

a1
qa



План лекции

1. Общие сведения и отличия при работе с терминалом
2. Базовые команды, работа с файловой системой, просмотр содержимого файлов
3. Установка и запуск приложений из консоли
4. Типы пользователей и их права, изменение прав
5. Работа с текстовыми редакторами в терминале

Непонятные слова и как всё это работает

ЯДРО LINUX – это главный компонент операционной системы.

УТИЛИТЫ GNU – специальные утилиты для решения различных задач, например, работы с файлами, редактирования текста, управления системой.

ДИСПЛЕЙНЫЙ СЕРВЕР (оконный менеджер) – посредник между пользователем и Linux. Его основная задача - управлять выводом на экран и принимать команды от пользователя через устройства ввода. Клиентские приложения, например, Firefox, Libreoffice используют дисплейный сервер, чтобы выводить содержимое своих окон на экран, а также чтобы получать команды от пользователя.

ГРАФИЧЕСКИЙ ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ (GUI) - компонент, который позволяет взаимодействовать с пользователем в графике и с помощью мыши, а не консольных команд. Консольный интерфейс используется в серверных редакциях Linux.

ДИСТРИБУТИВ – это совокупность программ (пакетов), некоторый набор утилит, собранных воедино на ядре системы. Ядро Linux выпускается централизованно, а дистрибутив Линукс, при определенном наличии знаний и умений, может собрать каждый желающий либо с нуля, либо на основе уже имеющегося дистрибутива.

Каждый Linux дистрибутив имеет свои цели и философию, которая и характеризует цель его создания. Например, Debian построен для широкого распространения свободного программного обеспечения, Ubuntu - для начинающих пользователей, а ArchLinux - позволяет каждому пользователю создать свою систему именно для себя.



CLI (command line interface)

CLI (command line interface)

CLI (command line interface)

Базовые Команды Linux

Базовые Команды Linux

КОМАНДЫ LINUX ДЛЯ УПРАВЛЕНИЯ ФАЙЛАМИ



Базовые Команды Linux

КОМАНДЫ LINUX ДЛЯ УПРАВЛЕНИЯ ФАЙЛАМИ



Базовые Команды Linux

КОМАНДЫ LINUX ДЛЯ УПРАВЛЕНИЯ ФАЙЛАМИ и РАБОТЫ С ТЕКСТОМ



Базовые Команды Linux

LINUX КОМАНДЫ КОНСОЛИ ДЛЯ РАБОТЫ С ТЕКСТОМ



Установка и запуск приложений из консоли

В начале был `.tar.gz`. Пользователи должны были сначала скомпилировать программы, которые они хотели использовать на своих системах GNU/Linux. Когда создавался Debian, возникла необходимость в системе управления пакетами, установленными на машине. Этой системе было дано имя `dpkg`.

Затем перед создателями GNU/Linux быстро возникла новая дилемма. Им понадобился способ для быстрого, практического и эффективного пути установки пакетов, который должен автоматически обслуживать зависимости и сохранять настроечные файлы при обновлении. Тут снова, Debian вышел вперед и предложил миру APT

Менеджер зависимостей APT (`advanced packaging tool`): используется для установки, обновления и удаления программных пакетов в операционных системах Debian и основанных на них (Ubuntu, Linux Mint и т. п.), выполняющий такие функции, как установка новых программных пакетов, обновление имеющихся пакетов, обновления индекса списка пакетов и даже обновление все системы Ubuntu.



Установка и запуск приложений из консоли

Индекс пакетов: Индекс пакетов APT по существу база данных доступных пакетов из репозитория, определенных в файле `/etc/apt/sources.list` и каталоге `/etc/apt/sources.list.d/`. Обновление индекса пакетов:

`sudo apt-get update`

Обновление пакетов: Со временем обновленные версии установленных на вашем компьютере пакетов могут стать доступными в репозиториях (например, обновления безопасности). Для обновления вашей системы сначала обновите индекс пакетов, как показано выше, а затем выполните:

`sudo apt-get upgrade`

Установка пакетов: `sources.list` готов и списки доступных пакетов обновлены, теперь только остается дать команду `apt-get`, чтобы установить предпочитаемый пакет. Например, вы можете запустить:

`sudo apt-get install zip`

APT будет искать в своей базе данных наиболее свежие версии пакетов и будет скачивать их из соответствующих архивов так, как они указаны в `sources.list`.

В случаях, когда пакет зависит от другого, APT проверит зависимости и установит необходимые пакеты.

