

Общая версия Linux. Уровень 1

Настройка интерфейса командной строки и знакомство с ним



На этом уроке

1. Узнаем, как подключиться к ОС, используя протокол удалённого управления SSH.
2. Познакомимся с командной строкой OS Linux.
3. Познакомимся с консольными текстовыми редакторами.
4. Настроим подключение по SSH, используя аутентификацию по ключу.

Оглавление

[Знакомство с интерфейсом](#)

[Навигация по файловой системе и основные операции с файлами и каталогами](#)

[Знакомство с текстовыми редакторами](#)

[Текстовый редактор vi/Vim](#)

[Текстовый редактор nano](#)

[Практическое задание](#)

[Дополнительные материалы](#)

[Используемые источники](#)

Глоссарий

[Терминал](#) — интерфейс взаимодействия между пользователем и операционной системой.

SSH-клиент — программное обеспечение, позволяющее подключиться к серверу, используя протокол SSH. Рекомендуемые из-за простоты установки SSH-клиенты: в Windows можно использовать [PowerShell](#) с 10 -й версии или [PuTTY](#) - в более старых версиях, в macOS — [iTerm2](#), в Linux — встроенное приложение [GNOME Terminal](#).

[Аутентификация](#) — процедура проверки подлинности, например сравнением введённого пароля пользователя с паролем, сохранённым в базе данных паролей.

Консольные текстовые редакторы — редакторы текста для командной строки в ОС Linux. [Небольшой обзор редакторов](#).

Знакомство с интерфейсом

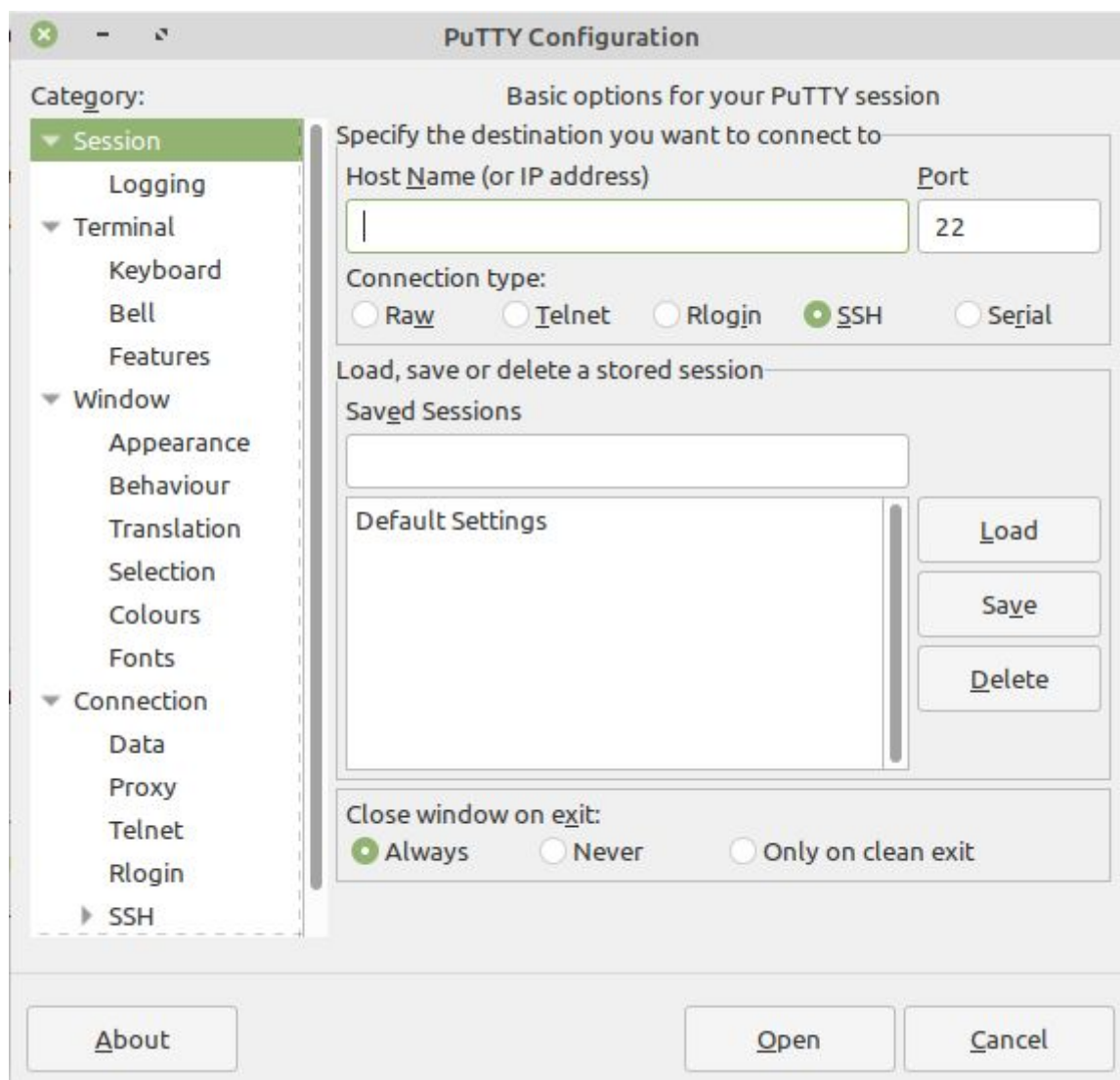
Основной пользовательский интерфейс в Linux — это терминал. Он позволяет нам вводить текстовые команды, получать ответ системы на них в текстовом виде и таким образом управлять операционной системой. После загрузки ОС нам становится доступно семь терминалов, переключаться между которыми можно, используя комбинацию клавиш.

