

Справочник по настройке и оптимизации GNU/Linux с нуля



© Кучеренко Вадим Сергеевич

Оглавление

| | |
|---|----|
| Zram или Zswap?..... | 9 |
| Установка..... | 9 |
| Preload и Prelink..... | 13 |
| Увеличение отзывчивости системы в пиковые нагрузки..... | 14 |
| Оптимизация дисковых операций..... | 16 |
| Tmpfs..... | 16 |
| Команды Linux на все случаи жизни!..... | 41 |
| Команды Linux для навигации в терминале..... | 42 |
| Команды Linux для получения прав суперпользователя..... | 43 |
| Команды Linux для управления пакетным менеджером..... | 44 |
| apt (Debian/Ubuntu/Mint)..... | 45 |
| Команды Linux для управления процессами..... | 46 |
| Команды Linux для управления файлами..... | 47 |
| Просмотр и изменение файлов..... | 47 |
| Создание и удаление файлов и папок..... | 48 |
| Копирование и перемещение файлов..... | 49 |
| Поиск файлов..... | 49 |
| Команды Linux для работы с разделами..... | 49 |
| Команды Linux для управления системой..... | 51 |
| Команды Linux для управления пользователями..... | 52 |
| Команды Linux для управления сетью..... | 52 |
| Устраняем тормоза системы при операциях ввода-вывода..... | 53 |
| Браузер..... | 59 |
| ФОН РАБОЧЕГО СТОЛА..... | 63 |
| РАСКЛАДКА КЛАВИАТУРЫ..... | 64 |
| СЕТЕВЫЕ АККАУНТЫ GNOME..... | 68 |
| ТЕМА GNOME..... | 70 |
| РАСШИРЕНИЯ ОБОЛОЧКИ GNOME..... | 72 |
| ШРИФТЫ GNOME..... | 74 |
| ЗНАЧКИ РАБОЧЕГО СТОЛА..... | 75 |

Debian 8



Настройка после установки Debian 8 "Jessie"

Свеже установленный Debian нуждается в небольшой доводке до ума. Начнём. Для начала нужно установить `sudo` - утилиту, с помощью которой можно давать права администратора программам, при необходимости (обновление системы, установка/удаление программ, административная работа и т.д):
`su` ваш пароль `root apt install sudo adduser имя пользователя sudo`

Подключим дополнительные репозитории программ:

`sudo nano /etc/apt/sources.list`

В каждой строчке, после слова `main`, пишем `contrib non-free`.

contrib - свободное ПО, которое имеет зависимость от несвободного. **non-free** - несвободное ПО и ПО ограниченное патентами. Например проприетарные драйверы.

Подключим репозиторий **Backports**. В нём находятся свежие версии некоторого ПО (например ядер, драйверов и многого другого), которое достаточно стабильное для **Debian Stable**, но не успевшее попасть в финальный выпуск.

Пакеты в данном репозитории переносятся из тестовой ветки Debian и адаптируются для использования в стабильной, без нарушения нормальной работы системы

Backports

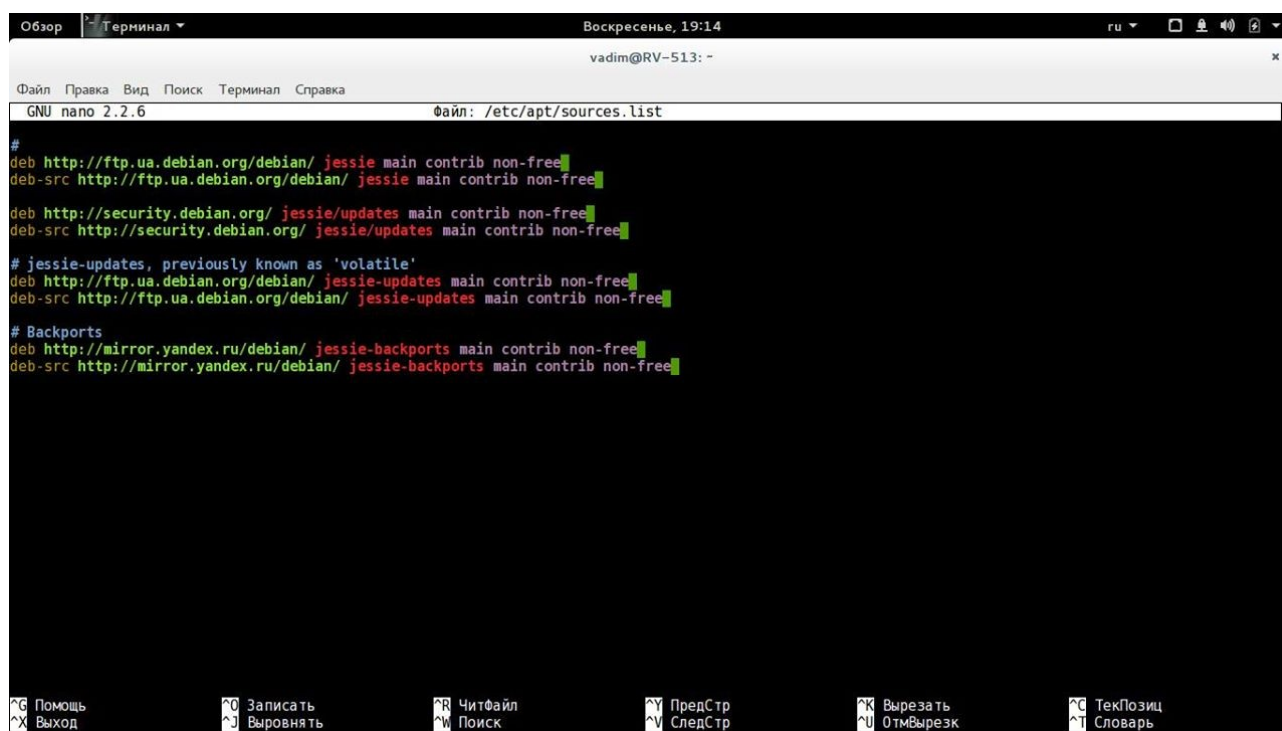
```
deb http://mirror.yandex.ru/debian/ jessie-backports main contrib non-free
```

```
deb-src http://mirror.yandex.ru/debian/ jessie-backports main contrib non-free
```