Для установки программ из текущей папки используется следующий синтаксис команды `install`:

`sudo apt-get install ./файл_установки`



Установка и запуск приложений из консоли

ЗАПУСК ПРОГРАММ В ТЕРМИНАЛЕ

Изначально в операционных системах Unix и Linux не было графического интерфейса, поэтому программы запускались командами из терминала. Сейчас это тоже возможно и достаточно активно используется опытными пользователями. Синтаксис запуска программы выглядит таким образом:

/путь/к/файлу/программы параметры

Параметры указываются только, когда они нужны, но всегда оболочка должна знать полный путь к программе. Все что после имени программы и пробела - это параметры. Вы, наверное, уже заметили, что обычно мы не указываем полный путь при выполнении программ. Это было бы очень долго и неудобно.

Разработчики придумали обходной путь. Была создана переменная PATH, в которой хранятся все пути к папкам где обычно находятся программы - /bin, /sbin, /usr/bin, /usr/sbin и так далее. Вы можете посмотреть ее содержимое командой:

echo \$PATH

Если же вы хотите запустить программу через терминал ubuntu, которая находится в текущей папке, то ситуация будет немного другой. Система выполняет только поиск по папкам из переменной PATH, в текущей директории она не ищет. Поэтому, если вы наберете имя исполняемого файла, то получите ошибку. Нужно указывать полный путь, как вы помните [путь к текущей папке](#) будет ./:

./helloworld



Установка и запуск приложений из консоли

Иногда возникает необходимость запустить долго выполняющуюся программу в терминале так, чтобы она не мешала дальше работать. Для этого можно использовать запуск программы в фоновом режиме linux:

\$:имя_программы &

Например:

\$:dd if=/dev/zero of=~/.file count=100000 &

Система выведет PID, уникальный идентификатор программы, который вы потом можете использовать чтобы закрыть ее командой **kill**.



Пользователи, группы, права

root и sudo

В любой Linux системе обязательно есть один привилегированный пользователь — root. Этот пользователь имеет права на выполнение любых действий, удаление любых файлов и изменение любых параметров. С другой стороны, все остальные пользователи системы обычно не имеют большинства необходимых прав, например, прав на установку программ, поскольку это является административной операцией, права на которую есть только у root. Ещё одной распространённой операцией, доступной только суперпользователю, является копирование и изменение файлов в системных папках, куда обычный пользователь доступа не имеет.

Раньше данная проблема решалась достаточно просто: при обладании паролем root можно было зайти в систему под его аккаунтом либо временно получить его права, используя команду su. Потом выполнить все необходимые операции и вернуться обратно под обычного пользователя. В принципе, такая схема работает неплохо, однако у неё есть много существенных недостатков, в частности, невозможно никак (точнее, очень сложно) ограничивать административные привилегии только определённым кругом задач.

Поэтому в современных дистрибутивах Linux вместо root аккаунта для администрирования используется утилита sudo.

В Ubuntu по умолчанию root аккаунт вообще отключён, т.е. вы никаким способом не сможете попасть под root, не включив его. root именно что отключён, т.е. он присутствует в системе, под него всего лишь нельзя зайти.



Пользователи, группы, права

sudo

sudo — это утилита, предоставляющая привилегии root для выполнения административных операций в соответствии со своими настройками. Она позволяет легко контролировать доступ к важным приложениям в системе. По умолчанию, при установке Ubuntu первому пользователю (тому, который создаётся во время установки) предоставляются полные права на использование sudo. Т.е. фактически первый пользователь обладает той же свободой действий, что и root. Однако такое поведение sudo легко изменить, об этом см. ниже в пункте про настройку sudo

Для запуска в терминале команды с правами администратора просто наберите перед ней sudo:

sudo <команда>

У вас попросят ввести ваш пароль. Будьте внимательны, пароль при вводе никак не отображается, это нормально и сделано в целях безопасности, просто вводите до конца и нажимайте Enter. После ввода пароля указанная команда исполнится от имени root.

