

Командная оболочка Midnight Commander

Ван Сихэм Франклин О Нил Джон¹

29 марта, 2024, Москва, Россия

¹Российский 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

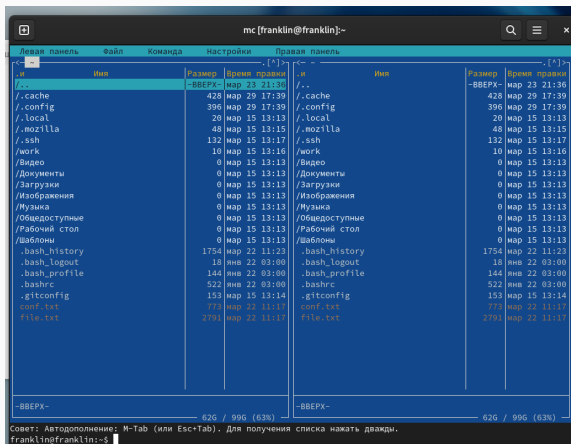


Рис. 1: Запуск mc

Работа с Midnight Commander

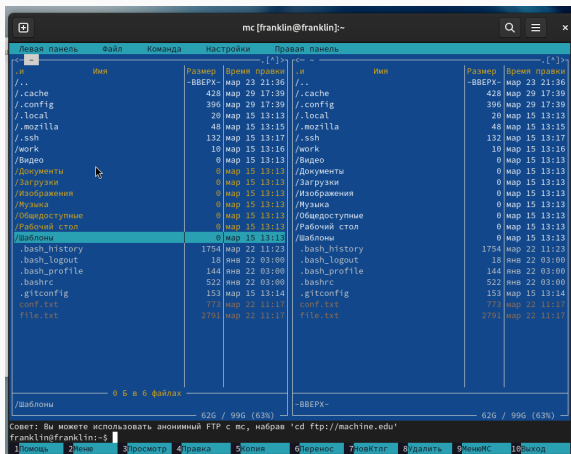


Рис. 2: Выделение

Работа с Midnight Commander

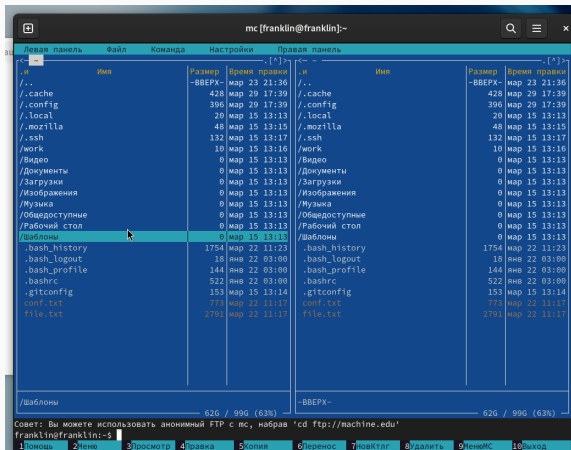


Рис. 3: Отмена

Работа с Midnight Commander

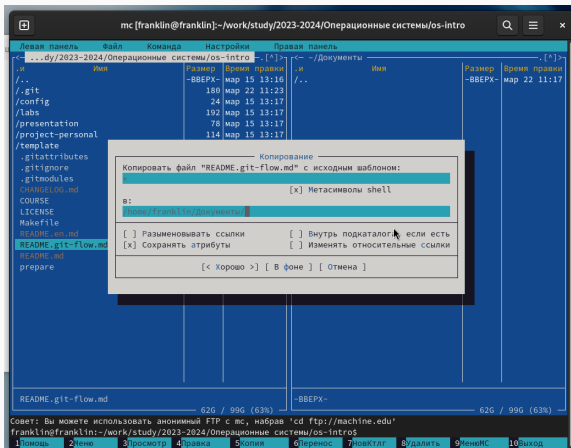


Рис. 4: Копирование