Пакеты из репозитория **backports** устанавливаются командой:

```
sudo apt install -t jessie-backports имя пакета
```

Когда все нужные репозитории прописаны, сохраняем (Ctrl+O, Enter) и выходим из редактора (Ctrl+X).



```
Обзор Терминал Воскресенье, 19:14 ru
vadim@RV-513: ~
GNU nano 2.2.6 Файл: /etc/apt/sources.list

#
deb http://ftp.ua.debian.org/debian/ jessie main contrib non-free
deb-src http://ftp.ua.debian.org/debian/ jessie main contrib non-free

deb http://security.debian.org/ jessie/updates main contrib non-free
deb-src http://security.debian.org/ jessie/updates main contrib non-free

# jessie-updates, previously known as 'volatile'
deb http://ftp.ua.debian.org/debian/ jessie-updates main contrib non-free
deb-src http://ftp.ua.debian.org/debian/ jessie-updates main contrib non-free

# Backports
deb http://mirror.yandex.ru/debian/ jessie-backports main contrib non-free
deb-src http://mirror.yandex.ru/debian/ jessie-backports main contrib non-free
```

Теперь осталось обновить список пакетов.

```
sudo apt-get update && sudo apt-get upgrade
```

Теперь установим драйвера, а также различные мультимедиа пакеты, расширяющие функционал системы. Начнём с драйверов:

```
sudo apt install firmware-linux-free firmware-linux-nonfree
```

Для обладателей видеокарт **AMD Radeon**, рекомендуется использовать свободный драйвер (модуль которого ставится с пактом **firmware-linux-free**), так как в отличие от фирменного проприетарного, он обладает гораздо более высокой стабильностью, а по производительности не сильно уступает проприетарному. Плюс поддерживает ряд фиш, недоступных проприетарному.

Для обладателей видеокарт **Nvidia**. Здесь ситуация обратная. Свободный драйвер сильно ограничен по функциональности и производительности (хотя в последних версиях он стал весьма неплох, и если вам не нужны игры и крутые графические эффекты - можете смело его использовать). Напротив, проприетарный драйвер обладает всеми достоинствами, поддержкой всех фирменных технологий Nvidia и замечательной производительностью в играх (которая сравнима с **Windows**, а порой и сильно превышает её). Устанавливаем:

```
sudo apt install nvidia-glx nvidia-kernel-`uname -r` nvidia-kernel-dkms nvidia-settings nvidia-xconfig
```

После установки выполните `sudo nvidia-xconfig` и перезагрузитесь.

Далее рекомендуется поставить драйверы на сетевую карту (особенно если это **wi-fi**). Для карт на чипах **Realtek**, это пакет **firmware-realtek**, для **Ralink** - **firmware-ralink**. Узнать модель вашей сетевой карты можно командой **lspci** (искать строки **Ethernet** и/или **Network**).

Теперь установим расширенную поддержку мультимедиа. Копируем следующую монструозную команду:

```
sudo apt install flashplugin-nonfree ttf-mscorefonts-installer freepats gstreamer0.10-plugins-base gstreamer0.10-plugins-good gstreamer1.0-fluendo-mp3 gstreamer0.10-fluendo-mp3 gstreamer0.10-plugins-ugly gstreamer0.10-pulseaudio gstreamer1.0-pulseaudio unrar gstreamer1.0-plugins-base gstreamer1.0-plugins-good gstreamer1.0-plugins-ugly cabextract gstreamer1.0-libav
```

Flash Player, шрифты Microsoft TTF (многие сайты без них выглядят вырвиглазно), плагины для GStreamer, улучшенная поддержка mp3, h.264 и распаковщик RAR-архивов. Всё это мы только что установили.

Кстати о шрифтах. Вы наверно заметили, что искоробочные шрифты в Debian выглядят не очень (хотя конечно дело вкуса). Я использую шрифт **Droid Sans**,

который довольно хорошо выглядит и не напрягает глаза. Вы же можете выбрать любой. Но хороший шрифт - половина дела. Нужно правильно настроить его сглаживание. Со шрифтами кроме Droid Sans, следующие настройки я не проверял, так что будьте внимательны (по умолчанию в Debian используется шрифт **DejaVu Sans**). Ну ладно, ближе к делу. В вашей домашней директории, включите показ скрытых файлов, и зайдите в директорию **.config/fontconfig**. В ней создайте файл **fonts.conf** со следующим содержимым:

```
<?xml version="1.0"?> <!DOCTYPE fontconfig SYSTEM "fonts.dtd"> <fontconfig> <match target="font">
<edit mode="assign" name="hinting"> <bool>true</bool> </edit> </match> <match target="font"> <edit
mode="assign" name="hintstyle"> <const>hintslight</const> </edit> </match> <match target="font"> <edit
mode="assign" name="rgba"> <const>rgb</const> </edit> </match> <match target="font"> <edit mode="assign"
name="antialias"> <bool>true</bool> </edit> </match> <match target="font"> <edit mode="assign"
name="lcdfilter"> <const>lcddefault</const> </edit> </match> </fontconfig>
```

Далее в самой домашней директории, создайте файл **.Xresources** со следующим содержимым:

```
Xft.autohint: 0 Xft.lcdfilter: lcddefault Xft.hintstyle: hintslight Xft.hinting: 1 Xft.antialias: 1 Xft.rgba: rgb Xft.dpi:
96
```

Обратите внимание на строку **Xft.dpi: 96**. Здесь указан DPI (количество точек на дюйм) данного монитора. У меня это 96. Узнать ваше значение можно командой `xdpinfo | grep resolution`

Далее набираем `sudo dpkg-reconfigure fontconfig-config`

На первой вкладке выбираем **Autohinter**, на второй **"Всегда"**, на третьей **"Нет"** Применяем: `sudo dpkg-reconfigure fontconfig`

Теперь выполните команду `xrdb -merge ~/.Xresources` и перезайдите в систему.

Шрифт станет значительно приятнее.