Система какое-то время помнит введённый пароль (сохраняет открытой sudo-сессию). Поэтому при последующих выполнениях sudo ввод пароля может не потребоваться. Для гарантированного прекращения сессии sudo наберите в терминале

sudo -K



Пользователи, группы, права

sudo

Иногда возникает необходимость выполнить подряд несколько команд с правами администратора. В этом случае можно временно перейти под суперпользователя командой

sudo -s

После этого вы перейдёте в режим суперпользователя (с ограничениями, наложенными через настройки sudo), о чём говорит символ # в конце приглашения командной строки. Данная команда по действию похожа на su, однако она не меняет домашний каталог на /root, что обычно очень удобно. Для того, чтобы при переходе под суперпользователя сменился так же и домашний каталог необходимо выполнить

sudo -s -H

Для выхода обратно в режим обычного пользователя наберите exit или просто нажмите Ctrl+D. Чтобы у пользователя была возможность пользоваться sudo необходимо их прописать в файл

/etc/sudoers

Чтобы дать пользователю привилегии суперпользователя, когда он вводит sudo перед командой, добавьте следующую строку:

имя_пользователя ALL=(ALL) ALL

Пользователи, группы, права

Управление пользователями

`/etc/passwd` — файл, содержащий в текстовом формате список пользовательских учётных записей (аккаунтов). Является первым и основным источником информации о правах пользователя операционной системы.

Добавление пользователя

Добавление пользователя осуществляется при помощи команды `useradd`. Пример использования:

```
sudo useradd vasyapupkin
```

Эта команда создаст в системе нового пользователя `vasyapupkin`. Чтобы изменить настройки создаваемого пользователя, вы можете использовать следующие ключи:

-b - Базовый каталог. Это каталог, в котором будет создана домашняя папка пользователя. По умолчанию `/home`

-c - Комментарий. В нем вы можете напечатать любой текст.

-d - Название домашнего каталога. По умолчанию название совпадает с именем создаваемого пользователя.

-g - Первичная группа пользователя. Можно указывать как GID, так и имя группы. Если параметр не задан будет создана новая группа название которой совпадает с именем пользователя.

Пользователи, группы, права

Управление пользователями

Изменение пользователя

Изменение параметров пользователя происходит с помощью утилиты `usermod`. Пример использования:

`sudo usermod -c "Эта команда поменяет комментарий пользователю" vasyapupkin`

`usermod` использует те же опции, что и `useradd`.

Изменение пароля

Изменить пароль пользователю можно при помощи утилиты `passwd`.

`sudo passwd vasyapupkin`

`passwd` может использоваться и обычным пользователем для смены пароля. Для этого пользователю надо ввести

`passwd`

и ввести старый и новый пароли.

Основные ключи `passwd`:

- d - Удалить пароль пользователю. После этого пароль будет пустым, и пользователь сможет входить в систему без предъявления пароля.
- e - Сделать пароль устаревшим. Это заставит пользователя изменить пароль при следующем входе в систему.
- i - Заблокировать учетную запись пользователя по прошествии указанного количества дней после устаревания пароля.
- n - Минимальное количество дней между сменами пароля.
- x - Максимальное количество дней, после которого необходимо обязательно сменить пароль.
- l - Заблокировать учетную запись пользователя.
- u - Разблокировать учетную запись пользователя.

Пользователи, группы, права

Управление пользователями

Удаление пользователя

Для того, чтобы удалить пользователя воспользуйтесь утилитой `userdel`. Пример использования:

```
sudo userdel vasyapupkin
```

`userdel` имеет всего два основных ключа:

- f - Принудительно удалить пользователя, даже если он сейчас работает в системе.
- r - Удалить домашний каталог пользователя.

Пользователи, группы, права

Управление пользователями

Управление группами

Создание группы

Программа `groupadd` создаёт новую группу согласно указанным значениям командной строки и системным значениям по умолчанию. Пример использования:

```
sudo groupadd testgroup
```

Основные ключи:

- g - Установить собственный GID.
- p - Пароль группы.
- r - Создать системную группу.

Изменение группы

Сменить название группы, ее GID или пароль можно при помощи `groupmod`. Пример:

```
sudo groupmod -n newtestgroup testgroup #Имя группы изменено с testgroup на newtestgroup
```

Опции `groupmod`:

- g -Установить другой GID.
- n - Новое имя группы.
- p - Изменить пароль группы.

Удаление группы

Удаление группы происходит так:

```
sudo groupdel testgroup
```

`groupdel` не имеет никаких дополнительных параметров.