Работа с Midnight Commander

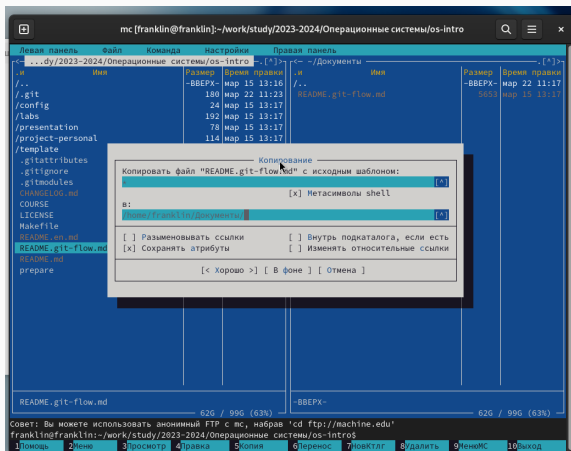


Рис. 5: Перемещение

Работа с Midnight Commander

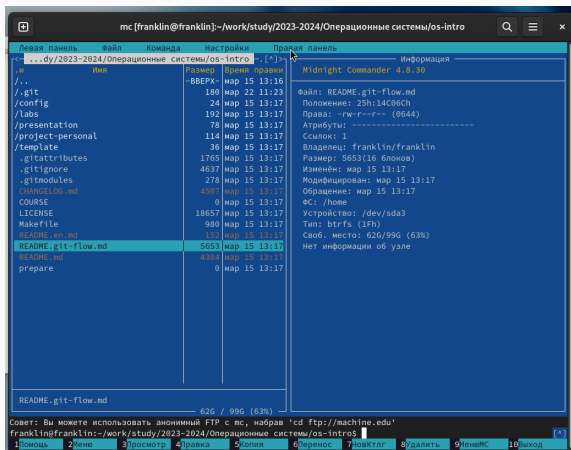


Рис. 6: Информация

Работа с Midnight Commander

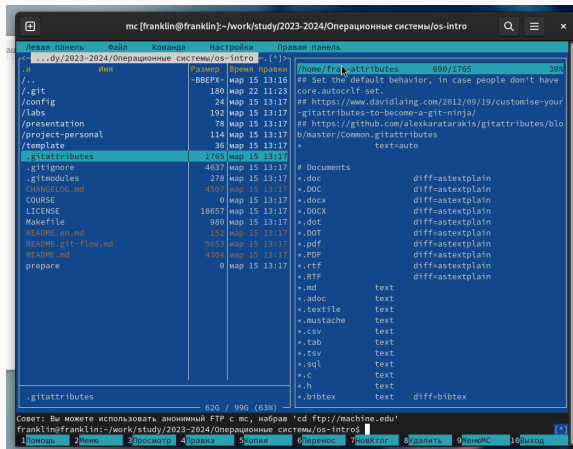


Рис. 7: Быстрый просмотр

Работа с Midnight Commander

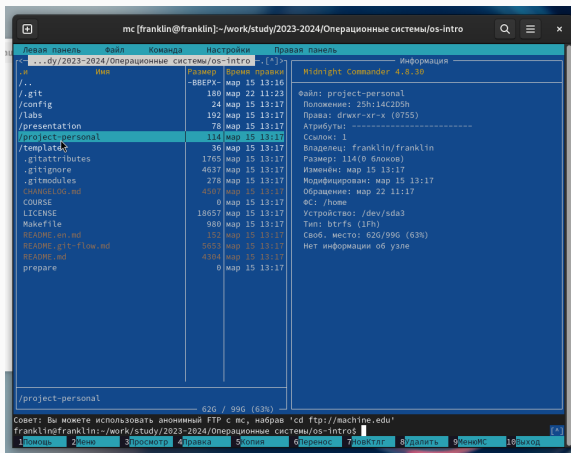


Рис. 8: Информация

Работа с Midnight Commander

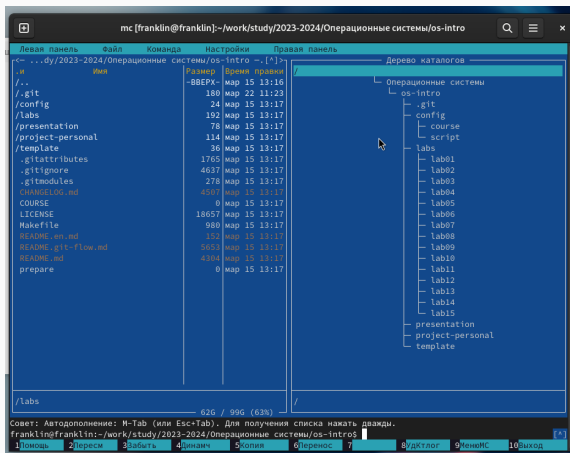
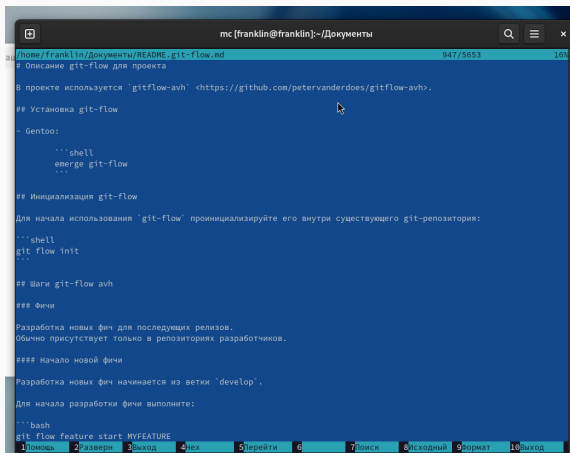


