

---

[www.debian.org](http://www.debian.org)

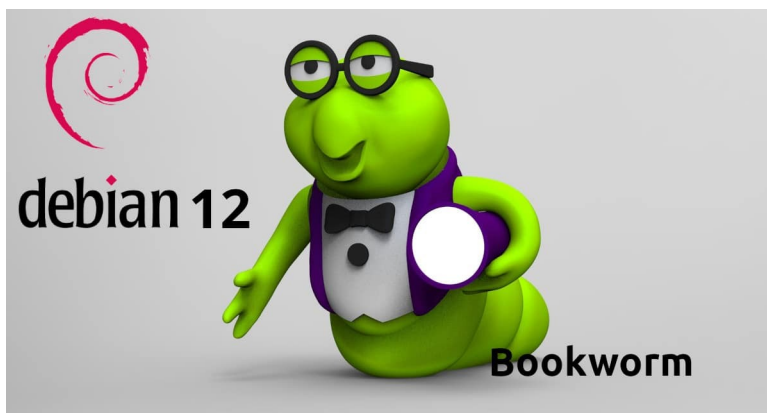
# Вариант моих настроек Linux Debian 12.1 "Bookworm" Xfce +

## Введение

В этой статье я хотел бы поделиться своим личным опытом использования операционной системы Debian 12.1 "Bookworm" с рабочим пространством Xfce. Я расскажу о своих впечатлениях и причинах, по которым я выбрал данную конфигурацию. Debian 12 "Bookworm" - это новая стабильная версия Debian, которая была выпущена после долгого периода разработки. Xfce - это легковесное и быстрое рабочее пространство, которое является одним из вариантов по умолчанию в Debian. Давайте рассмотрим подробности этой конфигурации и преимущества, которые она предлагает.

## Debian 12 "Bookworm"

**Debian 12 "Bookworm"** - это новая стабильная версия **Debian**, которая была выпущена после длительного периода разработки. Она предлагает множество новых функций, улучшений и обновлений, делая Debian еще более мощным и удобным для использования.



## Главные новые функции в Debian 12 "Bookworm"

- **Обновленное ядро**

Debian 12 "Bookworm" поставляется с обновленным ядром Linux 6.1, что обеспечивает более высокую производительность и поддержку нового аппаратного обеспечения.

- **Улучшенная поддержка Secure Boot**

В новой версии Debian 12 "Bookworm" была восстановлена поддержка Secure Boot на ARM64, что позволяет пользователям использовать эту функцию безопасности на поддерживаемом оборудовании.

- **Новые версии программного обеспечения**

Debian 12 "Bookworm" включает в себя обновленные версии различного программного обеспечения, включая Apache 2.4.57, GIMP 2.10.34, LibreOffice 7.4 и многие другие. Обновленное программное обеспечение обеспечивает улучшенную функциональность и исправление ошибок.

- **Улучшенная поддержка языков**

Debian 12 "Bookworm" предлагает больше переведенных страниц справки на разных языках, что делает его более доступным для пользователей по всему миру.

## Рабочее пространство Xfce 4.18



[www.xfce.org](http://www.xfce.org)

**Xfce** - это быстрое и легковесное рабочее пространство, которое предоставляет пользователю удобную и гибкую среду для работы с операционной системой. Оно является одним из наиболее популярных рабочих пространств в Linux-сообществе и широко используется в различных дистрибутивах, включая Debian. Xfce предлагает простой и интуитивно понятный интерфейс, который позволяет пользователям настраивать и адаптировать свою рабочую среду под свои потребности.

#### Преимущества Xfce:

- *Легковесность.*

Xfce потребляет меньше системных ресурсов по сравнению с другими рабочими пространствами, что делает его идеальным выбором для старых и медленных компьютеров. Оно обеспечивает быструю и отзывчивую работу даже на устройствах с ограниченными ресурсами.

- *Гибкость.*

Xfce предлагает широкий спектр настроек и возможностей для персонализации рабочего пространства. Пользователи могут настроить панели задач, меню, горячие клавиши и темы оформления по своему вкусу.

- *Стабильность.*

Xfce известно своей надежностью и стабильностью. Оно предлагает плавную и непрерывную работу, что особенно важно для профессионалов, которым требуется стабильная среда для работы.

#### Новые функции в Xfce 4.18

В **Xfce 4.18** представлено несколько новых функций и улучшений для рабочего пространства.

- Релиз 4.18 включает новый виджет ввода файлов и редактор ярлыков.
- Сервис **tumbler** имеет улучшения производительности и теперь поддерживает общие репозитории миниатюр.
- Панель включает новый плагин часов с различными вариантами расположения.
- Менеджер **xfdesktop** позволяет пользователям скрывать опцию "Удалить" в контекстном меню и добавляет запрос на подтверждение для перестановки значков на рабочем столе.
- Менеджер настроек **xfce4-settings** имеет упрощенную строку поиска и улучшения в настройках отображения и внешнего вида.

- **Thunar** - файловый менеджер, имеет новые функции, такие как отображение количества файлов в каталоге и даты создания файлов, а также два новых режима предварительного просмотра для изображений.
- Также добавлена возможность отменить и повторить операции с файлами и выделять файлы разными цветами.
- Панель инструментов теперь может быть настроена по желанию.

## Моя конфигурация Debian 12 "Bookworm" с Xfce

### Debian 12 "Bookworm" с Xfce

Я выбрал **Debian 12 "Bookworm"** с рабочим пространством Xfce для своей конфигурации, потому что это сочетание предлагает мне оптимальное сочетание производительности и функциональности. Xfce обеспечивает мне легкость использования и гибкость настройки рабочего пространства, а Debian 12 "Bookworm" предоставляет мне стабильность и широкий выбор программного обеспечения.

### Процесс установки

Установка Debian 12 "Bookworm" с рабочим пространством Xfce была простой и интуитивно понятной. Я загрузил установочный образ Debian 12 "Bookworm" с официального сайта Debian и записал его на USB-накопитель. Затем я перезагрузил свой компьютер с USB-накопителя и следовал инструкциям установщика.

Во время установки я выбрал рабочее пространство Xfce в качестве предпочтительного варианта. Установщик предложил мне выбрать компоненты и настройки системы, и я выбрал те, которые соответствуют моим потребностям. После завершения установки я перезагрузил компьютер и вошел в свою новую систему.

Пример настроек после сетевой установки Debian 12 в минимальном объеме с активной учетной записью root (без sudo):

### 1. Заменить репозитории в **sources.list**

Редактируем командами:

```
~$ su
```

**Password:** ввод пароля root

В открывшемся каталоге с правами root редактируем `sources.list` в nano, vim или любом текстовом редакторе:

```
~$ nano /etc/apt/sources.list
```

- Закомментируем следующую строку и сохраним файл с изменениями:

```
# deb cdrom:[Debian GNU/Linux 12.0.0 _Bookworm_ - Official amd64 DVD Binary-1 wi>
```

- При желании можно из файла удалить всё, вставив только следующие репозитории:

#### Официальные репозитории:

Стандартное содержимое конфигурационного файла **sources.list** :

```
deb http://deb.debian.org/debian/ bookworm main contrib non-free non-free-firmware>
```

```
deb-src http://deb.debian.org/debian/ bookworm main contrib non-free non-free-f>
```

```
deb http://security.debian.org/debian-security bookworm-security main contrib n>
```

```
deb-src http://security.debian.org/debian-security bookworm-security main contr>
```

```
deb http://deb.debian.org/debian/ bookworm-updates main contrib non-free non-fr>
```

```
deb-src http://deb.debian.org/debian/ bookworm-updates main contrib non-free no>
```

Мы должны заменять список репозитория в нашей системе лишь тогда, когда это действительно необходимо, в тех случаях, когда в списке репозитория появилась некорректная информация или сам файл со списком был поврежден.