Настроим использование раздела подкачки - параметр **swappiness**. Значение **swappiness**, варьирующееся от 0 до 100, является степенью, при которой система поддерживает анонимную память или кеш страниц. Высокое значение повышает производительность файловой системы, при агрессивной подкачке наименее активных процессов. Низкое значение позволяет избежать подкачки процессов из памяти, что обычно снижает время ожидания за счет

производительности ввода-вывода. Значение по умолчанию равно 60.
Открываем файл:

```
sudo nano /etc/sysctl.conf
```

и в конец пишем **vm.swappiness = 5**

сохраняем и применяем:

```
sudo sysctl -p
```

Внимание: при значении **vm.swappiness = 0** подкачка будет полностью отключена, что может привести к запуску **OOM Killer** при переполнении оперативной памяти.

Прочие полезные приложения:

GDebi (установщик deb-пакетов): `sudo apt install gdebi`

Synaptic (менеджер пакетов): `sudo apt install synaptic apt-xapian-index`

p7zip (7z архиватор): `sudo apt install p7zip`

Gparted (редактор дисковых разделов): `sudo apt install gparted`

NTFS-config (утилита для работы с дисками в файловой системе NTFS):

```
sudo apt install ntfs-config
```

Это всё что выше одной командой)))

```
sudo apt install gdebi apt-xapian-index p7zip gparted ntfs-config
```

Zram и Zswap или как увеличить эффективность оперативной памяти



ZSWAP - модуль ядра Linux, доступный с версии 3.11. Отличается от ZRAM тем, что использует существующий swap-раздел на диске, а в ОЗУ создаётся пул со сжатыми данными (кэшем). После того как пул до отказа забьётся сжатыми данными, он сбросит их в раздел подкачки и снова начнёт принимать и сжимать данные. Размер пула можно указать вручную, по умолчанию он динамический (то есть будет использовать всю доступную оперативку).

Реализация такого подхода позволяет, при возникновении необходимости сброса памяти в раздел подкачки, сократить ввод-вывод и повысить скорость работы системы в целом, за счет того, что по возможности избегается использование медленного носителя. Ценой сокращения ввода/вывода является увеличение нагрузки на процессор, который тратит дополнительные ресурсы на сжатие и распаковку данных.

По утверждению разработчиков, в их конфигурации при компиляции ядра в ситуации когда происходит своппинг, **выигрыш по объему ввода/вывода составил 76%**, а время выполнения операции **сократилось на 53%**.

При использовании ZSWAP, используется раздел swap на диске, в ОЗУ хранится только сжатый кэш. Можно считать ZSWAP продвинутым вариантом ZRAM. Данная статья будет актуальна для владельцев компьютеров с малым количеством оперативки, людей, которые запускают много ресурсо-ёмких приложений, которые в свою очередь потребляют много памяти и так далее. В **ядре Linux** не так давно появились две замечательные технологии - **zram** и **zswap**. Опишу что это и для чего:

ZRAM - модуль ядра Linux, ранее известный как **compcache**. До версии **ядра 3.14** находился в экспериментальной ветке, с 3.14 перемещён в основную. Суть его в том, что в оперативной памяти создаётся сжатый раздел подкачки (**swap**). Создавая swap в ОЗУ, мы тем самым хоть и уменьшаем объем доступной оперативной памяти, но тем не менее информация в оперативной памяти всегда

хранится в несжатом виде, а при использовании ZRAM происходит следующее: как только системе начинает не хватать оперативной памяти, она начинает активно занимать swap, а так как swap у нас в оперативной-же памяти, то по факту система начинает просто сжимать информацию из оперативки и помещать ее в оперативку же. Скорость работы ОЗУ всегда существенно выше чем дисковой подсистемы, а алгоритмы сжатия **lzo** и **lz4** настолько быстры, что в итоге мы получаем существенное "увеличение" оперативной памяти за счет небольших процессорных издержек на архивацию. Таким образом, ZRAM позволяет разместить в оперативной памяти в несколько раз больше информации за счёт сжатия. Эта технология активно используется в **Android**, **ТВ-приставках**, **ChromeOS**, **SteamOS** и много где ещё. При использовании ZRAM, swap-раздел на диске необязателен. Это особенно полезно для SSD-накопителей, так как частые записи для них вредны.

Zram или Zswap?

Чтобы узнать что из этого лучше подойдёт вам, возьмём два примера: ноутбук с 4 гигами оперативки и медленным жёстким диском (представьте себе работу подкачки на таком) или с SSD-накопителем; компьютер с 8 гигами оперативки, на котором планируется запускать много виртуальных машин и других ресурсоёмких приложений. В первом случае, более подходящим будет ZRAM, так как ему не нужен swap-раздел на диске (это особенно вредно для SSD) и он позволит размещать все данные непосредственно в памяти. Однако **раздел swap нужно обязательно создать если вы будете использовать на ноутбуке режим сна.**

На компьютере предпочтительнее будет ZSWAP, так как оперативка не будет занята виртуальными swap-файлами (а виртуалкам нужно много оперативки), а в оперативке будет храниться только сжатый кэш, что при нехватке памяти предотвратит своппинг, а в критической ситуации - быстренько скинет его на swap-раздел на диске.

Установка

Примеры установки я покажу на дистрибутиве **Ubuntu** и **Debian**. В остальных дистрибутивах различия будут не существенными (обратитесь к документации вашего дистрибутива). Начнём с ZRAM. Для Ubuntu всё просто:

sudo apt install zram-config

Этот скрипт определит количество оперативной памяти в вашей системе, а также количество ядер процессора и создаст swar-файла в памяти (блочные устройства **/dev/zram**) по количеству ядер. Это нужно потому что сжатие данных однопоточное (один поток==одно ядро).

По умолчанию размер такого swar-файла равен 1/2 от общего количества оперативки. После перезагрузки, выполните в терминале команду swapon -s и вы увидите помимо реального swar-раздела, несколько разделов **/dev/zram**. Далее. Если вы изменяли значение **vm.swappiness** (для уменьшения порога включения подкачки), например выставили значение **vm.swappiness = 10**, то измените его на 40, дабы уже при исчерпании 60% оперативки, включался ZRAM. Если вы ничего не трогали, то можете либо оставить значение по умолчанию (60, то есть при исчерпании 40% оперативки), либо:

sudo nano /etc/sysctl.conf

Дописываем в конец строку:

vm.swappiness = 40

сохраняем и выполняем:

sudo sysctl -p

либо перезагружаемся. Вот и всё.