Рис. 9: Дерево каталогов

Работа с Midnight Commander



The screenshot shows the Midnight Commander (mc) interface. The title bar indicates the current directory is ~/Документы. The main window displays the contents of the file README.git-flow.md. The text within the file describes the git-flow workflow, including installation instructions for Gentoo and the steps for initializing and starting a new feature branch. The interface includes a top status bar with file statistics and a bottom navigation bar with numbered shortcuts.

```
mc [franklin@franklin]:~/Документы
/home/franklin/Документы/README.git-flow.md 947/5653 16%
# 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

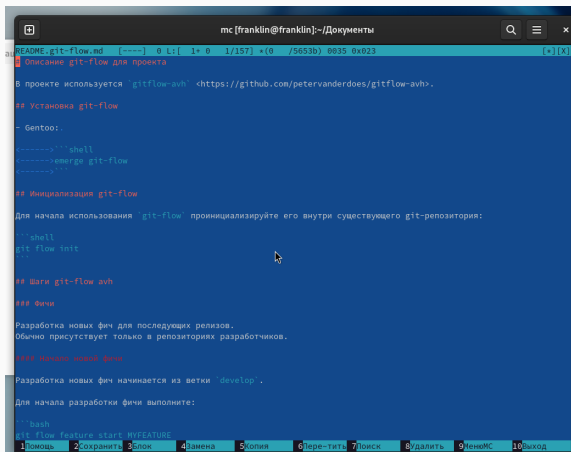
Разработка новых фич начинается из ветки 'develop'.

Для начала разработки фичи выполните:

    ```bash
 git flow feature start MYFEATURE
 1Помощь 2Разверн 3Выход 4Lex 5Перейти 6 7Поиск 8Скопийный 9Формат 10Выход
```