1. В случае с физической машиной: **Ctrl + Alt + F(1–7)**, где клавиши F1–F7 — номера виртуальных терминалов
2. В случае с виртуальной машиной (под VirtualBox) переключение между терминалами будет осуществляться при помощи комбинации **host_key + F(1–7)**, где host_key в большинстве случаев — клавиша правый Ctrl.

Особый тип терминала — консоль. Это интерфейс командной строки, доступный нам после загрузки операционной системы непосредственно на мониторе нашего сервера или в окне виртуальной машины.

Для последующей работы будем использовать подключение к серверу через протокол SSH. Для этого нам понадобится установить на свой компьютер клиент SSH. Самый распространённый клиент для Windows — [PuTTY](#). Если вы работаете из-под Linux или macOS, то для подключения к удалённому серверу можно использовать предустановленный в системе клиент.

1. **Подключение к серверу, используя PuTTY.** Запускаем программу. В поле Host Name (or IP address) вводим IP-адрес нашей виртуальной машины или сервера. Далее нажимаем кнопку Open, вводим логин и пароль. **Важно:** при вводе пароля символы не отображаются.



2. **Подключение к серверу, используя terminal/iTerm.** Запускаем программу и в окне вводим команду `SSH your_user@ip_server`, далее вводим пароль и получаем приглашение в командную строку.

Навигация по файловой системе и основные операции с файлами и каталогами

В первую очередь разберёмся, что такое [путь до файла или каталога](#). Это набор символов, показывающий расположение файла или каталога в файловой системе. Путь может быть полным (абсолютным) — это путь, который указывает на одно и то же место в файловой системе, вне зависимости от текущего рабочего каталога. Полный путь всегда начинается с корневого каталога, например `/usr/local/bin/`. Путь может быть также относительным — это путь по отношению к текущему рабочему каталогу пользователя.

Команда `pwd` — `print working directory` (показать рабочий каталог) — это первая команда, с которой мы познакомимся. Она покажет текущий каталог (каталог, в котором мы сейчас находимся),

при этом покажет полный путь. Команда необходима, чтобы понять, в каком месте файловой системы мы находимся.

Перемещение между каталогами осуществляется при помощи команды **cd — change directory**. Данная команда позволит нам сменить текущую директорию, используя полный или относительный путь. Например:

- используем полный путь: `cd /usr/local/bin;`
- используем относительный путь: `cd Загрузки;`
- быстро вернуться в домашний каталог: `cd ~.`

Просмотреть содержимое каталога нам поможет команда **ls**, у этой команды есть ряд полезных параметров:

1. `ls -l` покажет подробный список содержимого, сюда будут включены дата изменения, владелец и группа владельца, права и другие свойства файлов или каталогов в директории.
2. `ls -a` покажет скрытые файлы и каталоги. В Unix-подобных системах такие файлы и каталоги начинаются с точки. Этот параметр очень часто используют в сочетании с параметром **-l**, например `ls -al /home/user`.