Пользователи, группы, права

Существует три пути управления доступом к файлу или каталогу. Было определено, что каждый файл должен иметь владельца (**owner**), группового владельца (**group owner**), а также может потребоваться доступ для всех остальных пользователей (**everyone**). Эти названия обычно приводятся как пользователь/группа/остальные (**user/group/others**) или коротко **ugo**. Реализация управления доступом к файлам и каталогам в Unix позволяет или запрещает доступ по трем флагам: флаг чтения (**Read**), флаг записи (**Write**), флаг выполнения (**eXecute**). Они представляются следующим образом:

Flag user group other

rwx rwx rwx

Посмотреть права доступа на объекты можно командой **ls** с ключем **-l** («л»). Также можно добавить ключ **-a**, для того, чтобы были отображены скрытые объекты:

/media/Work/applicat > ls -l

-rwx----- 1 allexserv nogroup 638116 2009-06-25 20:42 autumnscolor.zip

drwxr-xr-x 13 allexserv nogroup 4096 2009-05-31 14:58 phpBB3

drwx----- 10 allexserv nogroup 4096 2009-06-25 14:29 phpMyAdmin-3.2.0-all-languages

-rwx----- 1 allexserv nogroup 677334 2009-06-25 20:42 pro_ubuntu.zip

drwxr-xr-x 2 allexserv nogroup 4096 2009-06-25 14:29 безымянная папка

Пользователи, группы, права

Для назначения прав используются три группы флагов, первая определяет права для владельца, вторая - права для основной группы пользователя, третья - для всех остальных пользователей в системе.

Для файлов: r - право на чтение из файла; w - разрешает запись в файл (в частности перезапись или изменение); x - позволяет исполнить файл.

Для каталогов, флаги r w x имеют несколько отличный смысл: r - позволяет читать только имена файлов в каталоге; x - позволяет иметь доступ к самим файлам и их атрибутам (но не именам); w имеет смысл только в сочетании с x, и позволяет (в дополнение к x) манипулировать с файлами в каталоге (создавать, удалять и переименовывать). w без x - не имеет никакого эффекта.

Рассмотрим таблицу, чтобы было понятнее:

ОСТ	BIN	Mask	Права на файл	Права на каталог
0	000	- - -	отсутствие прав	отсутствие прав
1	001	- - x	права на выполнение	доступ к файлам и их атрибутам_
2	010	- w -	права на запись	отсутствие прав
3	011	- w x	права на запись и выполнение	все, кроме доступа к именам файлов_
4	100	r - -	права на чтение	только чтение имен файлов
5	101	r - x	права на чтение и выполнение	чтение имен файлов и доступ файлам и их атрибутам_
6	110	r w -	права на чтение и запись	только чтение имен файлов
7	111	r w x	полные права	все права

Пользователи, группы, права

Для администрирования часто удобнее использовать не буквенное представление прав, а цифровое, в восьмеричном представлении (оно короче). Так, например, права на файл всем и вся, соответствуют записи 777 (что аналогично символьному представлению rwxrwxrwx).

Давайте рассмотрим пример, итак:

```
-rwx----- 1 allexserv nogroup 677334 2009-06-25 20:42 pro_ubuntu.zip  
drwxr-xr-x 2 allexserv nogroup 4096 2009-06-25 14:29 безымянная папка
```

Для первой строки:

Первый символ (флаг) пустой: » - » - для файлов.

Следующие три символа (rwx) обозначают права для владельца файла, в данном случае полные права для пользователя allexserv

Следующие три (- - -) - определяют права для группы nogroup, в нашем примере для всех пользователей группы nogroup доступ запрещен.

Ну и последние три символа (- - -) определяют права для всех остальных пользователей, в нашем случае доступ запрещен.

Восьмеричное обозначение прав для файла pro_ubuntu.zip: 700.

Для второй строки (это каталог, о чем свидетельствует флаг «d»), по аналогии:

Для владельца каталога allexserv - полные права (rwx).

Для группы nogroup - права на листинг каталога и доступ к файлам (r-x).

Для пользователя «все остальные» - права на листинг каталога и доступ к файлам (r-x).

Восьмеричное обозначение в этом примере: 755.

Пользователи, группы, права

chmod

Права устанавливаются командой `chmod`. Команда `chmod` поддерживает установку прав как в восьмеричном представлении, так и в символьном (маска режима доступа).

Синтаксис команды прост:

`chmod <опции> <права> <объект или регулярное выражение>`

Опции

Из самых полезных и часто используемых опций можно выделить одну:

-R - рекурсивное назначение прав. Т.е. назначить права всем объектам, руководствуясь регулярным выражением.

Например:

`chmod -R 755 *` - Назначение прав всем объектам текущего каталога, включая подкаталоги.