Рис. 10: Просмотр содержимого текстового файла

# Работа с Midnight Commander



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



# Работа с Midnight Commander

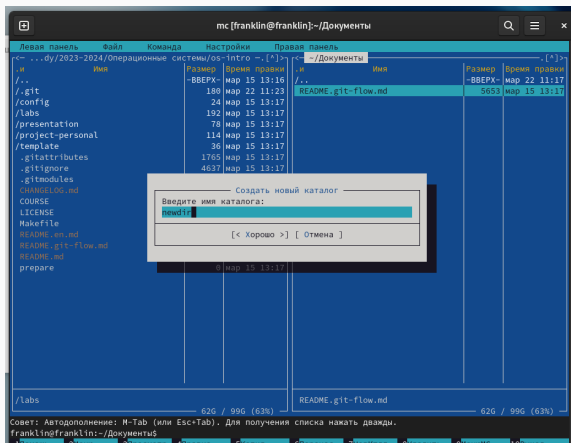


Рис. 12: Создание каталога

# Работа с Midnight Commander

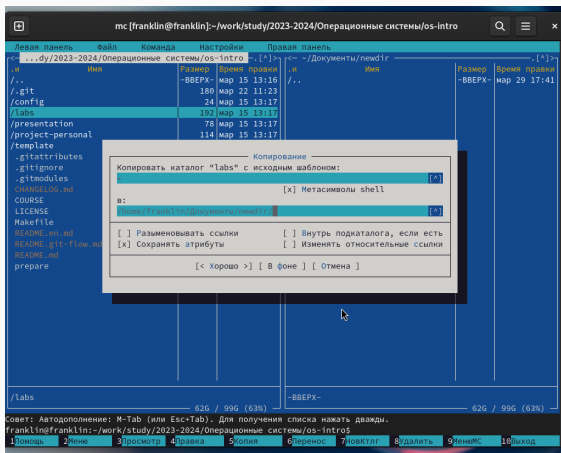


Рис. 13: Копирование в файлов в созданный каталог

# Работа с Midnight Commander

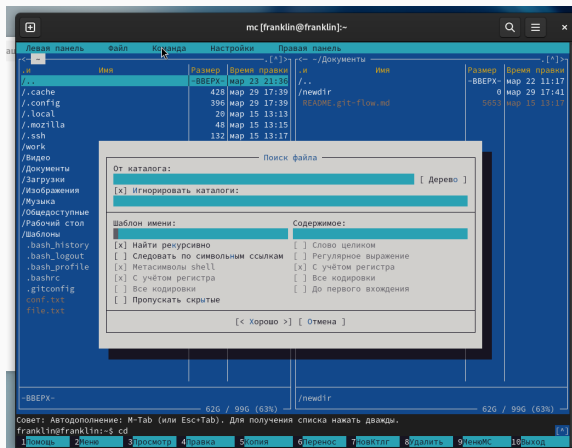


Рис. 14: Поиск файлов

# Работа с Midnight Commander

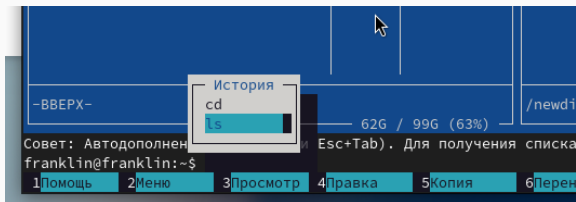
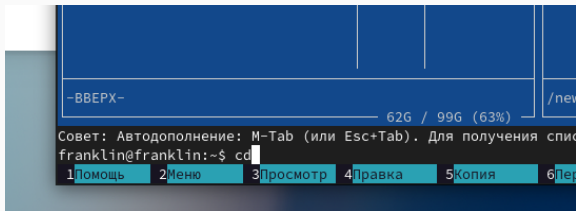


Рис. 15: История команд

# 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

# Работа с Midnight Commander

A screenshot of the Midnight Commander file menu. The window title is "menu" with a status bar at the top showing "1 t: 1\*16 17/378 \*(732 /11821b) 0010 0x00A" and a close button. The menu text is as follows:

```
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)
%v Run 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
```

At the bottom of the window is a toolbar with 10 icons and their labels: 1. Помощь (Help), 2. Сохранить (Save), 3. Поиск (Find), 4. Имена (Names), 5. Скопия (Copy), 6. Перенести (Move), 7. Повтор (Repeat), 8. Удалить (Delete), 9. НенавС (Not a File), 10. Выход (Exit).

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



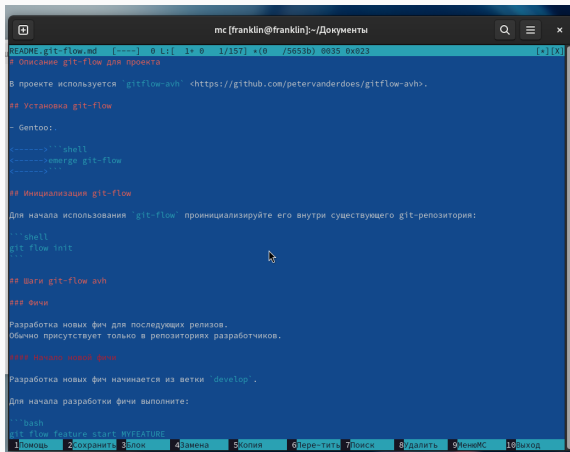
**Рис. 24:** Кодировка символов

# Работа с Midnight Commander



Рис. 25: Распознавание клавиш

# Работа с редактором Midnight Commander



The screenshot shows the Midnight Commander (MC) interface. The title bar indicates the user is 'franklin' and the current directory is '~/Документы'. The main window displays the contents of a file named 'README.git-flow.md'. The text is as follows:

```
README.git-flow.md [----] 0 L: 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
''

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'.

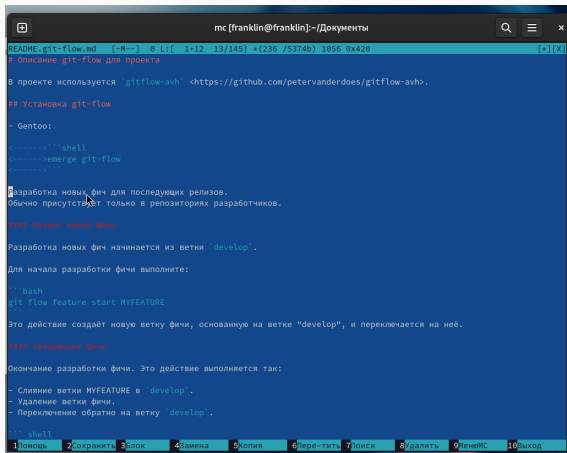
Для начала разработки фичи выполните:

'' bash
git flow feature start MYFEATURE
```