Для Debian всё немного сложнее. Но совсем чуть чуть. Открываем терминал, вводим:

sudo nano /etc/init.d/zram

Вставляем следующий скрипт:

```
#!/bin/sh #### BEGIN INIT INFO # Provides: zram # Required-Start: $local_fs # Required-Stop: $local_fs # Default-Start: S # Default-Stop: 0 1 6 # Short-Description: Use compressed
```

RAM as in-memory swap # Description: Use compressed RAM as in-memory swap ### END
INIT INFO

Author: Antonio Galea <antonio.galea@gmail.com> # Thanks to Przemysław Tomczyk for
suggesting swapoff parallelization

FRACTION=50

MEMORY=`perl -ne'/^MemTotal:\s+(\d+)/ && print \$1*1024;' < /proc/meminfo`
CPUS=`grep -c processor /proc/cpuinfo` SIZE=\$((MEMORY * FRACTION / 100 / CPUS))

```
case "$1" in "start") param=`modinfo zram|grep num_devices|cut -f2 -d:|tr -d ' '` modprobe  
zram $param=$CPUS for n in `seq $CPUS`; do i=$((n - 1)) echo $SIZE >  
/sys/block/zram$i/disksize mkswap /dev/zram$i swapon /dev/zram$i -p 10 done ;; "stop") for  
n in `seq $CPUS`; do i=$((n - 1)) swapoff /dev/zram$i && echo "disabled disk $n of $CPUS"  
& done wait sleep .5 modprobe -r zram ;; *) echo "Usage: `basename $0` (start | stop)" exit  
1 ;; esac
```

Сохраняем.

Далее даём права на исполнение:

`sudo chmod +x /etc/init.d/zram`

и активируем:

`sudo inserv zram`

Перезагружаемся и всё!



Комплексная оптимизация и ускорение Linux

Создайте в домашнем каталоге, пустую директорию **.compose-cache**. Это позволит задействовать кэширование информации, распарсенной Xorg и в результате немного увеличит скорость запуска графических приложений.

```
mkdir .compose-cache
```

Оптимизация интернет-соединения.

1) Можно увеличить скорость загрузки страниц, а также стабильность соединения, проведя кое-какие твики в сетевой подсистеме. Первым делом:

```
sudo nano /etc/nsswitch.conf
```

Находим строку hosts: files mdns4_minimal [NOTFOUND=return] dns

Снизу подписываем:

```
hosts: files dns
```

Должно получиться так:

```
hosts: files mdns4_minimal [NOTFOUND=return] dns hosts: files dns
```

Сохраняем и перезагружаемся.

Теперь страницы должны открываться быстрее.

Дополнительные твики для TCP/IP (*использовать с осторожностью!*)

```
sudo nano /etc/sysctl.conf
```

Добавляем в конец строки:

```
net.ipv4.tcp_fin_timeout = 30 net.ipv4.tcp_keepalive_time = 1800 net.ipv4.tcp_window_scaling = 0  
net.ipv4.tcp_sack = 0 net.ipv4.tcp_timestamps = 0
```

Сохраняем. Далее:

```
sudo sysctl -p
```

Preload и Prelink

1) Preload - это демон, который считывает статистику часто используемых приложений, и заранее готовится их подгрузить, поэтому запуск становится быстрее. Установка на примере Debian-based дистрибутивов:

```
sudo apt install preload
```

Настройка не требуется.

2) Prelink - утилита для связывания исполняемых файлов с разделяемыми библиотеками. Дело в том, что обычно программа не содержит в себе весь необходимый для работы код. Вместо этого, данный код помещается в так называемые разделяемые библиотеки (файлы с расширением .so, то есть shared object), которые могут использовать сразу несколько запущенных программ. Но при запуске программы, она начинает искать на диске и подгружать эти библиотеки, и в зависимости от их размера и количества, время запуска программы может оказаться весьма большим. Тут и поможет Prelink, который заранее "свяжет" все исполняемые файлы и необходимыми им библиотеками. Установка в Debian:

```
sudo apt install prelink sudo prelink -avfmR
```

Запускать prelink нужно после каждого обновления или установки программ. Можно включить периодическую перелинковку:

```
sudo nano /etc/default/prelink
```

Меняем параметр PRELINKING=unknown на PRELINKING=yes. Сохраняем и запускаем:

```
sudo /etc/cron.daily/prelink
```

В Ubuntu и её производных необходимости в Prelink нет

Увеличение отзывчивости системы в пиковые нагрузки.

1) Бывают ситуации, когда в момент очень высоких нагрузок, система ведёт себя не слишком отзывчиво, вплоть до подвисания курсора. Особенно это актуально если на компьютере весьма слабый процессор, а также мало оперативной памяти. Но можно улучшить ситуацию. Начиная с ядра **Linux 2.6.38**, в его состав включён знаменитый **200-строчный патч**, который серьёзно

повышает отзывчивость системы, путём динамической регулировки приоритетов процессорного времени для интерактивных и фоновых процессов. Что это значит? Допустим вы запустили компиляцию программы, и в этот момент включили плеер. Плеер получит больший приоритет процессорного времени, так как является интерактивным процессом, с которым работает пользователь, в то время как компиляция будет являться фоновым процессом, и если она начнёт "тянуть на себя одеяло", то ей понизится приоритет, в результате чего, плеер (либо другое интерактивное приложение) не будет тормозить от нехватки ресурсов процессора. Таким образом, даже при сильной загрузке системы, вы можете без проблем запускать любые ваши программы, такие как плееры, браузеры и даже игры. Однако несмотря на наличие этого патча в ядре, функция, которую он обеспечивает, не включена по умолчанию в некоторых дистрибутивах. Убедиться в её включении можно командой:

```
cat /proc/sys/kernel/sched_autogroup_enabled
```

Если вывод 1 - то всё включено, если 0 - то нет.

Включается она просто.

Открываем файл */etc/sysctl.conf*

```
sudo nano /etc/sysctl.conf
```

и в конец добавляем строку:

```
kernel.sched_autogroup_enabled = 1
```

Теперь выполните:

```
sudo sysctl -p
```

Оптимизация дисковых операций.