`chmod -R 700 z*` - Назначить полные права для владельца и исключить права для группы и всех остальных для всех объектов, которые начинаются именоваться на z, находящиеся в текущем каталоге и его подкаталогах 6)

Пользователи, группы, права

Права

Права можно записывать как в восьмеричном представлении так и в символьном. В восьмеричном представлении, для стандартных прав, указываются 3 восьмеричные цифры (1-я для владельца, 2-я для группы, 3-я для всех остальных. См. таблицу выше).

Например:

chmod 744 koszka.txt - установит права для файла koszka.txt - (r w x r - - r - -);

chmod -R 775 sobaki - установит права на каталог sobaki и на все объекты, что внутри этого каталога, включая содержимое подкаталогов (r w x r w x r - x);

chmod 700 * - установит права только для владельца на все файлы и каталоги в текущем каталоге, включая подкаталоги и их объекты (rwx - - - - -).

Другой способ назначения прав - это использование маски режима доступа (символьное представление). Помимо прав задается еще кому мы собираемся эти права выставить:

u - владельцу объекта;

g - группе объекта;

o - пользователю «все остальные»;

a - все вышеперечисленное.

Для назначения прав используются три знака: минус, плюс или равно:

- - убрать указанные права с объекта;

+ - добавить указанные права к существующим правам объекта;

= - заменить права объекта на указанные.

Пользователи, группы, права

Пример:

`chmod g+w koshki.txt` - Добавить пользователям группы файла koshki.txt права на запись в этот файл;

`chmod a=rwx sobaki.doc` - Заменит существующие права на файле sobaki.doc на полные права всем;

`chmod o-w test.cgi` - Уберет права на запись для пользователя «Все остальные».

`chmod ug=rw spisok.doc` - Выставить права на чтение и запись файлу spisok.doc для владельца и группы. Обратите внимание, что если у пользователя «все остальные» были какие-либо права, они сохранятся в неизменном виде.

Использование символического представления позволяет редактировать права файлов более гибко:

`chmod u+x,g+w-x koshki.txt` - Добавить владельцу файла koshki.txt права на его выполнение, пользователям группы разрешить запись и запретить выполнение и оставить права остальных пользователей без изменений;

`chmod u=rwx,g+w,go-x sobaki.doc` - Установить полные права для владельца файла, разрешить пользователям группы запись и запретить выполнение всем пользователям, кроме владельца файла.

Символьное назначение окажет неоценимую услугу, если требуется добавить права на объект к уже существующим правам.

Пользователи, группы, права

chown, chgrp: Изменение владельца и группы для одного или более файлов

Синтаксис команды **chown** (CHange OWNer - изменить владельца) выглядит следующим образом:

chown [опции] <пользователь[:группа]> <файл|каталог> [файл|каталог...]

Опции включают:

-R: рекурсия. Для изменения владельца для всех файлов и подкаталогов в указанном каталоге.

-v: подробный режим. Показывает все действия, выполняемые chown; сообщает, у каких файлов были изменены владельцы в результате выполнения команды, а какие файлы остались без изменений.

-c: подобна опции -v, но сообщает только о тех файлах, что были изменены.

Несколько примеров:

chown nobody /shared/book.tex: изменяет владельца файла /shared/book.tex на nobody.

chown -Rc queen:music *.mid concerts/: изменяет владельца всех файлов в текущем каталоге, чьи имена заканчиваются на .mid, и всех файлов и подкаталогов в каталоге concerts/ на пользователя queen и группу music, сообщая только о тех файлах, которые были затронуты командой.

Команда **chgrp** (CHange GRouP - изменить группу) позволяет вам изменить группу владельца для файла (или файлов); ее синтаксис очень похож на синтаксис команды chown:

chgrp [опции] <группа> <файл|каталог> [файл|каталог...]

Опции для этой команды такие же, как и у chown, и она используется очень похожим способом. Так, команда:

chgrp disk /dev/hd*

изменяет группу владельца всех файлов в каталоге /dev/ с именами, начинающимися с hd, на группу disk.

Текстовые редакторы и конвейер команд

Несмотря на то, что большинство действий в операционной системе Linux можно сделать через графический интерфейс, если вы хотите большого контроля над процессом и большей гибкости, вам придется прибегнуть к ручному редактированию конфигурационных файлов.

Для работы с конфигурационными файлами можно использовать разные редакторы. Новички, чаще всего, используют Gedit, более опытные пользователи, которые уже привыкли к терминалу предпочитают редактор nano, а многие профессионалы хотят использовать только Vim.