Копирование файлов или каталогов — команда **cp (copy): cp file1 file2**. При операции копирования можно использовать как полный, так и относительный путь. Например:

1. `cp /usr/local/etc/file /tmp/` скопирует файл с именем `file` из каталога `/usr/local/etc/` в каталог `/tmp`, сохранив название файла.
2. `cp /usr/local/etc/file /tmp/file1` скопирует файл с именем `file` из каталога `/usr/local/etc/` в каталог `/tmp`, изменив имя файла на `file1`.
3. `cp /usr/local/etc/file .` скопирует файл из каталога `/usr/local/etc/` в текущий каталог.
4. `cp file file1` создаст копию файла в текущем каталоге.
5. Копирование директорий происходит немного иначе, поскольку может содержать поддиректории, поэтому необходимо использовать параметр **-r (рекурсивно)**, например, `cp -r /dir1 .` скопирует каталог `/dir1` в текущую директорию.

Перемещение файлов или каталогов — команда **mv (move)**. `mv /home/user/file /home/user1/file` переместит файл из каталога `/home/user` в каталог `/home/user1`. Команда **mv**, применённая к файлу или каталогу в текущей директории, переименует файл или каталог. Например: `mv file1 file2`, `mv dir1 dir2`. Относительно каталогов операция **mv** не требует параметра **-r**, поскольку никак не воздействует на поддиректории.

Удаление файлов или каталогов — команда **rm (remove)**. Например, `rm file1` удалит файл. Для удаления каталогов необходимо использовать параметр **-rf (recursive, forced)** — удалить со всем содержимым, не спрашивая подтверждения.

Внимание! Операция удаления — необратимое действие. Debian-подобные дистрибутивы не спрашивают подтверждения действия. Ошибочное удаление файлов или каталогов может привести к неработоспособности системы.

Для создания файлов в ОС Linux есть несколько способов:

1. Используя утилиту **touch** — она создаст пустой файл.
2. Используя перенаправление потока вывода, например, с помощью утилит **cat** или **echo** (рассмотрим их на последующих уроках).
3. Используя текстовый редактор.

Создание каталогов — команда **mkdir** (в некоторых дистрибутивах **md**, make directory). Например, `mkdir /home/user/dir1` создаст каталог с именем `dir1` в домашнем каталоге пользователя `user`.

Бывают случаи, когда нам необходимо создать каталог и вложенные подкаталоги, для решения этой задачи используют параметр **-p (parents)**, например, `mkdir -p /home/user/dir1/dir2/` создаст в домашнем каталоге пользователя `user` каталог `dir1` и вложенный подкаталог `dir2`.

Для просмотра содержимого файлов тоже существует несколько способов:

1. Команда **cat (catenate)** позволяет быстро прочитать содержимое файла, а также склеить несколько файлов в один. Например, `cat file` выведет на экран содержимое файла с именем `file`.
2. Программы постраничного просмотра текста **less** и **more**. Основное их различие заключается в том, что **less** позволяет просмотр в обе стороны (вверх и вниз) за счёт создаваемого буфера.
3. Команда **tail** позволит вывести на экран заданное количество строк от конца файла или содержимое файла в режиме интерактивного просмотра. Например, `tail -20 file` покажет последние 20 строк файла `file`. `tail -f /var/log/syslog` непрерывно выводит на экран содержимое файла `syslog` по мере его обновления.
4. Текстовые редакторы.

Знакомство с текстовыми редакторами

Текстовый редактор vi/Vim

Внимание! В силу схожести управления редакторами vi и Vim далее в тексте рассматриваем работу с Vim.

