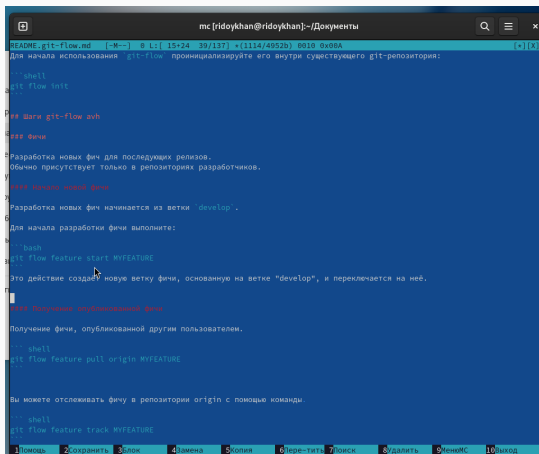
## Публикация фичи

Если вы разрабатываете фичу совместно с коллегами, то опубликуйте фичу
```

At the bottom of the window, there is a status bar with icons and labels for various actions: Помочь, Сохранить, Выход, and others.

Рис. 26: Файл с текстом

# Работа с редактором Midnight Commander



The screenshot shows the Midnight Commander (mc) interface. The title bar indicates the user is 'mc [ridoykhan@ridoykhan]~/Документы'. The file being edited is 'README.git-flow.md'. The text inside the editor is as follows:

```
README.git-flow.md  [-M--]  0 Li 15+24 39/137] +(1114/4952b) 0010 0x0BA
Для начала использования git-flow проинициализируйте его внутри существующего git-репозитория:

'' shell
git flow init

## Шаги git-flow avh
## Фичи

Разработка новых фич для последующих релизов.
Обычно присутствует только в репозиториях разработчиков.

### Начало новой фичи

Разработка новых фич начинается из ветки develop .
Для начала разработки фичи выполните:

'' bash
git flow feature start MYFEATURE

Это действие создаст новую ветку фичи, основанную на ветке "develop", и переключается на неё.

### Получение опубликованной фичи

Получение фичи, опубликованной другим пользователем.

'' shell
git flow feature pull origin MYFEATURE

Вы можете отслеживать фичу в репозитории origin с помощью команды.

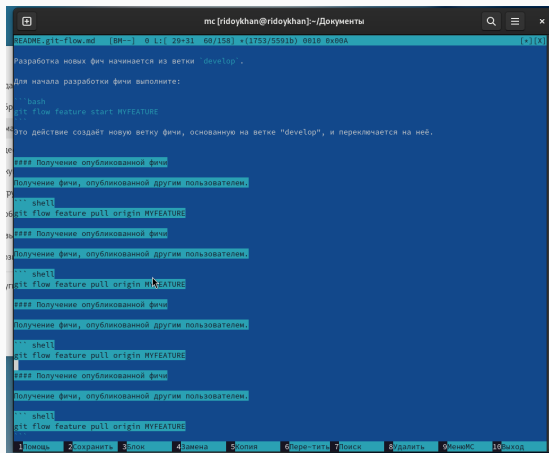
'' shell
git flow feature track MYFEATURE
```

At the bottom of the window, there is a toolbar with icons and labels for various actions: 1.Помощь (Help), 2.Сохранить (Save), 3.Скопировать (Copy), 4.Вставить (Paste), 5.Копия (Copy), 6.Переместить (Move), 7.Поиск (Search), 8.Удалить (Delete), 9.Выход (Exit), 10.Выход (Exit).

Рис. 27: Файл с текстом



# Работа с редактором Midnight Commander



The screenshot shows the Midnight Commander (mc) interface. The title bar indicates the user is 'mc [ridoykhan@ridoykhan]: ~/Документы'. The main window displays a file named 'README\_git-flow.md' with a size of 69,158 bytes. The file content is a text document describing the Git workflow, starting with 'Разработка новых фич начинается из ветки develop.' and detailing steps for creating, pulling, and pushing features. The text is displayed in a monospaced font on a dark blue background. At the bottom, there is a menu bar with options: 'Помощь', 'Сохранить', 'Блок', 'Замена', 'Копия', 'Пере-тит', 'Поиск', 'Удалить', 'Меню', and 'Выход'.

```
mc [ridoykhan@ridoykhan]: ~/Документы
README_git-flow.md  [BIN]  0 L: 29+31  69/158  +(1753/5501b) 0810 0x00A  (*)[L]

Разработка новых фич начинается из ветки  develop .