Gedit и OpenOffice обладают графическим интерфейсом и довольно интуитивны в управлении.

ЧТО ТАКОЕ РЕДАКТОР NANO?

Текстовый редактор Nano входит в набор программ GNU и является стандартным консольным редактором для многих дистрибутивов Linux. Разработка программы, как и многих других программ из набора GNU началась в девяностых годах.

В те времена существовала программа для отправки и приема электронной почты - Pine. Ее частью был текстовый редактор Pico, который можно было использовать для написания своих сообщений. Многим новичкам нравился Pico за его простоту и интуитивно понятный интерфейс. Но дистрибутивы Linux, в том числе Debian, не могли включить в свои репозитории закрытые бинарники для Pine и Pico.

Учитывая все это было принято решение создать редактор TIP или Tip Isn't Pico. Но потом из-за конфликта названий с другой утилитой, редактор переименовали в nano.

Главное преимущество nano в простоте использования. Работать с ним так же просто, как с OpenOffice или Gedit. Поэтому работе с программой очень просто научиться. А теперь давайте перейдем к вопросу как использовать nano.



Текстовые редакторы и конвейер команд

КАК ИСПОЛЬЗОВАТЬ РЕДАКТОР NANO?

В первую очередь нужно сказать, что nano - это консольный текстовый редактор, а это значит что его принято выполнять именно из терминала. Для этого применяется такой синтаксис:

\$ nano *опции /адрес/файла/файл*

Использовать опции командной строки не обязательно достаточно только указать имя файла, но в некоторых случаях опции могут быть очень полезны. Рассмотрим их подробнее:

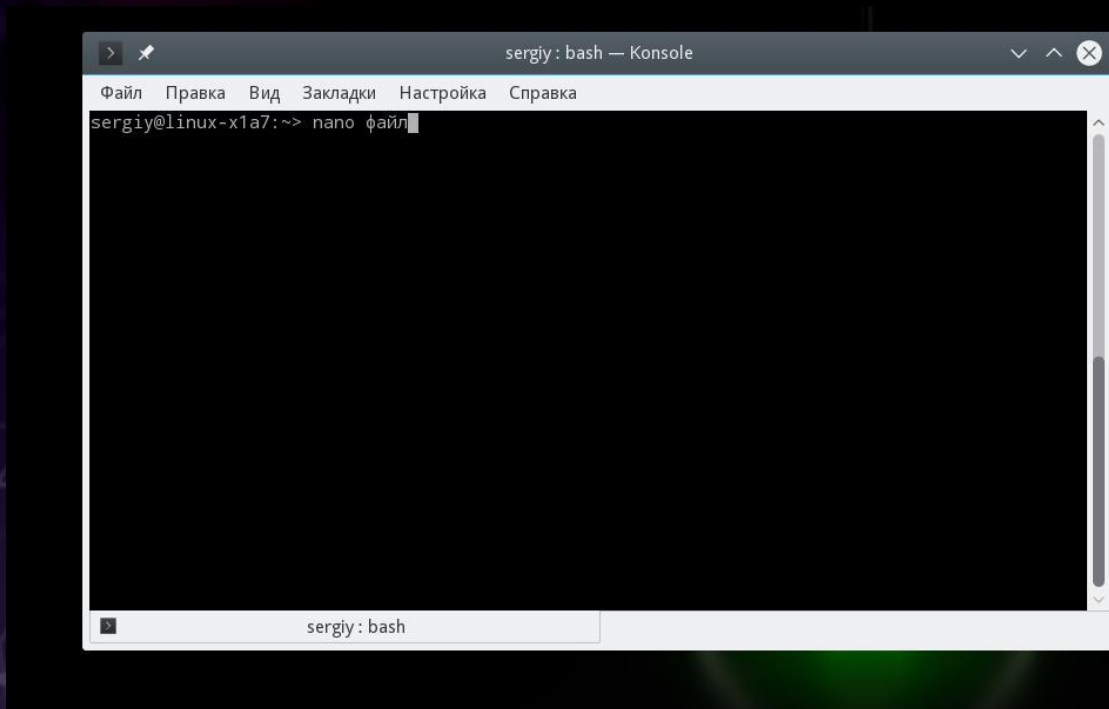
- +номер - указать строку в файле, с которой нужно начинать редактирование;
- B - создавать резервную копию для файла при сохранении;
- C - папка для резервной копии;
- D - выводить текст жирным;
- E - конвертировать табуляции в пробелы;
- F - включить поддержку редактирования нескольких файлов;
- H - сохранять историю поиска и замены;
- I - не читать конфигурацию из файлов nanorc;
- P - запоминать и восстанавливать позицию курсора во время последнего редактирования;
- T - размер табуляции в пробелах;
- m - включить поддержку мыши;
- v - режим только для чтения, вы не можете сохранить файл.