At the bottom of the window, there is a status bar with a row of icons and a row of numbers: 1Помощь, 2Сохранить, 3Блок, 4Замена, 5Копия, 6Пере-тить, 7Поиск, 8Удалить, 9МенюМС, 10Выход.

Рис. 26: Файл с текстом

# Работа с редактором Midnight Commander



```
mc [franklin@franklin]:~/Документы
README.git-flow.md [-M~] 0 L: [1*12 13/145] *(236 /5374b) 1056 0x420 [~][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
~~~~~'''

# 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

Окончание разработки фичи. Это действие выполняется так:

- Слияние ветки MYFEATURE в 'develop'.
- Удаление ветки фичи.
- Переключение обратно на ветку 'develop'.

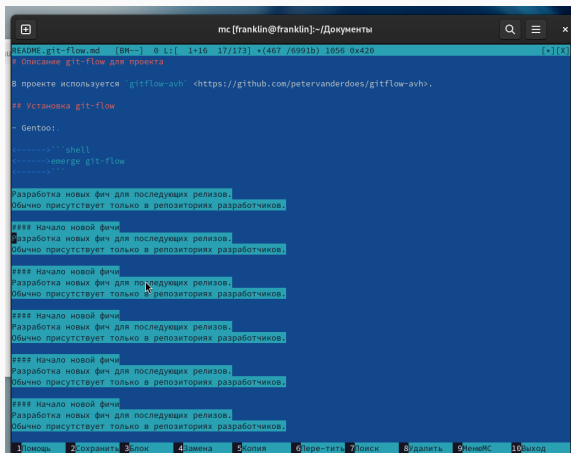
''' shell
```

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

Рис. 27: Файл с текстом



# Работа с редактором Midnight Commander



```
mc [franklin@franklin]:~/Документы
README.git-flow.md [BM--]  0 l: 1+16 17/173  *(467 /6991b) 1056 0x420  (*)[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
-----'''

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

#### Начало новой фичи
Разработка новых фич для последующих релизов.
Обычно присутствует только в репозиториях разработчиков.

#### Начало новой фичи
Разработка новых фич для последующих релизов.
Обычно присутствует только в репозиториях разработчиков.

#### Начало новой фичи
Разработка новых фич для последующих релизов.
Обычно присутствует только в репозиториях разработчиков.

#### Начало новой фичи
Разработка новых фич для последующих релизов.
Обычно присутствует только в репозиториях разработчиков.

#### Начало новой фичи
Разработка новых фич для последующих релизов.
Обычно присутствует только в репозиториях разработчиков.

#### Начало новой фичи
Разработка новых фич для последующих релизов.
Обычно присутствует только в репозиториях разработчиков.

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

Рис. 28: Копирование фрагмента

# Работа с редактором Midnight Commander

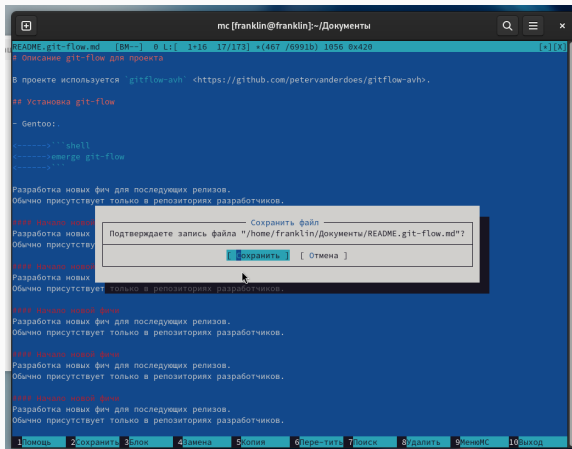
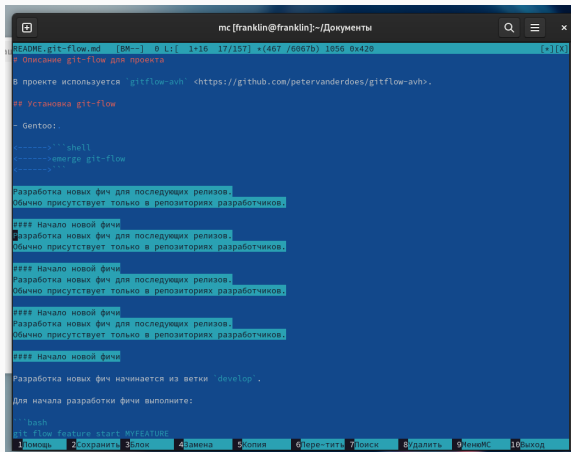