1) Это самое узкое место любой системы, не только Linux. Дело в том, что жёсткий диск является самым медленным устройством в компьютере. Если у вас установлен SSD, то это решает многие проблемы, но кое-что всё равно остаётся. Первой оптимизацией, является изменение размеров дискового кэша. При операциях копирования или перемещения, данные сначала записываются в оперативную память, а затем кучей скидываются на диск. В результате, особенно при нехватке памяти, это может привести к сильным задержкам. Исправить ситуацию поможет уменьшение дискового кэша до разумной величины. Также стоит уменьшить кэш файловых систем. Открываем уже знакомый нам `/etc/sysctl.conf` и в конец добавляем строки:

```
vm.dirty_bytes = 2097152 vm.dirty_background_bytes = 2097152 vm.vfs_cache_pressure = 1000
```

Прошу обратить внимание на последнюю строку: этот параметр отвечает за кэширование объектов файловой системы в оперативную память. При значении **0**, объекты не высвобождаются и так и остаются в оперативной памяти. Чем больше значение, тем чаще ядро будет проводить "зачистку" оперативной памяти. Поэтому если у вас оперативной памяти меньше 2 гигабайт, то поставьте значение **50 вместо 1000**, дабы сократить число дисковых операций в разделе подкачки. Это также полезно в случае если у вас SSD.

Но если у вас **больше 2 гигабайт оперативки**, и обычный жёсткий диск, то выставьте значение этого параметра на **1000**. Это позволит более агрессивно кэшировать дисковые операции, тем самым повысив быстродействие при достаточном количестве оперативной памяти.

Tmpfs

Tmpfs - это так называемый **RAM-диск**, то есть виртуальный диск, расположенный в оперативной памяти. В нём располагают какие-нибудь кэши или временные файлы, для мгновенного доступа к ним. В Линуксе (да и в других

Unix-подобных ОС), в нём чаще всего располагают каталог с временными файлами - **/tmp**. Это позволяет очень значительно повысить скорость работы, например, пакетного менеджера или любой другой программы, которая складывает туда временные файлы. Командой **df** можно посмотреть какие каталоги куда смонтированы:

```
vadim@RV-513: ~  
Файл  Правка  Вид  Поиск  Терминал  Справка  
vadim@RV-513:~$ df  
Файловая система 1K-блоков  Использовано  Доступно  Использовано%  Смонтировано в  
udev              10240          0          10240          0% /dev  
tmpfs             757820        9460        748360          2% /run  
/dev/sda1         48025944      5002388     40560940        11% /  
tmpfs             1894544        47368      1847176          3% /dev/shm  
tmpfs              5120           4          5116           1% /run/lock  
tmpfs             1894544          0      1894544          0% /sys/fs/cgroup  
tmpfs             2097152          4      2097148          1% /tmp  
/dev/sda6         255530516     23063864    219463388        10% /home  
tmpfs              378912         32       378880           1% /run/user/1000  
vadim@RV-513:~$
```

Как видим, в tmpfs уже по умолчанию смонтированы некоторые каталоги, к которым часто обращается система или программы. К слову, Tmpfs - не единственная реализация рамдиска в Linux. Популярной альтернативой ему является **Ramfs**. Чем они отличаются:

- Ramfs никогда не использует подкачку, а заполняет оперативку до предела;
- Ramfs, в отличие от Tmpfs, не поддерживает никаких параметров монтирования и всегда монтируется с правами root;
- Tmpfs может использовать подкачку, поддерживает параметры монтирования и позволяет жёстко указать размер файловой системы.

Для того чтобы смонтировать /tmp в Tmpfs, делаем так:

```
sudo nano /etc/fstab
```

В конец добавляем строку (обратите внимание, что вместо пробелов используется табуляция. То есть нужно нажимать клавишу Tab):

```
tmpfs /tmp tmpfs defaults,size=2G,mode=1777 0 0
```

Таким образом мы смонтировали /tmp в tmpfs, указали ему размер в 2 гига, после чего он начнёт использовать подкачку, а также выставили права доступа 1777, чтобы не было проблем в работе некоторых приложений. Сохраните и перезагрузитесь.

Вот в принципе и всё.



Безопасность Linux-Шифруемся

Сегодня шифрование везде и нигде одновременно. Нас заставляют пользоваться HTTPS-версиями сайтов, а нам все равно.

Шифрование — это единственное абсолютное средство защиты на сегодняшний день. А еще оно очень доступно и сглаживает морщины.

В Linux можно найти тонны средств шифрования всего и вся, от разделов на жестком диске до одиночных файлов. Три наиболее известных и проверенных временем инструмента — это dm-crypt/LUKS, ecryptfs и encfs. Первый шифрует целые диски и разделы, второй и третий — каталоги с важной информацией, каждый файл в отдельности, что очень удобно, если потребуется делать инкрементальные бэкапы или использовать в связке с Dropbox. Также есть несколько менее известных инструментов, включая TrueCrypt например.

Шифровать весь диск целиком — задача сложная и, что самое важное, бесполезная. Ничего особо конфиденциального в корневом каталоге нет и быть не может, а вот домашний каталог и своп просто кладезь инфы. Причем второй даже больше, чем первый, так как туда могут попасть данные и пароли уже в расшифрованном виде (нормальные программеры запрещают системе скидывать такие данные в своп, но таких меньшинство). Настроить шифрование и того и другого очень просто, достаточно установить инструменты ecryptfs:

```
$ sudo apt-get install ecryptfs-utils
```

И включить шифрование:

```
$ sudo ecryptfs-setup-swap $ ecryptfs-setup-private
```

Далее достаточно ввести свой пароль, используемый для логина, и перезайти в систему. Да, все действительно так просто. Первая команда зашифрует и перемонтирует своп, изменив нужные строки в /etc/fstab. Вторая — создаст каталоги ~/.Private и ~/Private, в которых будут храниться зашифрованные и расшифрованные файлы соответственно. При входе в систему будет срабатывать РАМ-модуль ram_ecryptfs.so, который смонтирует первый каталог на второй с прозрачным шифрованием данных. После размонтирования ~/.Private окажется пуст, а ~/.Private будет содержать все файлы в зашифрованном виде.