Мы рассмотрели только самые главные опции программы, естественно, что на самом деле их намного больше, но для начала этих будет вполне достаточно. Дальше мы рассмотрим как работать с самой программой ее внешний вид и сочетания клавиш.



Текстовые редакторы и конвейер команд

Чтобы открыть какой-либо файл, достаточно передать его в параметрах команде nano. Если такого файла не существует, будет создан новый. Например:

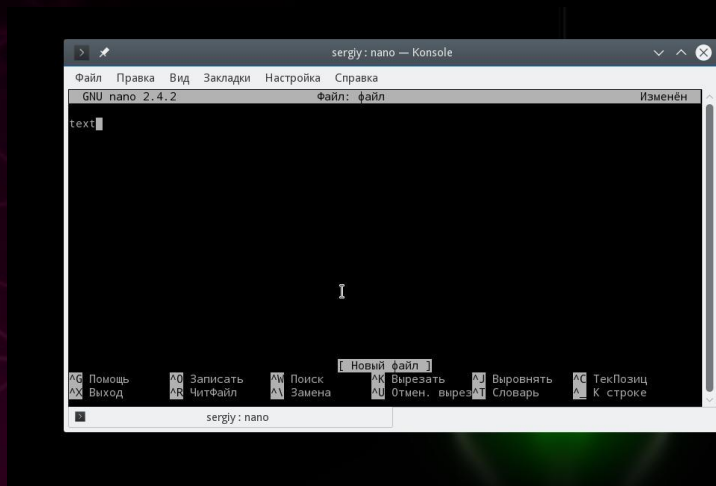


The screenshot shows a terminal window titled "sergiy : bash — Konsole". Inside the terminal, the command "nano файл" has been entered. The nano text editor interface is visible, with a menu bar at the top containing "Файл", "Правка", "Вид", "Закладки", "Настройка", and "Справка". The main editing area is currently empty, and the status bar at the bottom indicates "sergiy : bash".

Текстовые редакторы и конвейер команд

Главное окно программы выглядит очень просто, в верхней строке находится версия редактора и имя файла, а также уведомление был ли файл изменен. Дальше идет сам текст, а в самом низу опять имя файла и основные сочетания клавиш, которые вы можете использовать для работы с программой.

Для перемещения по тексту вы можете использовать мышку, стрелки на клавиатуре или специальные сочетания клавиш. В программе нет никакого меню, а все действия выполняются именно с помощью горячих клавиш. Возможно, некоторые из них будет не просто запомнить с первого раза, но после нескольких раз практики все получится. Тем более, что все самое необходимое отображается внизу экрана. Для активации того или иного действия достаточно нажать Ctrl и указанную букву. Клавиша Shift нигде не используется.



Текстовые редакторы и конвейер команд

ОСНОВНЫЕ КОМБИНАЦИИ КЛАВИШ

А теперь рассмотрим подробнее какие сочетания клавиш предлагает текстовый редактор nano linux.

Чтобы открыть более подробную информацию о сочетаниях клавиш нажмите **Ctrl+G**:

Когда завершите работу с редактором и сохраните изменения нажмите **Ctrl+X** для выхода, если файл не был сохранен, программа предложит это сделать:

Перед тем как закрывать программу, нужно сохранить изменения в файл на диске. Для этого используйте сочетание **Ctrl+O**. Если вы запустили nano без параметров, вам нужно будет ввести имя файла.

Вы можете вставить содержимое другого файла на позицию курсора, для этого нажмите **Ctrl+R** и введите имя целевого файла:

Чтобы отменить последнее действие нажмите **Alt+U**. А повторить последнее действие еще раз вы можете с помощью **Alt+E**. Для поиска по строке или регулярному выражению нажмите **Ctrl+W**, вы можете использовать стандартные регулярные выражения синтаксиса perl:

С помощью сочетания клавиш **Alt+R** вы можете использовать поиск и замену:

Мы рассмотрели основные комбинации клавиш, которые используются для управления функциями редактора. Возможно, они покажутся вам сложными, но для начала достаточно знать только **Ctrl+O** и **Ctrl+X**, все остальное придет потом.