Для начала разработки фичи выполните:

12  'bash
30  git flow feature start MYFEATURE
40  Это действие создаёт новую ветку фичи, основанную на ветке "develop", и переключается на неё.
60  ### Получение опубликованной фичи
80  Получение фичи, опубликованной другим пользователем.
90  ' ' shell
100 git flow feature pull origin MYFEATURE
110  ### Получение опубликованной фичи
120  Получение фичи, опубликованной другим пользователем.
130  ' ' shell
140 git flow feature pull origin MYFEATURE
150  ### Получение опубликованной фичи
160  Получение фичи, опубликованной другим пользователем.
170  ' ' shell
180 git flow feature pull origin MYFEATURE
190  ### Получение опубликованной фичи
200  Получение фичи, опубликованной другим пользователем.
210  ' ' shell
220 git flow feature pull origin MYFEATURE
230  ### Получение опубликованной фичи
240  Получение фичи, опубликованной другим пользователем.
250  ' ' shell
260 git flow feature pull origin MYFEATURE
270  ### Получение опубликованной фичи
280  Получение фичи, опубликованной другим пользователем.
290  ' ' shell
300 git flow feature pull origin MYFEATURE
310  ### Получение опубликованной фичи
320  Получение фичи, опубликованной другим пользователем.
330  ' ' shell
340 git flow feature pull origin MYFEATURE
350  ### Получение опубликованной фичи
360  Получение фичи, опубликованной другим пользователем.
370  ' ' shell
380 git flow feature pull origin MYFEATURE
390  ### Получение опубликованной фичи
400  Получение фичи, опубликованной другим пользователем.
410  ' ' shell
420 git flow feature pull origin MYFEATURE
430  ### Получение опубликованной фичи
440  Получение фичи, опубликованной другим пользователем.
450  ' ' shell
460 git flow feature pull origin MYFEATURE
470  ### Получение опубликованной фичи
480  Получение фичи, опубликованной другим пользователем.
490  ' ' shell
500 git flow feature pull origin MYFEATURE
510  ### Получение опубликованной фичи
520  Получение фичи, опубликованной другим пользователем.
530  ' ' shell
540 git flow feature pull origin MYFEATURE
550  ### Получение опубликованной фичи
560  Получение фичи, опубликованной другим пользователем.
570  ' ' shell
580 git flow feature pull origin MYFEATURE
590  ### Получение опубликованной фичи
600  Получение фичи, опубликованной другим пользователем.
610  ' ' shell
620 git flow feature pull origin MYFEATURE
630  ### Получение опубликованной фичи
640  Получение фичи, опубликованной другим пользователем.
650  ' ' shell
660 git flow feature pull origin MYFEATURE
670  ### Получение опубликованной фичи
680  Получение фичи, опубликованной другим пользователем.
690  ' ' shell
700 git flow feature pull origin MYFEATURE
710  ### Получение опубликованной фичи
720  Получение фичи, опубликованной другим пользователем.
730  ' ' shell
740 git flow feature pull origin MYFEATURE
750  ### Получение опубликованной фичи
760  Получение фичи, опубликованной другим пользователем.
770  ' ' shell
780 git flow feature pull origin MYFEATURE
790  ### Получение опубликованной фичи
800  Получение фичи, опубликованной другим пользователем.
810  ' ' shell
820 git flow feature pull origin MYFEATURE
830  ### Получение опубликованной фичи
840  Получение фичи, опубликованной другим пользователем.
850  ' ' shell
860 git flow feature pull origin MYFEATURE
870  ### Получение опубликованной фичи
880  Получение фичи, опубликованной другим пользователем.
890  ' ' shell
900 git flow feature pull origin MYFEATURE
910  ### Получение опубликованной фичи
920  Получение фичи, опубликованной другим пользователем.
930  ' ' shell
940 git flow feature pull origin MYFEATURE
950  ### Получение опубликованной фичи
960  Получение фичи, опубликованной другим пользователем.
970  ' ' shell
980 git flow feature pull origin MYFEATURE
990  ### Получение опубликованной фичи
1000  Получение фичи, опубликованной другим пользователем.

[Помощь] [Сохранить] [Блок] [Замена] [Копия] [Пере-тит] [Поиск] [Удалить] [Меню] [Выход]
```

Рис. 28: Копирование фрагмента

# Работа с редактором Midnight Commander

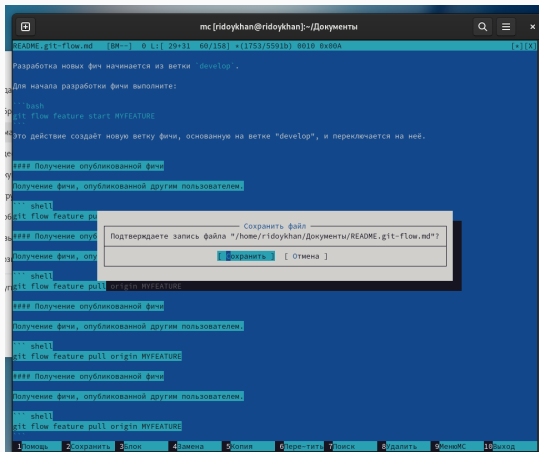
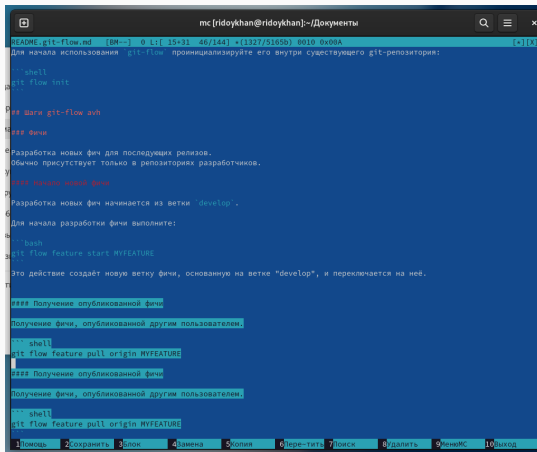


Рис. 29: Сохранение

# Работа с редактором Midnight Commander