Не возбраняется шифровать и весь домашний каталог целиком. Производительность при этом упадет не сильно, зато под защитой окажутся вообще все файлы.

Делается это так:

```
# ecryptfs-migrate-home -u имяпользователя
```

После завершения операции следует сразу войти под юзером и проверить работоспособность:

```
$ mount | grep Private
```

Если все ок, незашифрованную копию данных можно затереть:

```
$ sudo rm -r /home/имяпользователя.*
```

```
INFO: Checking disk space, this may take a few moments. Please be patient.
INFO: Checking for open files in /home/vasya
```

```
*****
YOU SHOULD RECORD YOUR MOUNT PASSPHRASE AND STORE IT IN A SAFE LOCATION.
  ecryptfs-unwrap-passphrase ~/.ecryptfs/wrapped-passphrase
THIS WILL BE REQUIRED IF YOU NEED TO RECOVER YOUR DATA AT A LATER TIME.
*****
```

```
Done configuring.
```

```
INFO: Encrypted home has been set up, encrypting files now...this may take a while.
```

```
=====
Some Important Notes!
```

1. The file encryption appears to have completed successfully, however, ashep MUST LOGIN IMMEDIATELY, `_BEFORE_THE_NEXT_REBOOT_`, TO COMPLETE THE MIGRATION!!!
2. If ashep can log in and read and write their files, then the migration is complete, and you should remove `/home/vasya.81yFQbNJ`. Otherwise, restore `/home/vasya.81yFQbNJ` back to `/home/vasya`.

```
Chain INPUT (policy ACCEPT)
target     prot opt source                destination            udp dpt:bootps
ACCEPT     udp  -- anywhere             anywhere              tcp dpt:bootps
ACCEPT     tcp  -- anywhere             anywhere              udp dpt:domain
ACCEPT     tcp  -- anywhere             anywhere              tcp dpt:domain

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination            state RELATED,ESTABLISHED
ACCEPT     all  -- 10.42.0.0/24          anywhere
ACCEPT     all  -- anywhere             anywhere
REJECT     all  -- anywhere             anywhere              reject-with icmp-port-unreachable
REJECT     all  -- anywhere             anywhere              reject-with icmp-port-unreachable

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
vadim@RV-513:~$
```

Debian-Настройка iptables: NAT, фаервола

Фаервол нам нужен для ограничения подключения из вне, NATа, а также проброса портов. Для удобства написал скрипт на bash.

Вставляем в файл /etc/init.d/rc.firewall

```
#!/bin/bash

IPTABLES="/sbin/iptables"

##### Config #####
LNETS="eth1 eth2 wlan0"

DESKTOP="192.168.1.1"
DESKTOP_OPEN_PORT="8010 9000"

DESKTOP2="192.168.2.1"
DESKTOP2_OPEN_PORT="9003"

HOME_MASKS="192.168.1.0/24 192.168.2.0/24 192.168.3.0/24"

PROVIDER="eth0"
```

```

PROVIDER_IP="10.0.4.59"
PROVIDER_MASK="10.0.0.0/8"
INET="ppp+"
WHITE_IP="77.77.77.77"

OPEN_PORTS="22,80"
#####
echo 1 > /proc/sys/net/ipv4/conf/default/rp_filter
echo 1 > /proc/sys/net/ipv4/ip_forward
modprobe ip_conntrack
modprobe ip_conntrack_ftp
modprobe nf_nat_pptp
modprobe nf_conntrack_pptp
modprobe nf_conntrack_proto_gre
modprobe nf_nat_proto_gre
modprobe iptable_nat
modprobe ip_nat_ftp
modprobe ipt_LOG

$IPTABLES -P INPUT ACCEPT
$IPTABLES -P OUTPUT ACCEPT
$IPTABLES -P FORWARD DROP

$IPTABLES -F
$IPTABLES -X
$IPTABLES -t nat -F PREROUTING
$IPTABLES -t nat -F POSTROUTING

##### DELETE IF ALL WORKING FINE #####
#$IPTABLES -A INPUT -j ACCEPT
#####

#mtu for vpn magick command, mega debian epic fail
$IPTABLES -o $INET -A FORWARD -p tcp --tcp-flags SYN,RST SYN -m tcpmss --mss 800:1536
-j TCPMSS --clamp-mss-to-pmtu

# DENY SECTIONS
$IPTABLES -A INPUT -p tcp ! --syn -m state --state NEW -j DROP
$IPTABLES -A INPUT -p tcp --tcp-flags SYN,ACK SYN,ACK -m state --state NEW -j DROP

# local interface, allow all
$IPTABLES -A INPUT -i lo -j ACCEPT

# ALLOW PACKETS IF CONNECTION ESTABLISHED
$IPTABLES -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

```



```

# access from white ip
$IPTABLES -A INPUT -s $WHITE_IP -j ACCEPT

# access from home net
for i in $HOME_MASKS; do
$IPTABLES -A INPUT -s $i -j ACCEPT
done

# defence for ssh for server
$IPTABLES -A INPUT -p tcp -m state --state NEW --dport 22 -m recent --update --seconds 20 -j
DROP
$IPTABLES -A INPUT -p tcp -m state --state NEW --dport 22 -m recent --set -j ACCEPT

# open ports for server
$IPTABLES -A INPUT -p tcp --syn -m multiport --destination-ports $OPEN_PORTS -j ACCEPT

##### FORWARD #####
$IPTABLES -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
#forward each home eth to provider eth and ppp+
for i in $LNETS; do
    $IPTABLES -A FORWARD -i $INET -o $i -j ACCEPT
    $IPTABLES -A FORWARD -i $i -o $INET -j ACCEPT
    $IPTABLES -A FORWARD -i $i -o $PROVIDER -j ACCEPT
#forward each home eth to other home eth
for j in $LNETS; do
    if [ "$i" != "$j" ] ; then
        $IPTABLES -A FORWARD -i $i -o $j -j ACCEPT
        $IPTABLES -A FORWARD -i $j -o $i -j ACCEPT
    fi
done
done

#####

##### LOCAL NAT #####
for i in $HOME_MASKS; do
    $IPTABLES -t nat -A POSTROUTING -s $i -d $PROVIDER_MASK -j SNAT --to-source
$PROVIDER_IP
done

#####

##### INET NAT #####
for i in $HOME_MASKS; do
    $IPTABLES -t nat -A POSTROUTING -s $i -j SNAT --to-source $WHITE_IP
done

#####