#### 2. Обновляем базу данных обновленных репозитория.

```
~$ apt update
```

#### 3. Возможно потребуется установить `cut`, `nano`, `wget` (но обычно пакет уже установлен):

```
~$ apt install cut nano wget
```

#### 4. Добавление 32-битной архитектуры (для 64-битных систем):

```
~$ sudo dpkg --add-architecture i386
```

#### 5. Настраиваем **sudo** для себя как **root**:

- открываем файл `sudoers`

```
~$ nano /etc/sudoers
```

- добавляем вместо `'user'` ваше имя пользователя:

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
'user'  ALL=(ALL:ALL) ALL
```

- переходим в каталог `lightdm.conf.d`

```
~$ cd /usr/share/lightdm/lightdm.conf.d
```

- открываем файл `01_debian.conf`

```
~$ nano 01_debian.conf
```

- изменяем в нем `true` на `false`:

```
[Seat:*]
greeter-session=lightdm-greeter
greeter-hide-users=false
session-wrapper=/etc/X11/Xsession
```

- далее после сохранения устанавливаем инструмент для настройки и управления внешним видом и поведением LightDM GTK+ Greeter, который является графическим интерфейсом для входа в систему LightDM командой:

```
~$ apt install lightdm-gtk-greeter-settings
```

## 6. Настраиваем **автоматический вход** при запуске системы.

Для этого необходимо отредактировать файл `lightdm.conf`:

```
~$ nano /etc/lightdm/lightdm.conf
```

В открывшемся для редактирования `lightdm.conf`, после параметра `[Seat:*]` находим строки:

```
#autologin-user=
#autologin-user-timeout=0
```

Нужно обе строки раскомментировать и прописать свое имя пользователя в `autologin-user= ...`, при котором запланирован автовход в систему.

Далее не забываем **сохранить** исправленный конфигурационный файл.

**# Пример:**

```
autologin-user=my_name
autologin-user-timeout=0
```

## 7. Устанавливаю пакеты для Блютуз:

```
~$ sudo apt install blueman blueman bluez pulseaudio-module-bluetooth
```

8. Синхронизируем тему **GTK** с **Qt**:

```
~$ apt install qt5ct qt5-style-plugins
```

9. Добавляем пользователя в группу **sudo** (активация sudo) командами:

```
~$ su -
```

```
~$ adduser имя_пользователя sudo
```

10. Выполняем перезагрузку компьютера

```
~$ reboot
```

---

## После перезагрузки:

11. Обновляю систему компьютера уже с использованием sudo и подключенной ранее 32x битной архитектуры:

```
~$ sudo apt update && sudo apt upgrade -y
```

## 12. Установка Python3 в системе:

Теоретически в дистрибутивах Linux можно установить несколько версий Python, но по умолчанию может быть только одна версия. Настройка Python 3.11 по умолчанию требует некоторых дополнительных шагов. Следуйте дальше.

- **Запустим** команду **ls**, чтобы узнать, какие двоичные исполняемые файлы Python доступны в вашей системе:

```
~$ ls /usr/bin/python*
```

```
/usr/bin/python3      /usr/bin/python3.11-config
```

```
/usr/bin/python3.11  /usr/bin/python3-config
```

- **Проверим** предустановленную версию *Python3* в нашей системе командой:

```
~$ python3 --version
```

Python 3.11.5

- Добавим репозиторий:

```
~$ sudo add-apt-repository ppa:deadsnakes/ppa
```

- Если версия нас устраивает, то доустанавливаем полный пакет **Python3** командой:

```
~$ sudo apt install python3-full
```

- **Устанавливаем** менаджер пакетов **pip**:

```
~$ sudo apt install python3-pip
```

- **Проверяем** его установку командой:

```
~$ pip -V
```

```
pip 23.0.1 from /usr/lib/python3/dist-packages/pip (python 3.11)
```

- **Если версия предустановленного Python вас НЕ-устраивает**,  
то прежде чем сделать что-либо убедитесь, что вы знаете, какие приложения зависят от уже предустановленного в системе Python3 версии которую вы получили в результате выполнения в терминале команды:

```
~$ python3 --version
```

```
Python 3.11.5
```

- Просмотрим зависимости, используя команду:

```
~$ apt-cache rdepends python3.11
```

```
python3.11
Reverse Depends:
  libpython3.11-testsuite
  python3-uno
  virtnbdbackup
  stimfit
  python3-stfio
  python3-skorch
  rhythmbox-plugins
  python3-torchvision
  python3-torchtext
  python3-torchaudio
  python3.11-venv
  python3.11-minimal
  python3.11-full
  python3.11-doc
  python3.11-dev
  python3.11-dbg
  python3
  idle-python3.11
  idle-python3.11
  python3-all
  cluster-glue
  python3-escript-mpi
  python3-escript
  plasma-firewall
```



```
pitivi
obs-studio
liferea
python3-sbml5
python3-uno
atac
kitty
kdevelop-python
libglib2.0-tests
gedit
```

- **Устанавливаем** новую версию python 3.12:

```
~$ sudo apt install python3.12-full
```

- **Добавление** Python 3.11 и 3.12 в **Update Alternatives**:

Для добавления обеих версий Python в утилиту "update-alternatives", выполним следующие команды:

```
~$ sudo update-alternatives --install /usr/bin/python3 python3
/usr/bin/python3.11 2
~$ sudo update-alternatives --install /usr/bin/python3 python3
/usr/bin/python3.12 1
```

*Здесь:*

`/usr/bin/python3` - это символическая ссылка на текущую версию Python 3.

`/usr/bin/python3.11` и `/usr/bin/python3.12` - это пути к установленным версиям Python 3.11 и 3.12 соответственно.

`1` и `2` - это приоритеты, где более высокий приоритет означает более предпочтительную версию.

- **Выбор** новой версии Python по умолчанию:

Теперь, после выполнения вышеуказанных команд, получим возможность выбрать, какую версию Python 3 использовать, выполнив команду:

```
~$ sudo update-alternatives --config python3
```

There is 1 choice for the alternative python3 (providing /usr/bin/python3).

Selection	Path	Priority	Status
* 0	/usr/bin/python3.12	1	auto mode
1	/usr/bin/python3.11	2	manual mode
2	/usr/bin/python3.12	1	manual mode

Press <enter> to keep the current choice[\*], or type selection number:

- **Проверим** успешную установку версии Python3 по умолчанию в нашей системе командой:

```
~$ python3 -V
```

Python 3.12.2

#### Важно:

После завершения этих шагов, мы можем переключаться между **Python 3.11** и **Python 3.12** с помощью утилиты **"update-alternatives"** в зависимости от наших потребностей.

### 13. Одной командой устанавливаю необходимые мне утилиты