Текстовые редакторы и конвейер команд

СОЧЕТАНИЯ КЛАВИШ ДЛЯ НАВИГАЦИИ

Давайте теперь рассмотрим горячие клавиши для быстрого перемещения по тексту в редакторе nano. Конечно, вы всегда можете использовать уже привычные клавиши **PageUP**, **PageDown**, **Home**, **End** и клавиши со стрелками, но это не совсем удобно учитывая, что у nano есть свои сочетания. Хотя они не настолько практичны как у Vim, но вы все же можете их использовать.

Для перемещения курсора вперед или назад нажимайте **Ctrl+F** и **Ctrl+B**, а для перемещения вверх/вниз по одной строке применяйте **Ctrl+P** и **Ctrl+N**.

Для перемещения в начало строки используйте **Ctrl+A**, а в конец - **Ctrl+E**. Также существуют сочетания для перемещения по страницам, **Ctrl+V** перемещает на страницу вниз, а **Ctrl+Y** - на страницу вверх. Но это еще не все. Вы можете перемещаться по словам с помощью **Ctrl+Пробел** и **Alt+Пробел**, а также нажать **Ctrl+_** и ввести номер нужной строки, а через запятую даже символ. Чтобы посмотреть где сейчас находится курсор, нажмите **Ctrl+C**.

Текстовые редакторы и конвейер команд

КОПИРОВАНИЕ ВЫРЕЗАНИЕ И ВСТАВКА

Вы можете копировать текст с помощью мышки, но это только если вы работаете в графическом интерфейсе. Но если вы используете программу в чистом терминале, такой вариант не подойдет. Для таких случаев у nano есть собственные инструменты выделения и копирования.

Чтобы выделить участок текста нажмите **Ctrl+^** переместите курсор к концу нужного участка и опять нажмите это же сочетание клавиш. Весь текст между начальной и конечной точкой не включая позицию курсора будет выделен и вы можете его скопировать или вырезать.

Для копирования выделенного текста нажмите **Alt+^**, а чтобы его вырезать - **Ctrl+K**. Чтобы вставить вырезанный текст переместите курсор на нужную позицию и нажмите **Ctrl+U**.

Чтобы вырезать всю строку текста достаточно нажать **Ctrl+K**, иногда это очень удобно при редактировании файлов конфигурации.

НЕСКОЛЬКО ФАЙЛОВ В NANO

Не все знают, но текстовый редактор nano linux поддерживает открытие нескольких файлов. Вы не можете открыть их на одном экране, но зато можете свободно переключаться между редактированием каждого из них.

Для этого передайте утилите при запуске несколько файлов:

\$: nano файл1 файл2

А затем для перемещения между ними используйте сочетания клавиш **Alt+>** и **Alt+<**. Чтобы закрыть ненужный больше файл нажмите **Ctrl+X**. Закроется не вся программа, а только активный буфер с файлом.



Текстовые редакторы и конвейер команд

КОНВЕЙЕР КОМАНД

Конвейеры чаще всего используются в shell-скриптах для связи нескольких команд путем перенаправления вывода одной команды на вход последующей, используя символ конвеера '|':

cmd1 | cmd2 | | cmdN

Конвейер обеспечивает асинхронное выполнение команд с использованием буферизации ввода/вывода. Таким образом все команды в конвейере работают параллельно, каждая в своем процессе.

Пример:

\$ grep -i "error" ./log | wc -l 43

grep выполняет регистронезависимый поиск строки "error" в файле log, но результат поиска не выводится на экран, а перенаправляется на вход команды wc, которая в свою очередь выполняет подсчет количества строк.



ДРУГИЕ ВАРИАНТЫ ИСПОЛЬЗОВАНИЯ КОНВЕЙЕРА

команда1; команда2 – команда2 выполняется после команда1 вне зависимости от результата команда1
команда1 && команда2 - команда2 выполняется после команда1, если последняя выполнена успешно
команда1 || команда2 - команда2 выполняется после команда1, если последняя выполнена с ошибкой

Пример:

```
mkdir newdir && cd newdir; cd ..; mkdir newdir || mkdir newdir1
```

- 1) создается каталог newdir
- 2) так как предыдущая команды выполнена успешно, переходим в каталог newdir
- 3) переходим на каталог выше
- 4) пытаемся создать каталог newdir - получаем ошибку "mkdir: cannot create directory `newdir': File exists"
- 5) так как предыдущая команда не выполнялась, создаем каталог newdir1

Всем спасибо за
внимание