```

```
##### FORWARD PORTS #####
for i in $DESKTOP_OPEN_PORT; do
    $IPTABLES -t nat -A PREROUTING -p tcp --dport $i -j DNAT --to $DESKTOP:$i
    $IPTABLES -A FORWARD -p tcp -d $DESKTOP --dport $i -j ACCEPT
done

for i in $DESKTOP2_OPEN_PORT; do
    $IPTABLES -t nat -A PREROUTING -p tcp --dport $i -j DNAT --to $DESKTOP2:$i
    $IPTABLES -A FORWARD -p tcp -d $DESKTOP2 --dport $i -j ACCEPT
done

# ssh for destop with defence
$IPTABLES -t nat -A PREROUTING -p tcp --dport 12345 -j DNAT --to $DESKTOP:22
$IPTABLES -A FORWARD -p tcp -m state -d $DESKTOP --state NEW --dport 12345 -m recent --
update --seconds 20 -j DROP
$IPTABLES -A FORWARD -p tcp -m state -d $DESKTOP --state NEW --dport 12345 -m recent --
set -j ACCEPT

# ping
$IPTABLES -A INPUT -p ICMP --icmp-type 8 -j ACCEPT
# deny other ICMP packets
$IPTABLES -A INPUT -p icmp -j DROP

# other reject
$IPTABLES -A INPUT -j DROP $IPTABLES -A FORWARD -j DROP
```

Делаем скрипт исполняемым:

```
chmod +x /etc/init.d/rc.firewall
```

И прописываем в автозагрузку

```
update-rc.d rc.firewall defaults
```

По сути в нём всё лаконично закоментированно, но iptables – та ещё магия. В скрипте почти все настройки вынесены. Скрипт разрешает форвардинг между всеми интерфейсами, включает NAT, разрешает пинги до сервера. Всё что явно не разрешено, то запрещено. Бонусом есть защита ssh от брута, форвардинг ssh с порта 12345 на 22 порт домашнего компьютера, форвардинг портов на

внутренние компьютеры (разделяются пробелом) Там найдёте магическую строку с моим негативом. Дебиан почему-то не может поставить MTU для ppp. Всё бы ничего, но некоторые сайты не открываются (nix.ru, mail.ru). Как говорят знакомые админы баг тянется ещё с etch.



Автоматизация рутинных задач

Ни для кого не секрет что система в процессе своей работы засоряется. Установка и удаление программ, выполнение скриптов, обновления, неверная установка программ из исходников, ошибки в программах, все это оставляет в системе лишние, ненужные пакеты. Со временем этих файлов может накапливаться достаточно большое количество.

Linux засоряется намного меньше Windows, здесь нет реестра, и архитектура системы рассчитана так, чтобы оставлять мусор по минимуму. Но кроме системных недочетов, в файловой системе могут накапливаться ненужные файлы пользователей и программ. Поэтому очистка от мусора необходима.

Выполнять ежедневно кучу команд в терминале попросту надоедает, так что напишем скрипт который автоматизирует работу по очистке и обновлению системы за нас.

Открываем терминал и создаём файл скрипта:

```
touch update.sh
```

Открываем файл в любом удобном редакторе и заполняем таким содержимым.

Я использую nano

```
nano update.sh
```

Заполняем:

```
#!/bin/bash
```

```
# Проверка на наличие прав root
```

```
ROOT_UID=0 #Только пользователь с $UID 0 имеет привилегии root.
```

```
E_NOTROOT=67 #Признак отсутствия root привелегий.
```

```
if [ "$UID" -ne "$ROOT_UID" ]
```

```
then
```

```
echo -e "$Для работы скрипта требуются права root. Перезапустите скрипт используя sudo"
```

```
tput sgr0
```

```
exit $E_NOTROOT
```

```
fi
```

```
echo "Обновление списка пакетов"
```

```
apt-get update -y
```

```
sleep 2
```

```
echo "Установка обновлений"
```

```
apt-get upgrade -y
```

```
sleep 2
```

```
echo "Очистка"
```

```
apt-get clean -y
```

```
apt-get autoremove -y
```

```
clear
```

Сохраняем

```
Ctrl+O
```

Теперь дадим файлу права на исполнение:

```
sudo chmod +x update.sh
```

Вот собственно и всё!

Теперь очистка и обновление системы можно выполнять одной командой:

```
sudo ./update.sh
```

Кому нужно, может пойти немного дальше, и добавить выполнение в **cron**



10 расширений для «Браузера», которые пригодятся каждому

Прокачайте браузер с помощью этих бесплатных программ.