```
~$ sudo apt install gdebi ntfs-3g gtkhash thunar-gtkhash nautilus  
fuseiso gnome-disk-utility gnome-system-tools synaptic firmware-misc-  
nonfree curl apt-transport-https dirmngr  
ttf-mscorefonts-installer fonts-freefont-otf fonts-freefont-ttf  
fonts-noto-core rar unrar libavcodec-extra
```

Вот объяснения утилит и пакетов, перечисленных выше:

#### Утилиты

---

ntfs-3g	Это драйвер для чтения и записи файловых систем NTFS, которые часто используются в операционных системах Windows. Он позволяет пользователям Linux взаимодействовать с NTFS-дисками.
gtkhash	Это инструмент для вычисления и проверки хэш-сумм файлов. Он помогает убедиться в целостности и подлинности файлов, сравнивая их хэш-суммы.
thunar-gtkhash	Это плагин для файлового менеджера Thunar, который добавляет функциональность GtkHash для вычисления хэш-сумм файлов прямо из Thunar.
nautilus	Это файловый менеджер для графической среды GNOME. Он предоставляет удобный интерфейс для навигации по файлам и папкам.
fuseiso	Эта утилита позволяет монтировать образы оптических дисков (например, ISO-образы) в виде файловой системы. Она позволяет вам просматривать и работать с содержимым образов без их

## Утилиты

---

	фактической записи на диск.
gnome-disk-utility	Это инструмент управления дисками в среде GNOME. Он предоставляет информацию о вашем жестком диске, позволяет управлять разделами и выполнять другие операции, связанные с дисками.
gnome-system-tools	Этот набор утилит предоставляет графический интерфейс для настройки различных системных параметров в среде GNOME. Он включает в себя инструменты для управления пользователями, сетью и другими системными настройками.
synaptic	Это графический интерфейс для управления пакетами на Debian-подобных системах. Он позволяет легко устанавливать, обновлять и удалять программы и пакеты.
firmware-misc-nonfree	Этот пакет содержит несвободные (проприетарные) драйверы и прошивки для различных устройств. Он может быть полезен, если вам нужны дополнительные драйверы для работы с вашим оборудованием.
curl	Утилита командной строки для выполнения HTTP-запросов. Она используется для скачивания файлов с Интернета, отправки данных на серверы и других сетевых операций.
apt-transport-https	Этот пакет добавляет поддержку протокола HTTPS для APT (Advanced Package Tool), что обеспечивает безопасную загрузку пакетов из репозитория.
dirmngr	Утилита для управления ключами GnuPG, используемыми для проверки подписей пакетов и репозитория.
ttf-mscorefonts-installer, fonts-freefont-otf и fonts-freefont-ttf	Этот пакет устанавливает шрифты Microsoft Core Fonts, такие как Arial и Times New Roman, на вашей системе.
fonts-noto-core	Этот пакет содержит семейство шрифтов Noto, разработанных Google, и предоставляет поддержку множества языков и символьных наборов.
rar и unrar	Утилиты для создания и извлечения файлов в формате RAR,

который является одним из популярных форматов сжатия данных.

libavcodec-extra

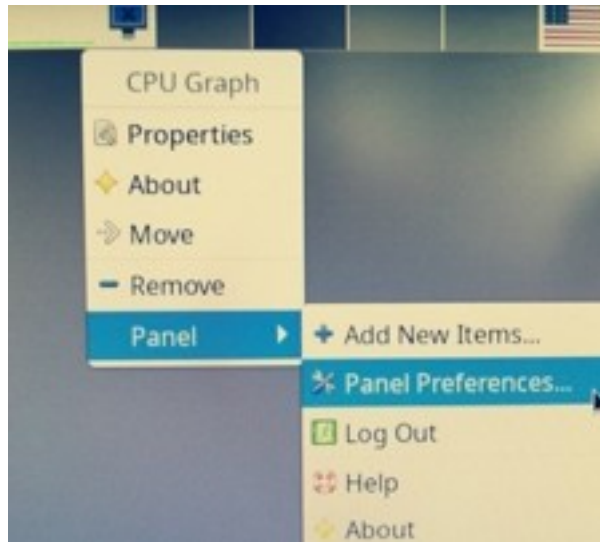
Этот пакет содержит дополнительные библиотеки для кодирования и декодирования аудио и видео файлов. Он может потребоваться для поддержки некоторых форматов мультимедиа.

### Как в XFCE добавить индикатор раскладки на панель ?

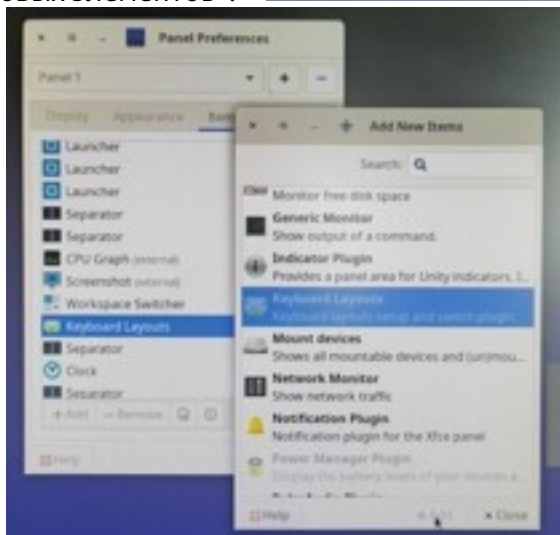
Если у вас после установки или по каким-то иным причинам(случайно удалили через *autoremove*) на панели отсутствует индикатор раскладки клавиатуры, то эта проблема решается нехитрым способом:

```
~$ sudo apt install xfce4-xkb-plugin
```

Затем правой кнопкой мыши кликайте на панели. Пункт "Панель" → "Добавить новые элементы", ищите пункт "Раскладки клавиатуры", выделяете его и жмете кнопку "Добавить". Появился флажок. Если хотите, кликните на флажке правой кнопкой мыши, выберите "Переместить" и переместите его куда угодно. Закрываем окно "Добавление"

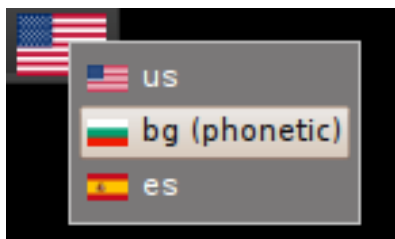


новых элементов".



**xfce4-xkb-plugin** - это плагин для управления несколькими раскладками клавиатуры

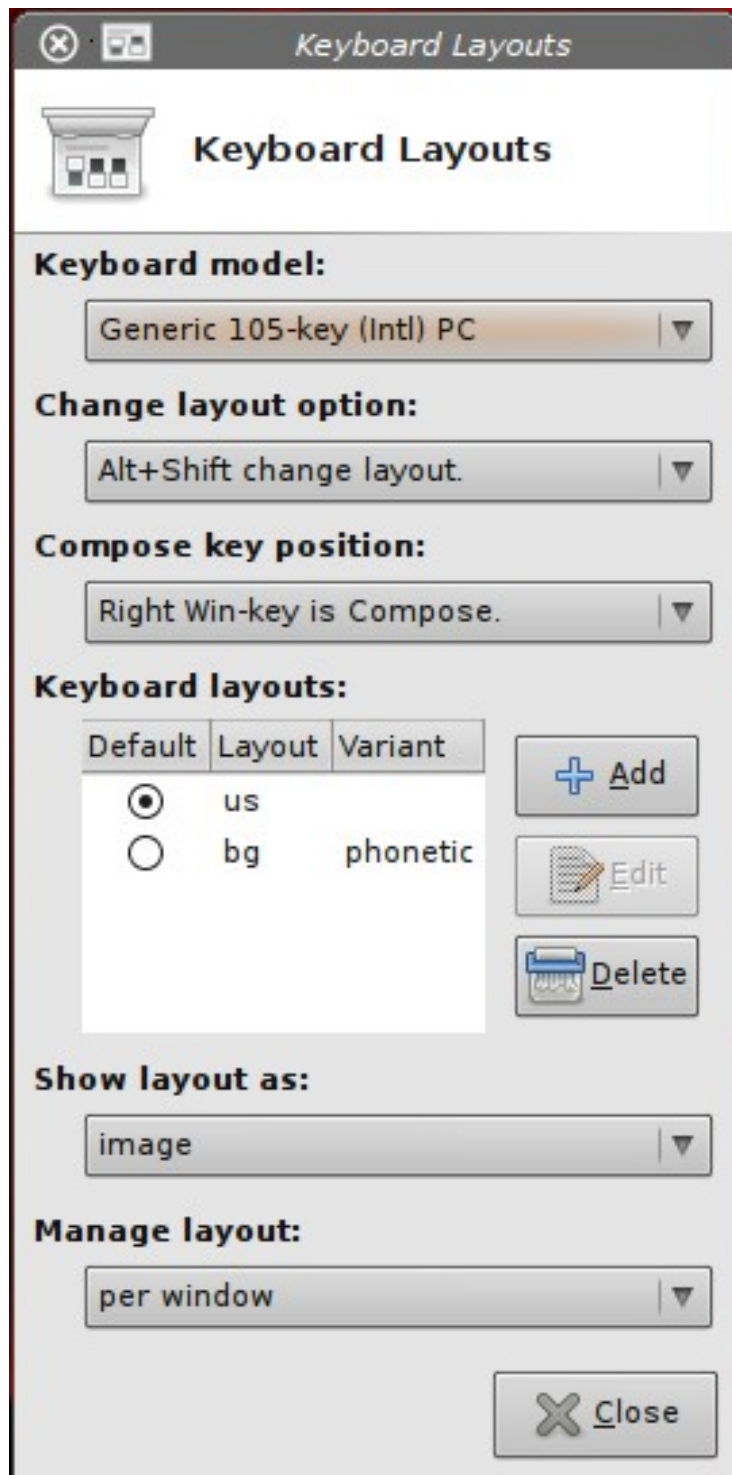
Он позволяет выбрать модель клавиатуры, комбинацию клавиш для переключения между раскладками, фактические раскладки клавиатуры, способ отображения текущей раскладки (изображение флага страны или текст) и политику раскладки, которая заключается в том, сохранять ли раскладку глобально (для всех окон), для каждого приложения или для каждого окна.



1. Установка плагина:

```
~$ sudo apt install xfce4-xkb-plugin
```

1. Выбор языка:



## 14. Установка заголовков и модулей ядра (kernel headers и kernel modules)

Эта установка может быть полезной, если вам нужно собирать и устанавливать дополнительные драйверы или программы, которые зависят от ядра Linux.

Если вы не планируете компилировать дополнительные модули ядра или устанавливать драйверы, не предоставляемые стандартными средствами Debian, то установка заголовков и модулей ядра может быть необязательной.