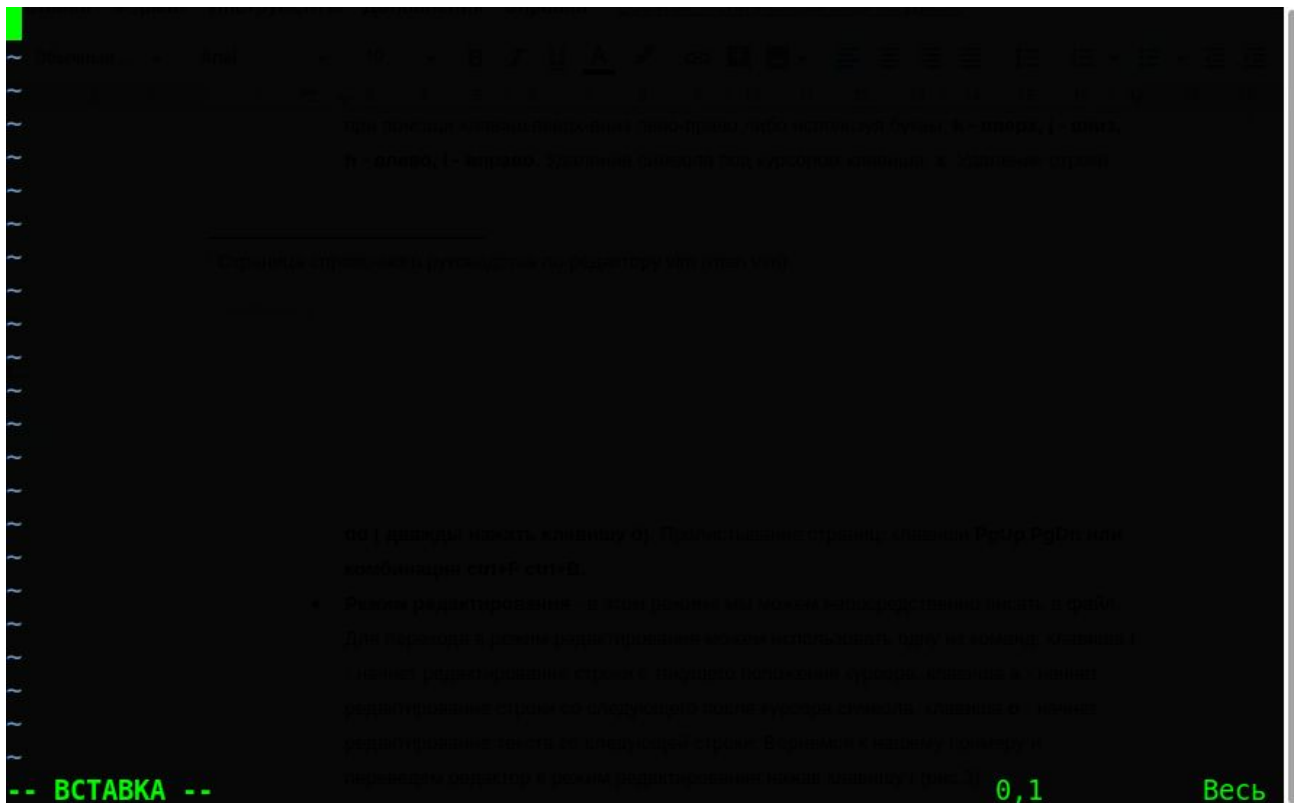
Vim не устанавливается в системе по умолчанию, его необходимо установить: `sudo apt install vim -y`.

Vim — текстовый редактор, обратно совместимый с vi. Он может быть использован для правки всех видов простого текста. По сравнению с vi, Vim имеет много усовершенствований: многократная отмена операций, множественность окон и буферов, подсветка синтаксиса, правка командной строки, автодополнение имён файлов, встроенная справка, визуальное выделение и т. п. Vim имеет большое количество плагинов, используя которые можно превратить этот редактор в довольно удобную [IDE](#). Также с редактором устанавливается программа-тренажер **vimtutor**, которая поможет почувствовать себя более уверенно в работе с данным редактором.