```
mc [ridoykhan@ridoykhan]:~/Документы
README.git-flow.md [BM---] 0 Li( 15+31 46/144) +(1327/51656) 0010 0x00A [*|X]
Для начала использования git-flow проинициализируйте его внутри существующего git-репозитория:

...
"shell
git flow init
...

$! Шаг1 git-flow avh

### Фичи

Разработка новых фич для последующих релизов.
Обычно присутствует только в репозиториях разработчиков.

### Начало новой фичи

Разработка новых фич начинается из ветки "develop".
Для начала разработки фичи выполните:

...
"bash
git flow feature start MYFEATURE
...

Это действие создаст новую ветку фичи, основанную на ветке "develop", и переключается на неё.

### Получение опубликованной фичи

Получение фичи, опубликованной другим пользователем.

...
"shell
git flow feature pull origin MYFEATURE
...

### Получение опубликованной фичи

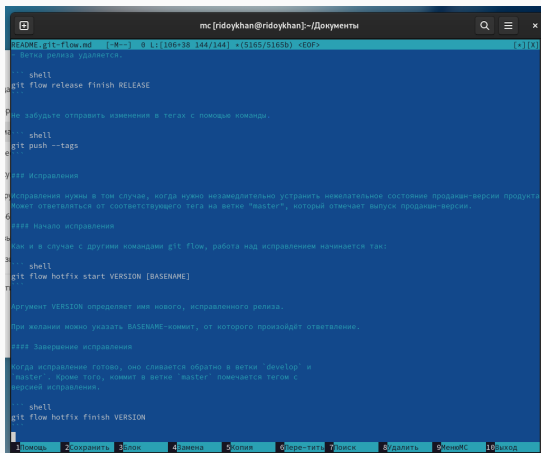
Получение фичи, опубликованной другим пользователем.

...
"shell
git flow feature pull origin MYFEATURE
...

1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Пере-тить 7Поиск 8Удалить 9ВнеМС 10Выход
```

Рис. 30: Отмена

# Работа с редактором Midnight Commander



```
mc [ridoykhan@ridoykhan]:~/Документы
README.git-flow.md [-M-- 0 Li:106+38 144/144] *(5165/5165b) <EOF>
  ветка релиза удалится.

  .. shell
git flow release finish RELEASE
  ..

не забудьте отправить изменения в тегх с помощью команды.

  .. shell
git push --tags
  ..

### Исправления

Исправления нужны в том случае, когда нужно незамедлительно устранить нежелательное состояние продаван-версии продукта.
Может отвечать от соответствующего тега на ветке "master", который отмечает выпуск продаван-версии.

### Начало исправления

Так и в случае с другими командами git flow, работа над исправлением начинается так:

  .. shell
git flow hotfix start VERSION [BASENAME]
  ..

Аргумент VERSION определит имя нового, исправленного релиза.

При желании можно указать BASENAME-коммит, от которого произойдет ответвление.

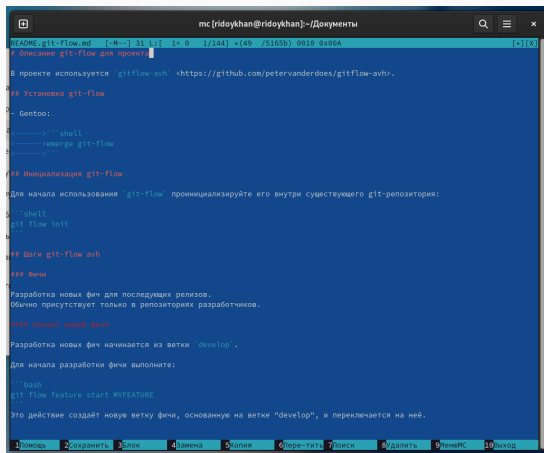
### Завершение исправления

Когда исправление готово, оно сливается обратно в ветки "develop" и
"master". Кроме того, коммит в ветке "master" помечается тегом с
версией исправления.