```
~$ sudo apt install linux-headers-$(uname -r|sed 's,[^-]*-[^-]*-,,')
```

## 15. Устанавливаю дополнительные бинарные файлы (проприетарные бинарники)

Для некоторых беспроводных адаптеров и устройств, они могут потребоваться для полной поддержки определенных аппаратных компонентов на вашем компьютере. Эти бинарные файлы включают в себя микропрограммы (firmware) для некоторых беспроводных карт и других устройств.

```
~$ sudo apt install firmware-linux-nonfree
```

## 16. Устанавливаю дополнительные инструменты для управления беспроводными сетями

Эти инструменты могут быть полезными для выполнения различных задач, связанных с настройкой и мониторингом беспроводных соединений. Например, с их помощью вы можете сканировать доступные беспроводные сети, настраивать параметры соединения, управлять беспроводными интерфейсами и многое другое.

```
~$ sudo apt install wireless-tools
```

## 17. Установлю графический интерфейс для управления сетевыми соединениями и дисками

```
~$ sudo apt install network-manager-gnome gnome-disk-utility
```

## 18. Перезагрузка!!!

```
~$ sudo reboot
```

---

## После перезагрузки настройка Xfce:

После перезагрузки **Debian 12 "Bookworm"** с рабочим пространством **Xfce**, я настраиваю его под свои нужды.

**Xfce** легка для системы и предлагает множество настроек и возможностей для персонализации моего рабочего пространства.

## 19. Устанавливаю утилиты Xfce, одной командой.

Каждый из этих компонентов предоставляет дополнительную функциональность и возможности настройки для среды рабочего стола Xfce, что делает ее более гибкой и удобной для пользователей.

Вы можете устанавливать или отключать их в зависимости от ваших потребностей.

```
~$ sudo apt install xfce4-battery-plugin xfce4-clipman xfce4-clipman-  
plugin xfce4-cpufreq-plugin xfce4-datetime-plugin xfce4-diskperf-  
plugin xfce4-fsguard-plugin  
xfce4-genmon-plugin xfce4-goodies xfce4-mount-plugin xfce4-sensors-  
plugin xfce4-smartbookmark-plugin xfce4-timer-plugin xfce4-wavelan-  
plugin xfce4-power-manager-plugins
```

*Приведу краткое описание каждого из перечисленных плагинов и утилит для среды рабочего стола Xfce:*

### Утилиты

xfce4-battery-plugin	Этот плагин предназначен для отображения информации о состоянии батареи ноутбука на панели. Он показывает уровень заряда, время работы от аккумулятора и другую информацию о батарее.
xfce4-clipman и xfce4-clipman-plugin	Эти компоненты предоставляют менеджер буфера обмена для Xfce. Они позволяют вам копировать и вставлять текст или другие данные между различными приложениями
xfce4-cpufreq-plugin	Этот плагин позволяет мониторить и управлять частотой процессора (CPU frequency scaling) на компьютере. Он может быть полезен для оптимизации производительности и управления энергопотреблением.
xfce4-datetime-plugin	Этот плагин отображает текущее время и дату на панели. Он может быть настроен на отображение различных форматов даты и времени.
xfce4-diskperf-plugin	Этот плагин предназначен для мониторинга производительности диска и отображает информацию о скорости чтения/записи данных на диске.
xfce4-fsguard-plugin	Этот плагин предоставляет мониторинг свободного места на диске и уведомляет пользователя, если свободное



пространство заканчивается.

xfce4-genmon-plugin	Этот плагин позволяет создавать пользовательские сценарии (скрипты) и отображать результат их выполнения на панели. Он полезен для мониторинга различных системных параметров.
xfce4-goodies	Этот пакет включает в себя набор различных плагинов и утилит для среды Xfce, включая рассмотренные выше и другие полезные инструменты.
xfce4-mount-plugin	Этот плагин предоставляет быстрый доступ к средствам монтирования и отмонтирования различных устройств, таких как USB-накопители и сетевые диски.
xfce4-sensors-plugin	Этот плагин позволяет мониторить информацию о датчиках температуры, напряжения и других системных параметрах на вашем компьютере.
xfce4-smartbookmark-plugin	Этот плагин предназначен для управления закладками (bookmarks) в файловом менеджере Thunar.
xfce4-timer-plugin	Этот плагин позволяет устанавливать таймеры и отсчитывать время на панели.
xfce4-wavelan-plugin	Этот плагин предназначен для мониторинга беспроводных сетей (Wi-Fi) и отображения информации о сетях на панели.
xfce4-power-manager-plugins	Этот пакет включает в себя дополнительные плагины для управления энергопотреблением и настройки энергосбережения на компьютере.

## 20. Устанавливаю утилиты для настройки и управления брандмауэром (firewall): **ufw** и **gufw**

**ufw (Uncomplicated Firewall)** - это текстовый интерфейс для настройки брандмауэра в командной строке. **ufw** упрощает управление брандмауэром, предоставляя простой способ добавления правил для разрешения или блокирования сетевого трафика.

**gufw (Graphical Uncomplicated Firewall)** - это графический интерфейс для **ufw**, который делает настройку брандмауэра более интуитивной и доступной для пользователей среды рабочего стола.

```
~$ sudo apt install ufw gufw
```

Как альтернатива можете скачать мой готовый скрипт из каталога **setup-ufw.sh** и запустить его следующим образом:

- Сохраните этот файл **setup-ufw.sh** на свой компьютер, затем сделайте его исполняемым с помощью команды:

```
~$ chmod +x setup-ufw.sh
```

- Теперь вы можете выполнить этот скрипт для настройки **ufw**, запустив его с правами суперпользователя:

```
~$ sudo ./setup-ufw.sh
```

Скрипт **setup-ufw.sh** выполнит все команды поочередно, настроит брандмауэр и включит его при каждой загрузке системы. Убедитесь, что у вас есть административные права (**sudo**) для выполнения этих команд.

## 21. Ставим утилиту для установки **.deb** пакетов

**GDebi** служит для установки **.deb**-пакетов с помощью графического интерфейса. Вы можете просто дважды щелкнуть на **.deb-файле**, и **GDebi** откроет его и предложит установить пакет, обрабатывая все зависимости автоматически.

```
~$ sudo apt install gdebi -y
```

## 22. Темы оформления и значки (иконки)

Одной из первых вещей, которую я сделал, было выбор темы оформления и значков(иконок). **Xfce** предлагает разнообразные темы оформления и значки, и я выбрал те, которые мне нравятся.

Каждый может выбрать и установить понравившуюся ему тему и значки с ресурса [www.xfce-look.org](http://www.xfce-look.org): -> [Темы](#)

**Устанавливаются все темы и иконки соответственно в папки домашней директории:**

- **~/themes** - для тем
- **~/icons** - для иконок

Как альтернатива можно установить в каталогах:

- **/usr/share/icons/** - обычно содержит предустановленные значки (общие для всех пользователей)
- **/usr/share/themes** - находятся темы оформления (общие для всех пользователей)

Теперь моя система выглядит стильно и современно.

Далее я настроил панели задач и меню. Я добавил несколько ярлыков на панель задач для быстрого доступа к моим любимым приложениям. Я также настроил меню, чтобы оно отображало только те приложения, которые я часто использую.

Теперь моя система выглядит стильно и современно. Панели задач и меню

Далее я настроил панели задач и меню. Я добавил несколько ярлыков на панель задач для быстрого доступа к моим любимым приложениям. Я также настроил меню, чтобы оно отображало только те приложения, которые я часто использую. Я настроил горячие клавиши для быстрого запуска приложений и выполнения других часто используемых команд.

Теперь я могу легко управлять своей системой с помощью горячих клавиш.

---



## 23. Установка MS Visual Studio Code for Debian

- Скопируйте скрипт **install\_vscode.sh** из репозитория на свой компьютер.
- Сделайте его исполняемым командой:

```
~$ chmod +x install_vscode.sh
```

- Затем выполните скрипт:

```
~$ ./install_vscode.sh
```

**Скрипт** выполнит все необходимые шаги для добавления репозитория **Visual Studio Code**, установки зависимостей и установки самого приложения на вашей системе **Debian**.

---



## 24. Установка из исходников!

Я всегда устанавливаю `git` из исходников, потому что в этом случае устанавливаю его последнюю версию **Git** со всеми компонентами, что делает его наиболее полным и удобным способом установки и использования Git на Debian.

1. Для установки всех зависимостей, используемых для предстоящей сборки и установки бинарных файлов Git необходимо:

```
~$ sudo apt-get install dh-autoreconf libcurl4-gnutls-dev libexpat1-dev \gettext libz-dev libssl-dev
```

2. Для того, чтобы собрать документацию в различных форматах *doc*, *html*, *info* установлю дополнительные зависимости:

```
~$ sudo apt-get install asciidoc xmlto docbook2x
```

3. Устанавливаю пакет *install-info*:

```
~$ sudo apt-get install install-info
```

4. Скачиваем последнюю версию **Git**, вы можете скачать самый свежий архив с исходниками из следующих мест:

- с сайта [Kernel.org](https://kernel.org)
- с зеркала на сайте [GitHub](https://github.com)
- Затем выполните команду в терминале указав свою версию (в моем случае это *git-2.42.0.tar.xz*):

```
~$ wget https://mirrors.edge.kernel.org/pub/software/scm/git/git-2.42.0.tar.xz
```

5. Затем компилирую и устанавливаю Git, выполняя команды в следующей очередности :

```
~$ tar -xvf git-2.42.0.tar.xz
~$ cd git-2.42.0
~$ make configure
~$ ./configure --prefix=/usr
~$ make all doc info
~$ sudo make install install-doc install-html install-info
```

6. Проверьте установку и добавьте свои записи в конфигурационный файл Git:

```
~$ sudo nano ~/.gitconfig
```

Пример содержания конфигурационного файла можно посмотреть [здесь](#)

Или можно воспользоваться более простыми записями в **.gitconfig**:

```
git config --global user.name "Simona Igls"
git config --global user.email simona@igls.io
```

- Устанавливаю **Visual Studio Code** как редактора **Git**:

```
git config --global core.editor "code --wait"
```

Добавлю его в свой файл **.gitignore** используя следующую запись:

```
# Ignore VSCode Workspace Files
*.code-workspace
```

---



## 25. Установка JupyterLab Desktop

Для моей работы я часто использую **JupyterLab Desktop**, потому что это мощное интегрированное средство разработки для анализа данных и научных вычислений.

**JupyterLab** обеспечивает удобную среду для создания и выполнения блокнотов, в которых я могу комбинировать код, текст, графику и результаты вычислений в одном документе.

**JupyterLab** также предоставляет широкий выбор языков программирования, инструментов и библиотек, что делает его идеальным инструментом для работы с данными, машинным обучением и исследованиями. Он обладает гибкой настройкой и поддержкой расширений, что позволяет мне адаптировать среду разработки под свои нужды.

Я написал скрипт `install_jupyterlab_desktop.sh` для установки **JupyterLab Desktop**:

- Скопируйте скрипт `install_jupyterlab_desktop.sh` из репозитория на свой компьютер.
- Сделайте его исполняемым командой:

```
~$ chmod +x install_jupyterlab_desktop.sh
```

- Затем выполните скрипт:

```
~$ ./install_jupyterlab_desktop.sh
```

---

## 26. Установка дополнительных программ

Необходимые мне программы я устанавливаю блоками, эти блоки можно редактировать, удаляя, или дописывая необходимые вам программы:

```
~$ sudo apt install audacity audacious bleachbit cpufrequtils clamav  
clamtk evince fbreader fluid-soundfont-gm fluid-soundfont-gs gnome-mpv  
gpick gvidm gpicview guvcview mediainfo  
mediainfo-gui mkvtoolnix usb-modeswitch net-tools gkrellm plank  
  
~$ sudo apt install puddletag qsynth simplescreenrecorder seahorse  
soundconverter timidity uget winff engrampa mpg321 vorbis-tools grub-  
customizer filezilla isomaster qshutdown  
gparted easytag cherrytree xfce4-screenshooter mtools kcolorchooser  
onboard mintstick gstreamer1.0-pulseaudio modem-manager-gui sox  
libsox-fmt-al
```

- Устанавливаю драйвера для принтеров:

```
~$ sudo apt-get install printer-driver-all  
  
~$ sudo apt-get install cups hplip
```

- Установка Google Chrome:

```
~$ cd tmp  
  
~$ sudo wget https://dl.google.com/linux/direct/google-chrome-  
stable_current_amd64.deb  
  
~$ sudo apt install ./google-chrome-stable_current_amd64.deb
```

- Можно удалить Firefox-браузер, но это по желанию:

```
~$ sudo apt remove --purge xarchiver firefox-esr  
~$ sudo apt autoremove
```

## Уменьшаю зависания выкл/вкл проблеммных процессов с 1.5 минуты на 10 секунд

- Редактирую конфигфайл с **root** правами `/etc/systemd/system.conf`:

```
~$ sudo nano /etc/systemd/system.conf
```

- Рекомендую раскомментировать и исправить в этих строках значения на **10s**:

```
DefaultTimeoutStartSec=10s  
DefaultTimeoutStopSec=10s
```

---

## 27. ZSH вместо bash

Терминал Kali Linux

Хотя терминал **Xfce** работает хорошо, но я предпочитаю внешний вид терминала **Kali Linux**, поэтому напишу как я это делаю.

## Установка **ZSH** на Debian

Оболочка **Z** - это оболочка Unix, которая была разработана как расширение к **BASH** (**Bourne shell**) в начале **90-х** годов.

**Z shell** - это интерактивная оболочка, которая включает в себя многие функции других оболочек **Unix/GNU Linux**, таких как **bash**, **fish**, **dash** и **ksh**.

1. Устанавливаю **zshell**:

```
~$ sudo apt update
```

```
~$ sudo apt install zsh
```

После установки **zsh**, запускаем команду **zsh**, чтобы переключиться с приглашения **bash** на приглашение **zsh**. Когда запустите команду, увидите приглашение настройки оболочки **Z** - выберите **0**(нулевой) вариант из предложенных в приглашении и нажмите **enter**, чтобы применить:

```
This is the Z Shell configuration function for new users,
zsh-newuser-install.
You are seeing this message because you have no zsh startup files
(the files .zshenv, .zprofile, .zshrc, .zlogin in the directory
~). This function can help you with a few settings that should
make your use of the shell easier.

You can:

(q) Quit and do nothing. The function will be run again next time.

(0) Exit, creating the file ~/.zshrc containing just a comment.
    That will prevent this function being run again.

(1) Continue to the main menu.

(2) Populate your ~/.zshrc with the configuration recommended
    by the system administrator and exit (you will need to edit
    the file by hand, if so desired).

--- Type one of the keys in parentheses ---
```



```
~$ zsh
```

1. Установка плагинов **zshell**

```
~$ sudo apt install zsh-syntax-highlighting zsh-autosuggestions
```

1. Установка **шрифтов** и **qterminal**

```
~$ sudo apt install qterminal fonts-firacode
```

1. Изменение оболочки входа по умолчанию

Воспользуюсь **chsh** - это мощный инструмент, используемый для изменения оболочки входа в систему. Нет необходимости устанавливать команду **chsh**, поскольку это стандартный пакет во всех дистрибутивах Linux.