Рис. 29: Сохранение

# Работа с редактором Midnight Commander



The screenshot shows the Midnight Commander (MC) interface. The title bar indicates the user is 'franklin@franklin' in the '~ / Документы' directory. The main window displays the contents of a file named 'README.git-flow.md'. The text within the file describes the 'git-flow' workflow, including instructions on how to set it up using 'gitflow-avh' and 'Gentoo'. It also outlines the development process, such as creating new features from the 'develop' branch and the typical structure of commit messages (e.g., '##### Начало новой фичи'). At the bottom of the window, a status bar shows the current file 'git\_flow.feature.start MYFEATURE' and a numeric keypad for navigation (1-10).

```
mc [franklin@franklin]:~/Документы
README.git-flow.md [8M--] 0 L: [ 1+16 17/157] *(467 /6867b) 1856 0x428 (*) [X]
# Описание git-flow для проекта

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

## Установка git-flow

- Gentoo:

-----'''shell
-----xmerge git-flow
-----'''

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

#### Начало новой фичи
Разработка новых фич для последующих релизов.
Обычно присутствует только в репозиториях разработчиков.

#### Начало новой фичи
Разработка новых фич для последующих релизов.
Обычно присутствует только в репозиториях разработчиков.

#### Начало новой фичи
Разработка новых фич для последующих релизов.
Обычно присутствует только в репозиториях разработчиков.

#### Начало новой фичи
Разработка новых фич для последующих релизов.
Обычно присутствует только в репозиториях разработчиков.

#### Начало новой фичи
Разработка новых фич для последующих релизов.
Обычно присутствует только в репозиториях разработчиков.

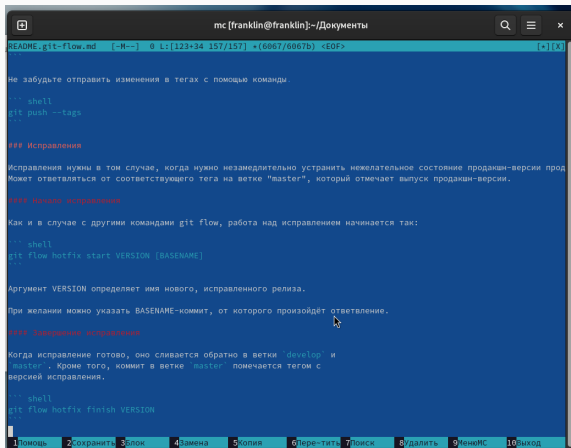
Разработка новых фич начинается из ветки 'develop'.

Для начала разработки фичи выполните:

    bash
git flow feature start MYFEATURE
1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Пере-тить 7Поиск 8Удалить 9МенюМС 10Выход
```