Запускаем редактор командой **vim myfile**:

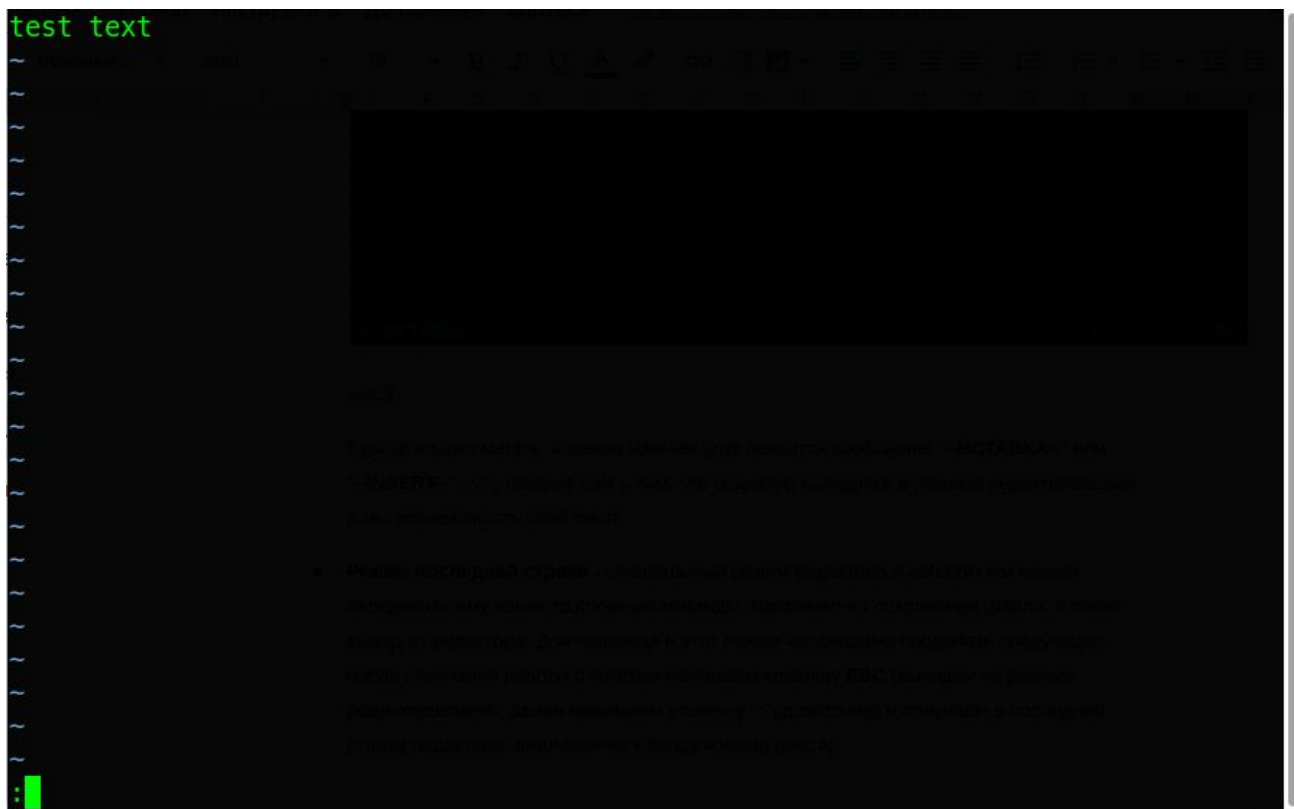
Vim/vi имеет три режима:

курсора символа, клавиша **o** начнёт редактирование текста со следующей строки. Вернёмся к нашему примеру и переведём редактор в режим редактирования, нажав клавишу **i**:

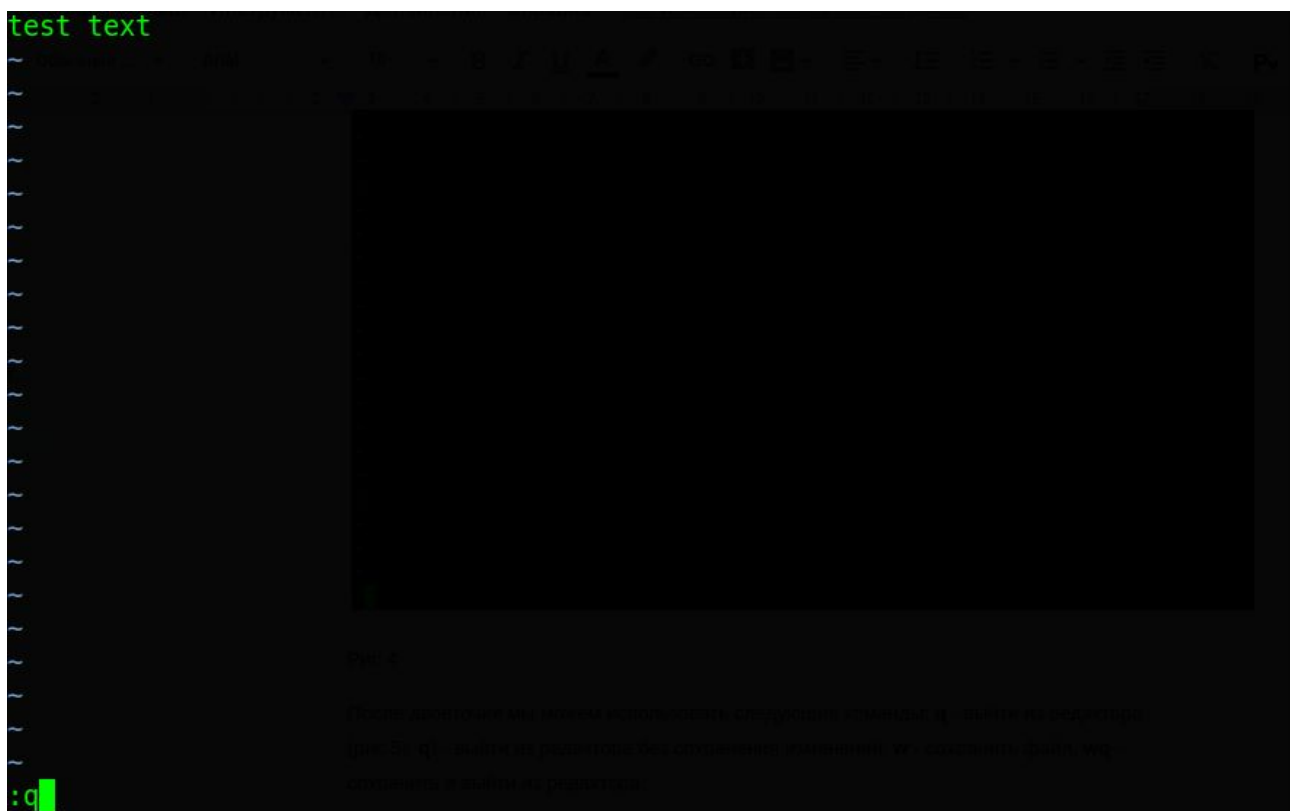


Курсор начнёт мигать, в левом нижнем углу появится сообщение **--ВСТАВКА--** или **--INSERT--**, что говорит нам о том, что редактор находится в режиме редактирования и мы можем писать свой текст.

3. **Режим последней строки** — специальный режим редактора, в котором мы можем передавать ему сложные команды, например на сохранение файла, выход из редактора. Для перехода в этот режим необходимо проделать следующее: после окончания работы с текстом нажимаем клавишу **Esc** (выходим из режима редактирования), далее нажимаем клавишу «:» (двоеточие) и получаем в последней строке редактора приглашение к вводу команд:

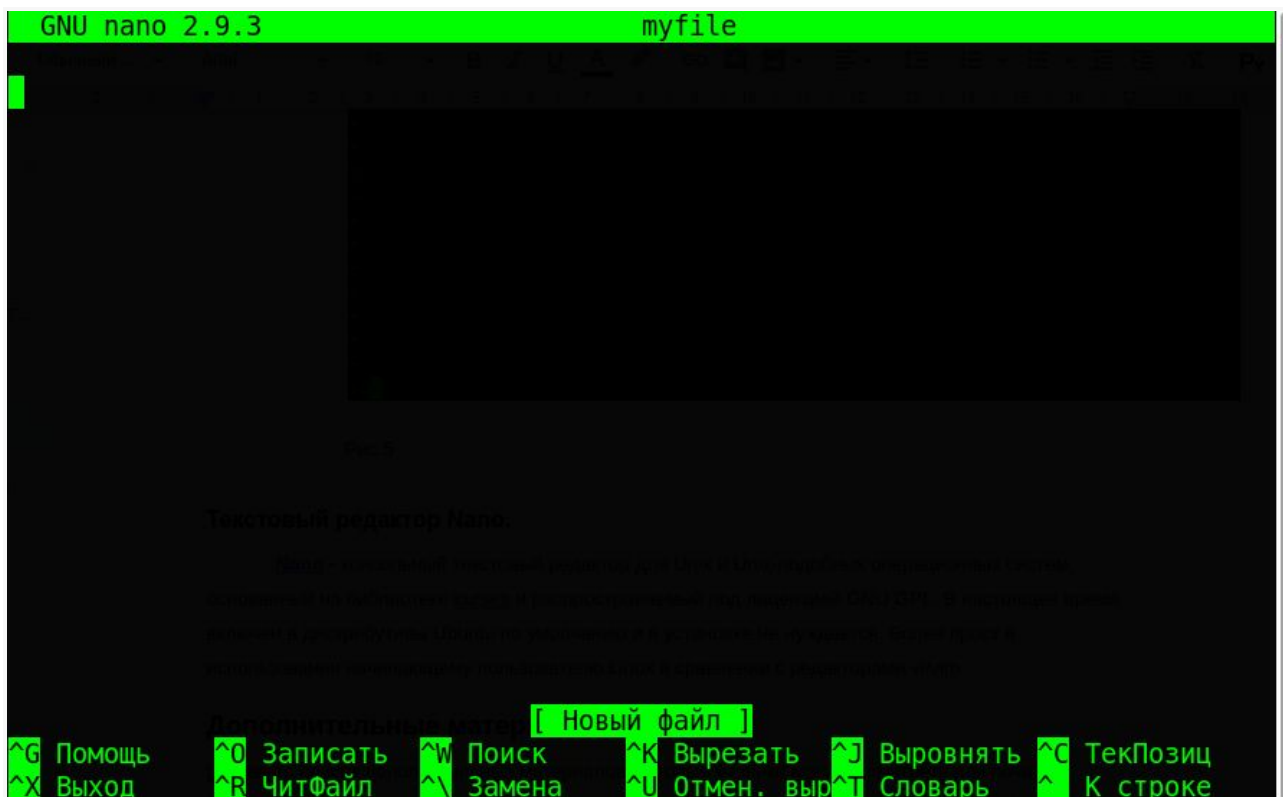


После двоеточия мы можем использовать следующие команды: **q** — выйти из редактора, **q!** — выйти из редактора без сохранения изменений, **w** — сохранить файл, **wq** — сохранить и выйти из редактора. После ввода нужной команды нажимаем клавишу **enter**. Более подробно ознакомиться с командами редактора Vim, как и говорилось выше, поможет программа **vimtutor**.



Текстовый редактор nano

[nano](#) — консольный текстовый редактор для Unix и Unix-подобных операционных систем, основанный на библиотеке curses и распространяемый под лицензией GNU GPL. В настоящее время включён в дистрибутивы Ubuntu по умолчанию и в установке не нуждается. Более прост и понятен для начинающего пользователя Linux в сравнении с редакторами vi и Vim. Запускаем редактор следующей командой: **nano myfile**.



Сразу после запуска открытый файл становится доступным для редактирования. В отличие от редактора Vim, у nano нет каких-то особых режимов работы. Ниже области редактирования располагается панель с наиболее популярными комбинациями клавиш управления редактором. ^ означает клавишу Ctrl. Сохранение файла — комбинация клавиш **Ctrl + o**, выход из редактора — **Ctrl + x**.

Практическое задание

1. **Навигация по файловой системе.** Попрактиковаться в перемещении между каталогами, используя полный и относительный путь. Перечислить, какие параметры команды `cd` позволят быстро вернуться в домашний каталог, позволят перейти на уровень выше.
2. **Управление файлами и каталогами и текстовые редакторы.** Создать файл с наполнением, используя несколько способов. Использовать разобранные текстовые редакторы для наполнения файлов данными. Создать копии созданных файлов, создать несколько каталогов с подкаталогами, перенести несколько файлов в созданные каталоги. Перечислить команды и используемые параметры команд.
3. * Используя дополнительный материал, настроить авторизацию по SSH с использованием ключей.

Дополнительные материалы

[Авторизация по ключу](#)

[Настройка авторизации по ключу, используя программу PuTTY](#)

[Небольшой обзор редакторов](#)

Используемые источники

[Статья, посвящённая работе с терминалом](#)

[Статья, посвящённая текстовым редакторам vi/Vim](#)

[Костромин В. Linux для пользователя](#)