```
~$ chsh -s /bin/zsh
```

- После запроса подтверждения пароля, необходимо выйти из системы, а затем снова войти, чтобы увидеть изменения.

Приглашение **bash** будет заменено приглашением **zsh**.

## Настройка **zshell** в **Debian**

1. Изменяем файл **.zshrc**

Файл **.zshrc** является файлом запуска, эквивалентным файлу **.bashrc** ([download](#)) для **bash**(Bourne Again shell), который используется для настройки **zshell**.

Этот файл (~/.zshrc) является скрытым файлом и размещается в домашнем каталоге.

- Удаляю старый файл

```
~$ rm ~/.zshrc
```

- Создаю новый файл **.zshrc**

```
~$ touch ~/.zshrc
```

1. После успешного создания файла **.zshrc**, открою его в текстовом редакторе **nano**

```
~$ sudo nano ~/.zshrc
```

- и вставлю в мой **.zshrc** файл следующий скрипт:

```
# ~/.zshrc file
```

```
setopt autocd          # change directory just by typing its name
#setopt correct        # auto correct mistakes
```

```

setopt interactivecomments # allow comments in interactive mode
setopt magicequalsubst     # enable filename expansion for arguments
of the form 'anything=expression'
setopt nonomatch           # hide error message if there is no match
for the pattern
setopt notify              # report the status of background jobs
immediately
setopt numericglob sort    # sort filenames numerically when it makes
sense
setopt promptsubst         # enable command substitution in prompt

WORDCHARS=${WORDCHARS//\//} # Don't consider certain characters part of
the word

# hide EOL sign ('%')
PROMPT_EOL_MARK=""

# configure key bindings
bindkey -e # emacs key bindings
bindkey ' ' magic-space # do history
expansion on space
bindkey '^U' backward-kill-line # ctrl + U
bindkey '^[[3;5~' kill-word # ctrl + Supr
bindkey '^[[3~' delete-char # delete
bindkey '^[[1;5C' forward-word # ctrl + ->
bindkey '^[[1;5D' backward-word # ctrl + <-
bindkey '^[[5~' beginning-of-buffer-or-history # page up
bindkey '^[[6~' end-of-buffer-or-history # page down
bindkey '^[[H' beginning-of-line # home
bindkey '^[[F' end-of-line # end
bindkey '^[[Z' undo # shift + tab undo
last action

# enable completion features
autoload -Uz compinit
compinit -d ~/.cache/zcompdump
zstyle ':completion:*:*:*:*:*' menu select
zstyle ':completion:*' auto-description 'specify: %d'
zstyle ':completion:*' completer _expand _complete
zstyle ':completion:*' format 'Completing %d'
zstyle ':completion:*' group-name ''
zstyle ':completion:*' list-colors ''
zstyle ':completion:*' list-prompt %SAt %p: Hit TAB for more, or the
character to insert%s
zstyle ':completion:*' matcher-list 'm:{a-zA-Z}={A-Za-z}'
zstyle ':completion:*' rehash true
zstyle ':completion:*' select-prompt %SScrolling active: current
selection at %p%s
zstyle ':completion:*' use-compctl false

```

```

zstyle ':completion:*' verbose true
zstyle ':completion*:kill:*' command 'ps -u $USER -o pid,
%cpu, tty, cputime, cmd'

# History configurations
HISTFILE=~/.zsh_history
HISTSIZE=1000
SAVEHIST=2000
setopt hist_expire_dups_first # delete duplicates first when HISTFILE
size exceeds HISTSIZE
setopt hist_ignore_dups      # ignore duplicated commands history
list
setopt hist_ignore_space    # ignore commands that start with space
setopt hist_verify          # show command with history expansion to
user before running it
#setopt share_history        # share command history data

# force zsh to show the complete history
alias history="history 0"

# configure `time` format
TIMEFMT=$'\nreal\t%E\nuser\t%U\nsys\t%S\ncpu\t%P'

# make less more friendly for non-text input files, see lesspipe(1)
#[ -x /usr/bin/lesspipe ] && eval "$(SHELL=/bin/sh lesspipe)"

# set variable identifying the chroot you work in (used in the prompt
below)
if [ -z "${debian_chroot:-}" ] && [ -r /etc/debian_chroot ]; then
    debian_chroot=$(cat /etc/debian_chroot)
fi

# set a fancy prompt (non-color, unless we know we "want" color)
case "$TERM" in
    xterm-color|*-256color) color_prompt=yes;;
esac

# uncomment for a colored prompt, if the terminal has the capability;
turned
# off by default to not distract the user: the focus in a terminal
window
# should be on the output of commands, not on the prompt
force_color_prompt=yes

if [ -n "$force_color_prompt" ]; then
    if [ -x /usr/bin/tput ] && tput setaf 1 >&/dev/null; then
        # We have color support; assume it's compliant with Ecma-48
        # (ISO/IEC-6429). (Lack of such support is extremely rare, and
        such

```

```

        # a case would tend to support setf rather than setaf.)
        color_prompt=yes
    else
        color_prompt=
    fi
fi

configure_prompt() {
    prompt_symbol=☹
    # Skull emoji for root terminal
    #[ "$EUID" -eq 0 ] && prompt_symbol=💀
    case "$PROMPT_ALTERNATIVE" in
        twoline)
            PROMPT='${F{%(#.blue.green)}}└─${debian_chroot:+
($debian_chroot)-}${VIRTUAL_ENV:+($(basename $VIRTUAL_ENV))-}(%B{%(
#.red.blue)}%n'$prompt_symbol'$m%b{%(#.blue.green)}-[%B{reset}
%(6~.-1~/.../%4~.%5~)%b{%(#.blue.green)}}\n└─%B{#.F{red}#.F{blue}
$)%b{reset} '
            # Right-side prompt with exit codes and background
            processes
            #RPPROMPT='${F{%(?... %? %F{red}%B×%b{reset}}%(1j. %j
%F{yellow}%B×%b{reset}.)}'
            ;;
        oneline)
            PROMPT='${F{%(#.blue.green)}}└─${debian_chroot:+($debian_chroot)-}${VIRTUAL_ENV:+
($(basename $VIRTUAL_ENV))-}(%B{%(#.red.blue)}%n@%m%b{reset}:%B{%(
#.blue.green)}%~%b{reset}}%(#.F{red}#.F{blue}) '
            RPPROMPT=
            ;;
        backtrack)
            PROMPT='${F{%(#.blue.green)}}└─${debian_chroot:+($debian_chroot)-}${VIRTUAL_ENV:+
($(basename $VIRTUAL_ENV))-}(%B{%(#.red.blue)}%n@%m%b{reset}:%B{%(
#.blue.green)}%~%b{reset}}%(#.F{red}#.F{blue}) '
            RPPROMPT=
            ;;
    esac
    unset prompt_symbol
}

# The following block is surrounded by two delimiters.
# These delimiters must not be modified. Thanks.
# START KALI CONFIG VARIABLES
PROMPT_ALTERNATIVE=twoline
NEWLINE_BEFORE_PROMPT=yes
# STOP KALI CONFIG VARIABLES

if [ "$color_prompt" = yes ]; then
    # override default virtualenv indicator in prompt
    VIRTUAL_ENV_DISABLE_PROMPT=1

```

configure\_prompt