Рис. 30: Отмена

# Работа с редактором Midnight Commander



The screenshot shows the Midnight Commander (mc) interface. The title bar indicates the user is 'mc [franklin@franklin]:~/Документы'. The file being edited is 'README.git-flow.md'. The status bar at the bottom shows the file is at line 0, column 1, with a size of 123+34 157/157 bytes. The file content is displayed in a blue background with white text. The text includes instructions on how to use git flow to create a new production-ready version, including commands like 'git push --tags' and 'git flow hotfix start VERSION [BASENAME]'. The file ends with a shell command 'git flow hotfix finish VERSION'. The status bar at the bottom of the window shows a series of numbered icons: 1. Помощь, 2. Сохранить, 3. Блок, 4. Замена, 5. Копия, 6. Пере-титл, 7. Поиск, 8. Удалить, 9. Меню, 10. Выход.

```
mc [franklin@franklin]:~/Документы
README.git-flow.md [-M--] 0 L:123+34 157/157] +(6067/6067b) <EOF> [+][X]

...

Не забудьте отправить изменения в тегах с помощью команды.

''' shell
git push --tags
'''

### Исправления

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

### Начало исправления

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

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

Аргумент VERSION определяет имя нового, исправленного релиза.

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

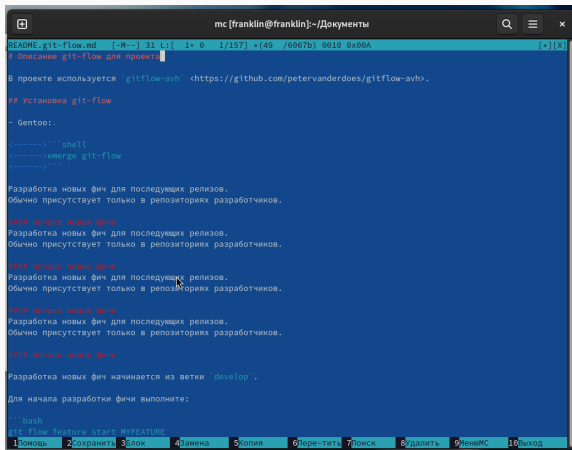
### Завершение исправления

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



The screenshot shows the Midnight Commander (mc) interface. The title bar indicates the user is 'franklin@franklin' in the directory '~/Документы'. The file being edited is 'README.git-flow.md'. The content of the file is as follows:

```
README.git-flow.md [-M--] 31 L: { 1+ 0 1/157 } + (49 /6067b) 0010 0x00A [*][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
<----->''

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

### Начало новой фичи
Разработка новых фич для последующих релизов.
Обычно присутствует только в репозиториях разработчиков.

### Начало новой фичи
Разработка новых фич для последующих релизов.
Обычно присутствует только в репозиториях разработчиков.

### Начало новой фичи
Разработка новых фич для последующих релизов.
Обычно присутствует только в репозиториях разработчиков.

### Начало новой фичи
Разработка новых фич для последующих релизов.
Обычно присутствует только в репозиториях разработчиков.

Разработка новых фич начинается из ветки 'develop'.

Для начала разработки фичи выполните:

'''bash
git flow feature start MYFEATURE
```

At the bottom of the window, there is a status bar with a menu: 1.Помощь, 2.Сохранить, 3.Век, 4.Замена, 5.Копия, 6.Переместить, 7.Поиск, 8.Удалить, 9.Меню, 10.Выход.

Рис. 32: Переход в начало файла

## Работа с редактором Midnight Commander

```

[+]
mc [franklin@franklin]:~/Документы
tcp_sk_buf_3_10_to_4_8.c [----] 0 L: 25+ 0 25/ 83) +(212 /1521b) 0105 0x069
int
tcp()
{
    ----->int i;
    ----->struct inet_hashinfo *tab;
    ----->struct sock_common *off = 0;

    ----->tab = &tcp_hashinfo;

    ----->for (i = 0; i < 32; i++) {
    ----->----->struct hlist_nulls_node *pos;

    ----->----->pos = tab->listening_hash[i].head.first;

    ----->----->while (!((unsigned long)pos & 1)) {
    ----->----->----->struct sock *sk;
    ----->----->----->struct sk_buff *next;
    ----->----->----->struct sk_buff_head *head;
    ----->----->----->struct hlist_nulls_node *node;

    ----->----->----->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;

```

### Рис. 33: Файл с программой

# Работа с редактором Midnight Commander

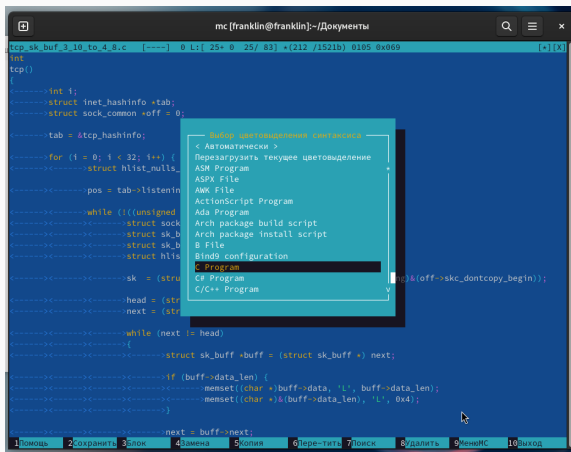


Рис. 34: Цветовыделение синтаксиса

## **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.