  .. shell
git flow hotfix finish VERSION
```

Рис. 31: Переход в конец файла

# Работа с редактором Midnight Commander



```
mc [ridoykhan@ridoykhan]: ~/Документы
README:git-flow.md [-M--] 31 KiB 1+0 1/144 *x(49 /5165b) 0010 8x00A [*] [X]
# Описание git-flow для проекта

В проекте используется 'git-flow-avh' <https://github.com/petervanderdoes/git-flow-avh>.

## Установка git-flow

- Gentoo:

-----> shell
-----> emerge git-flow
----->

## Инициализация git-flow

Для начала использования 'git-flow' проинициализируйте его внутри существующего git-репозитория:

$ shell
git flow init

## Шаг git-flow avh

## Фичи

Разработка новых фич для последующих релизов.
Обычно присутствует только в репозиториях разработчиков.

### Начало новой фичи

Разработка новых фич начинается из ветки 'develop'.

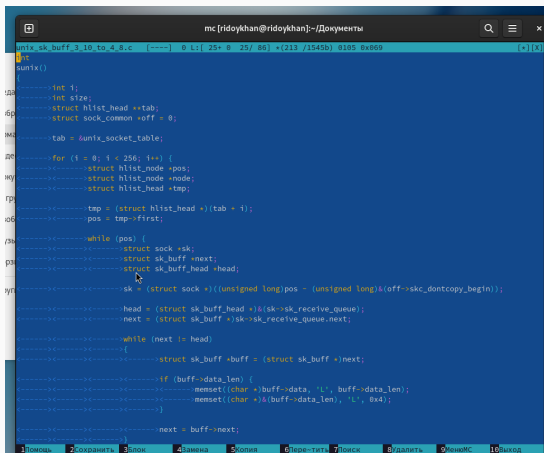
Для начала разработки фичи выполните:

$ bash
git flow feature start MYFEATURE

Это действие создаёт новую ветку фичи, основанную на ветке "develop", и переключается на неё.
```

Рис. 32: Переход в начало файла

# Работа с редактором Midnight Commander



```
mc [ridoykhan@ridoykhan]:~/Документы
unix_sk_buff.3.10.to.4.8.c  [----]  0  L: [ 25+ 0  25/ 86]  +(213 /1545b) 0105 8x860  (*) [X]
int
unix()
{
    int i;
    int size;
    struct hlist_head *tab;
    struct sock_common *off = 0;

    tab = &unix_socket_table;

    for (i = 0; i < 256; i++) {
        struct hlist_node *pos;
        struct hlist_node *node;
        struct hlist_head *tmp;

        tmp = (struct hlist_head *) (tab + i);
        pos = tmp->first;

        while (pos) {
            struct sock *sk;
            struct sk_buff *next;
            struct sk_buff_head *head;

            sk = (struct sock *) ((unsigned long) pos - (unsigned long) &off->skc_dontcopy_begin);

            head = (struct sk_buff_head *) &(sk->sk_receive_queue);
            next = (struct sk_buff *) sk->sk_receive_queue.next;

            while (next != head) {
                struct sk_buff *buff = (struct sk_buff *) next;

                if (buff->data_len) {
                    memset((char *) buff->data, 'L', buff->data_len);
                    memset((char *) &(buff->data_len), 'L', 0x4);
                }

                next = buff->next;
            }
        }
    }
}
```

1 Помощь 2 Сохранить 3 Выход 4 Замена 5 Копия 6 Пере-титул 7 Поиск 8 Удалить 9 Меню MC 10 Выход

Рис. 33: Файл с программой

# Работа с редактором Midnight Commander

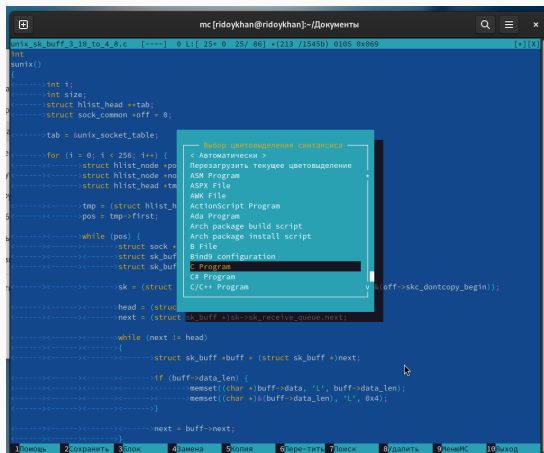


Рис. 34: Цветовыделение синтаксиса

## **Выводы по проделанной работе**

---



В данной работе мы ознакомились с инструментами поиска файлов и фильтрации текстовых данных. А также приобрели практические навыки по управлению процессами.