```
# enable syntax-highlighting
if [ -f /usr/share/zsh-syntax-highlighting/zsh-syntax-
highlighting.zsh ]; then
    . /usr/share/zsh-syntax-highlighting/zsh-syntax-
highlighting.zsh
    ZSH_HIGHLIGHT_HIGHLIGHTERS=(main brackets pattern)
    ZSH_HIGHLIGHT_STYLES[default]=none
    ZSH_HIGHLIGHT_STYLES[unknown-token]=fg=white,underline
    ZSH_HIGHLIGHT_STYLES[reserved-word]=fg=cyan,bold
    ZSH_HIGHLIGHT_STYLES[suffix-alias]=fg=green,underline
    ZSH_HIGHLIGHT_STYLES[global-alias]=fg=green,bold
    ZSH_HIGHLIGHT_STYLES[precommand]=fg=green,underline
    ZSH_HIGHLIGHT_STYLES[commandseparator]=fg=blue,bold
    ZSH_HIGHLIGHT_STYLES[autodirectory]=fg=green,underline
    ZSH_HIGHLIGHT_STYLES[path]=bold
    ZSH_HIGHLIGHT_STYLES[path_pathseparator]=
    ZSH_HIGHLIGHT_STYLES[path_prefix_pathseparator]=
    ZSH_HIGHLIGHT_STYLES[globbing]=fg=blue,bold
    ZSH_HIGHLIGHT_STYLES[history-expansion]=fg=blue,bold
    ZSH_HIGHLIGHT_STYLES[command-substitution]=none
    ZSH_HIGHLIGHT_STYLES[command-substitution-
delimiter]=fg=magenta,bold
    ZSH_HIGHLIGHT_STYLES[process-substitution]=none
    ZSH_HIGHLIGHT_STYLES[process-substitution-
delimiter]=fg=magenta,bold
    ZSH_HIGHLIGHT_STYLES[single-hyphen-option]=fg=green
    ZSH_HIGHLIGHT_STYLES[double-hyphen-option]=fg=green
    ZSH_HIGHLIGHT_STYLES[back-quoted-argument]=none
    ZSH_HIGHLIGHT_STYLES[back-quoted-argument-
delimiter]=fg=blue,bold
    ZSH_HIGHLIGHT_STYLES[single-quoted-argument]=fg=yellow
    ZSH_HIGHLIGHT_STYLES[double-quoted-argument]=fg=yellow
    ZSH_HIGHLIGHT_STYLES[dollar-quoted-argument]=fg=yellow
    ZSH_HIGHLIGHT_STYLES[rc-quote]=fg=magenta
    ZSH_HIGHLIGHT_STYLES[dollar-double-quoted-
argument]=fg=magenta,bold
    ZSH_HIGHLIGHT_STYLES[back-double-quoted-
argument]=fg=magenta,bold
    ZSH_HIGHLIGHT_STYLES[back-dollar-quoted-
argument]=fg=magenta,bold
    ZSH_HIGHLIGHT_STYLES[assign]=none
    ZSH_HIGHLIGHT_STYLES[redirection]=fg=blue,bold
    ZSH_HIGHLIGHT_STYLES[comment]=fg=black,bold
    ZSH_HIGHLIGHT_STYLES[named-fd]=none
    ZSH_HIGHLIGHT_STYLES[numeric-fd]=none
    ZSH_HIGHLIGHT_STYLES[arg0]=fg=cyan
```

```

        ZSH_HIGHLIGHT_STYLES[bracket-error]=fg=red,bold
        ZSH_HIGHLIGHT_STYLES[bracket-level-1]=fg=blue,bold
        ZSH_HIGHLIGHT_STYLES[bracket-level-2]=fg=green,bold
        ZSH_HIGHLIGHT_STYLES[bracket-level-3]=fg=magenta,bold
        ZSH_HIGHLIGHT_STYLES[bracket-level-4]=fg=yellow,bold
        ZSH_HIGHLIGHT_STYLES[bracket-level-5]=fg=cyan,bold
        ZSH_HIGHLIGHT_STYLES[cursor-matchingbracket]=standout
    fi
else
    PROMPT='${debian_chroot:+($debian_chroot)}%n@m:%~%(#.##.$) '
fi
unset color_prompt force_color_prompt

toggle_online_prompt(){
    if [ "$PROMPT_ALTERNATIVE" = oneline ]; then
        PROMPT_ALTERNATIVE=twoline
    else
        PROMPT_ALTERNATIVE=oneline
    fi
    configure_prompt
    zle reset-prompt
}
zle -N toggle_online_prompt
bindkey ^P toggle_online_prompt

# If this is an xterm set the title to user@host:dir
case "$TERM" in
xterm*|rxvt*|Eterm|aterm|kterm|gnome*|alacritty)
    TERM_TITLE=${'\e]0;${debian_chroot:+($debian_chroot)}${VIRTUAL_ENV:
+($(basename $VIRTUAL_ENV))}%n@m: %~\a'
    ;;
*)
    ;;
esac

precmd() {
    # Print the previously configured title
    print -Pnr -- "$TERM_TITLE"

    # Print a new line before the prompt, but only if it is not the
    first line
    if [ "$NEWLINE_BEFORE_PROMPT" = yes ]; then
        if [ -z "$_NEW_LINE_BEFORE_PROMPT" ]; then
            _NEW_LINE_BEFORE_PROMPT=1
        else
            print ""
        fi
    fi
}

```

```

# enable color support of ls, less and man, and also add handy aliases
if [ -x /usr/bin/dircolors ]; then
    test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)" ||
eval "$(dircolors -b)"
    export LS_COLORS="$LS_COLORS:ow=30;44:" # fix ls color for folders
with 777 permissions

    alias ls='ls --color=auto'
    #alias dir='dir --color=auto'
    #alias vdir='vdir --color=auto'

    alias grep='grep --color=auto'
    alias fgrep='fgrep --color=auto'
    alias egrep='egrep --color=auto'
    alias diff='diff --color=auto'
    alias ip='ip --color=auto'

    export LESS_TERMCAP_mb='${E[1;31m'      # begin blink
    export LESS_TERMCAP_md='${E[1;36m'      # begin bold
    export LESS_TERMCAP_me='${E[0m'         # reset bold/blink
    export LESS_TERMCAP_so='${E[01;33m'     # begin reverse video
    export LESS_TERMCAP_se='${E[0m'         # reset reverse video
    export LESS_TERMCAP_us='${E[1;32m'     # begin underline
    export LESS_TERMCAP_ue='${E[0m'         # reset underline

    # Take advantage of $LS_COLORS for completion as well
    zstyle ':completion:*' list-colors "${(s..)LS_COLORS}"
    zstyle ':completion:*:*kill:*:processes' list-colors '=(#b) #([0-
9]#)*=0=01;31'
fi

# some more ls aliases
alias ll='ls -l'
alias la='ls -A'
alias l='ls -CF'

# enable auto-suggestions based on the history
if [ -f /usr/share/zsh-autosuggestions/zsh-autosuggestions.zsh ]; then
    . /usr/share/zsh-autosuggestions/zsh-autosuggestions.zsh
    # change suggestion color
    ZSH_AUTOSUGGEST_HIGHLIGHT_STYLE='fg=#999'
fi

# enable command-not-found if installed
if [ -f /etc/zsh_command_not_found ]; then
    . /etc/zsh_command_not_found
fi
compinit

```

- НЕ ЗАБЫВАЮ СОХРАНИТЬ мой .zshrc !!!
1. Чтобы изменения вступили в силу просто закрываю и снова открываю свой терминал.

## Загрузка цветовых схем и темы Kali Linux

1. Для начала склонируем цветовые схемы и темы **Kali Linux** из репозитория **GitHub** командой:

```
$ git clone https://github.com/linuxopsys/ubuntu-to-kali-terminal.git
```

1. После клонирования перехожу в каталог **ubuntu-to-kali-terminal** и извлекаю сжатые файлы:

```
~$ cd ubuntu-to-kali-terminal
~$ tar -xvf color-schemes.tar
~$ tar -xvf kali-dark-theme.tar
```

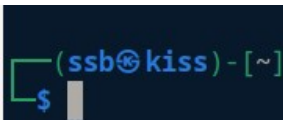
1. Удаляю каталог **qtermwidget5**, расположенный в каталоге **/usr/share**, и заменяю его каталогом из извлеченного архива tar:

```
sudo rm -rf /usr/share/qtermwidget5
sudo mv -f usr/share/qtermwidget5 /usr/share
```

## Изменение настроек Qterminal

1. Откройте настройки Qterminal: измените цветовую схему на Kali-Dark, а затем нажмите кнопку [Apply], это изменит тему терминала на Kali.
2. В нижней части настроек внешнего вида нашего терминала изменяю "Прозрачность приложения" с "0%" на "5%", так же, как в Kali, и применяю, чтобы изменения вступили в силу.

После внесения изменений ваш **qterminal** должен выглядеть следующим образом:



```
(ssb@kiss)-[~]
$
```

- если вам необходимо временно перейти на оболочку **bash**, а затем вернуться к оболочке **zsh** то выполните следующие команды в терминале
- для перехода: `~$ bash`
- для возврата:

```
~$ zsh
```



```
(ssb@kiss) - [~]  
$ bash  
ssb@kiss:~$  
ssb@kiss:~$  
ssb@kiss:~$  
ssb@kiss:~$  
ssb@kiss:~$ zsh  
(ssb@kiss) - [~]  
$
```

## 28. Устанавливаю обязательно пароль на Grub

Для защиты моего компьютера с установленным на нем дистрибутивом Linux не лишним будет поставить пароль на доступ к загрузчику GRUB.

Делается это в три шага:

### 1) Генерирую хэш пароль

Это действие проделываю при помощи утилиты **grub-mkpasswd-pbkdf2**. В Debian она по умолчанию установлена в систему. Выполняю в терминале команду:

```
~$ grub-mkpasswd-pbkdf2
```

Получаю:

```
(ssb@kiss) - [~]  
$ bash  
ssb@kiss:~$ grub-mkpasswd-pbkdf2  
Enter password:  
Reenter password:  
PBKDF2 hash of your password is grub.pbkdf2.sha512.10000.FBA28D4B4E98ED1C646FA62  
1489F26B4470144B8676D2669829F27A476D3071461272E55A3CB34392299AFED715693B7C9F9616  
ADA4EA2F620D6E4A0EB0E1B35.10230ABC69D2B22E5D6AEF4D4220A07446CBB1498EAF381B2BAFE5  
2871E685ACF923B2A2E8513F807CCD47A772D9AE12ACF3350AA3D0B864C769B6C67229F9F5  
ssb@kiss:~$
```

Полученный **хэш** – это длинная строка, которая начинается с **grub.pbkdf2.....**

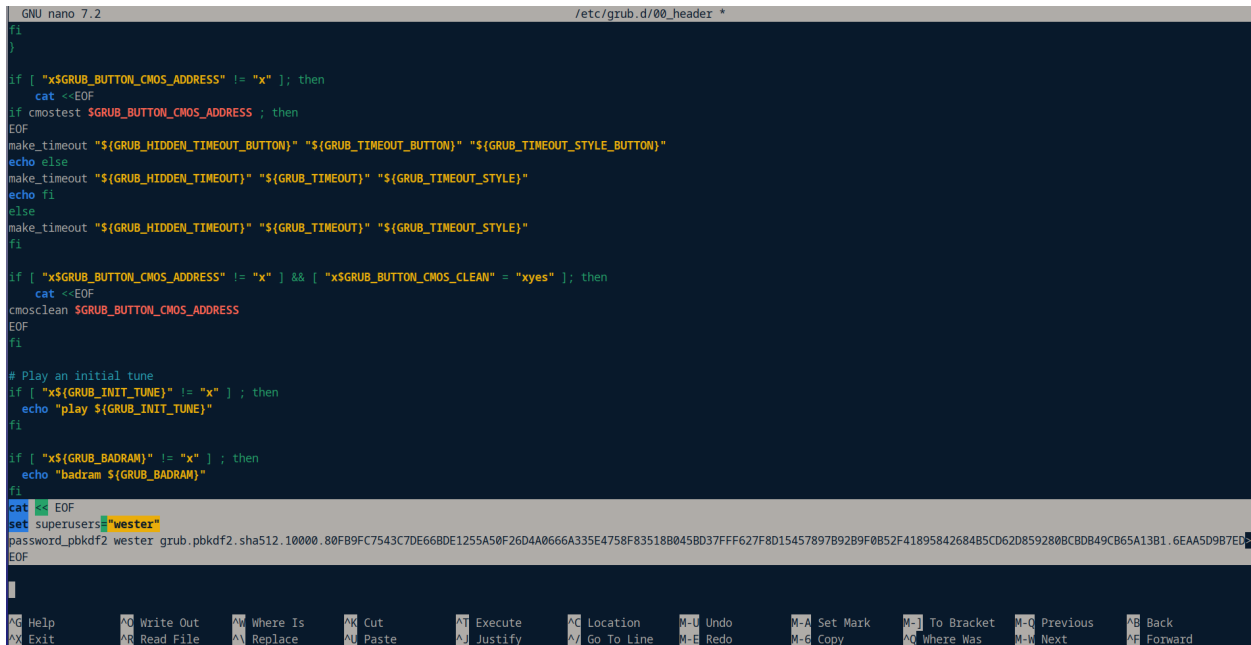
Копирую ее в текстовый редактор и сохраняю, в отдельный файл-документ на диске.

## 2) Назначаю супер-пользователя для GRUB

- Сначала открываю в текстовом редакторе файл-документ, куда сохранил свой **хэш**, и вставляю свой **хэш** дописывая недостающие строчки, чтобы скрипт получился следующего вида:
- **ваш\_name** - во второй и третьей строке изменяю на моё имя *супер-пользователя GRUB*, например на **wester** получаю:
- Затем выполняю настройку файла **00\_header** расположенного в **/etc/grub.d/**. Открываю его в текстовом редакторе и редактирую его содержимое следующим образом.

```
~$ sudo nano /etc/grub.d/00_header
```

- И вставляю мой приготовленный заранее **скрипт** в самый конец открытого в текстовом редакторе файла **00\_header** вот так:



```
GNU nano 7.2 /etc/grub.d/00_header *
fi

if [ "$GRUB_BUTTON_CMOS_ADDRESS" != "x" ]; then
cat <<EOF
if cmosclean $GRUB_BUTTON_CMOS_ADDRESS ; then
EOF
make_timeout "${GRUB_HIDDEN_TIMEOUT_BUTTON}" "${GRUB_TIMEOUT_BUTTON}" "${GRUB_TIMEOUT_STYLE_BUTTON}"
echo else
make_timeout "${GRUB_HIDDEN_TIMEOUT}" "${GRUB_TIMEOUT}" "${GRUB_TIMEOUT_STYLE}"
echo fi
else
make_timeout "${GRUB_HIDDEN_TIMEOUT}" "${GRUB_TIMEOUT}" "${GRUB_TIMEOUT_STYLE}"
fi

if [ "$GRUB_BUTTON_CMOS_ADDRESS" != "x" ] && [ "$GRUB_BUTTON_CMOS_CLEAN" = "xyes" ]; then
cat <<EOF
cmosclean $GRUB_BUTTON_CMOS_ADDRESS
EOF
fi

# Play an initial tune
if [ "$GRUB_INIT_TUNE" != "x" ]; then
echo "play ${GRUB_INIT_TUNE}"
fi

if [ "$GRUB_BADRAM" != "x" ]; then
echo "badram ${GRUB_BADRAM}"
fi
cat << EOF
set superusers:"wester"
password_pbkdf2 wester grub.pbkdf2.sha512.10000.80FB9FC7543C7DE668DE1255A50F26D4A0666A335E4758F83518B045BD37FF627F8D15457897B92B9F0B52F41895842684B5CD62D859280BCBDB49CB65A13B1.6EAA5D9B7ED
EOF
```

## 3) Обновляю конфигурацию GRUB

Выполняю это для применения ранее внесенных изменений при помощи команды в терминале:

```
~$ sudo update-grub
```

Получаю:

```
(ssb@kiss) - [~]  
$ sudo update-grub  
Generating grub configuration file ...  
using custom appearance settings  
Found background image: /root/.images/stock-abstract-stock-vivo-nex-dark-4512x21  
60-467 (1).jpg  
Found linux image: /boot/vmlinuz-6.1.0-12-amd64  
Found initrd image: /boot/initrd.img-6.1.0-12-amd64  
Found linux image: /boot/vmlinuz-6.1.0-10-amd64  
Found initrd image: /boot/initrd.img-6.1.0-10-amd64  
Warning: os-prober will be executed to detect other bootable partitions.  
Its output will be used to detect bootable binaries on them and create new boot  
entries.  
done  
  
(ssb@kiss) - [~]  
$
```

Выполняю перезагрузку и получаю приветственный экран:

```
Enter username:  
western  
Enter password:
```

## Заключительное слово

Моя конфигурация **Debian 12 "Bookworm"** с рабочим пространством **Xfce** предоставляет мне отличную комбинацию производительности и функциональности, но это моя личная конфигурация, обусловленная моими привычками, моими потребностями и симпатиями.

Каждый из вас может иметь свой Debian 12 "Bookworm" и свою конфигурацию.

Дайте **Debian 12 "Bookworm"** с **Xfce** шанс и вы не пожалеете о своем выборе! Буду рад принять ваши замечания и предложения!

## Authors

[Siergej Sobolewski](#)

## Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2023-09-15	0.2	Siergej	Update Lab to Git, Bash
2023-09-08	0.1	Siergej	Created JupyterLab

© 2023. All rights reserved.