1) *LastPass*

LastPass ●●●|

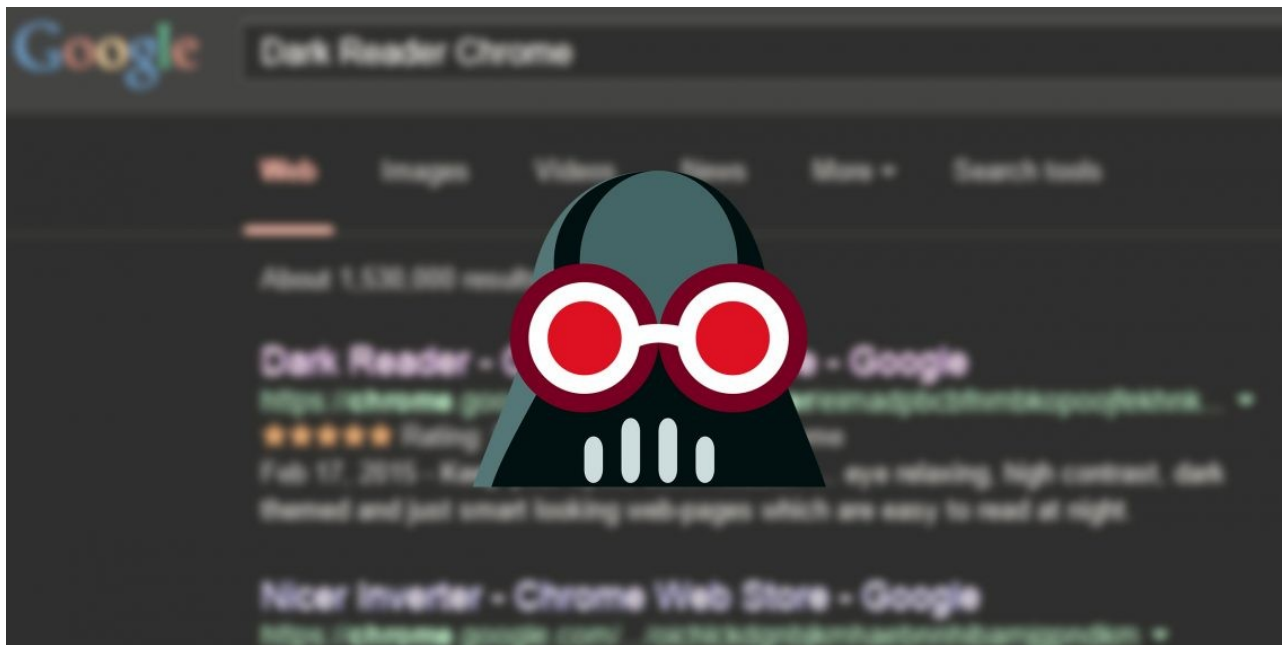
LastPass избавляет от необходимости запоминать данные для входа в аккаунты. Сервис помнит логины и пароли вместо вас и хранит их в защищённом виде на сервере. Более того, LastPass автоматически вводит пароли, когда это нужно.

Можно менять браузеры и устройства — необходимые данные будут всегда оставаться под рукой благодаря синхронизации. Вам нужно помнить только пароль от аккаунта LastPass.

Mozilla Firefox: <https://addons.mozilla.org/ru/firefox/addon/lastpass-password-manager/>

Google Chrome: <https://chrome.google.com/webstore/detail/lastpass-free-password-manager/ldapiejdjndhahjhdhceplioahd?hl=ru>

2) Dark Reader



Dark Reader помогает снизить нагрузку на глаза в ночное время суток. Работает расширение просто: заменяет светлые элементы веб-страниц тёмными. Таким образом браузер переходит в ночной режим работы.

Вы можете использовать стандартную схему замены цветов или настроить собственную. Кроме того, Dark Reader умеет включать ночной режим только на выбранных сайтах.

Mozilla Firefox: <https://addons.mozilla.org/ru/firefox/addon/darkreader/?src=search>

Google Chrome:

<https://chrome.google.com/webstore/detail/dark-reader/eimadpbcbfnmbkopoojfekhnkhdhiech?hl=ru>

3)Checker Plus for Gmail

CHECKER PLUS

for Gmail™



Если вы активно пользуетесь популярнейшим почтовым сервисом Gmail, обязательно попробуйте это расширение. Checker Plus уведомляет о получении новых писем и позволяет открывать их или отмечать прочитанными, не переходя на новую вкладку.

Mozilla Firefox: <https://addons.mozilla.org/ru/firefox/addon/checker-plus-gmail/?src=search>

Google Chrome: <https://chrome.google.com/webstore/detail/checker-plus-for-gmail/oeopbcgkkoapgobdbedcemjljbihmemj?hl=ru>

4) HTTPS Everywhere



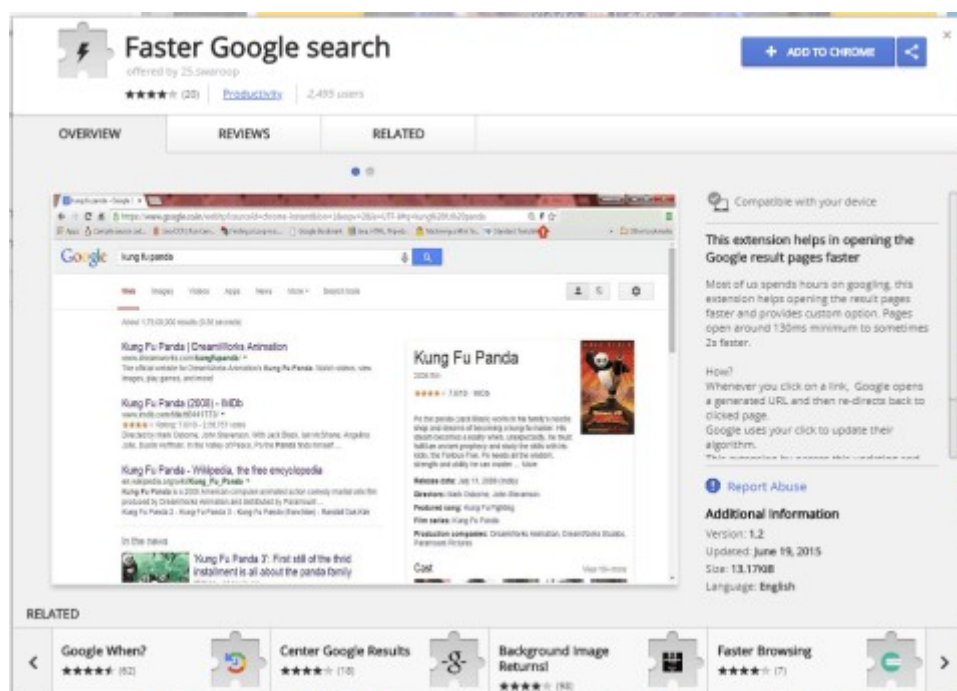
Расширение, повышающее безопасность интернет-сёрфинга. Оно принудительно переключает соединение с небезопасного протокола HTTP на шифрованный HTTPS везде, где это возможно.

Mozilla Firefox: <https://addons.mozilla.org/ru/firefox/addon/https-everywhere/>

Google Chrome:

<https://chrome.google.com/webstore/detail/https-everywhere/gcbommkclmclpchllfjekcdonpmejbdp?hl=ru>

5) Faster Google Search



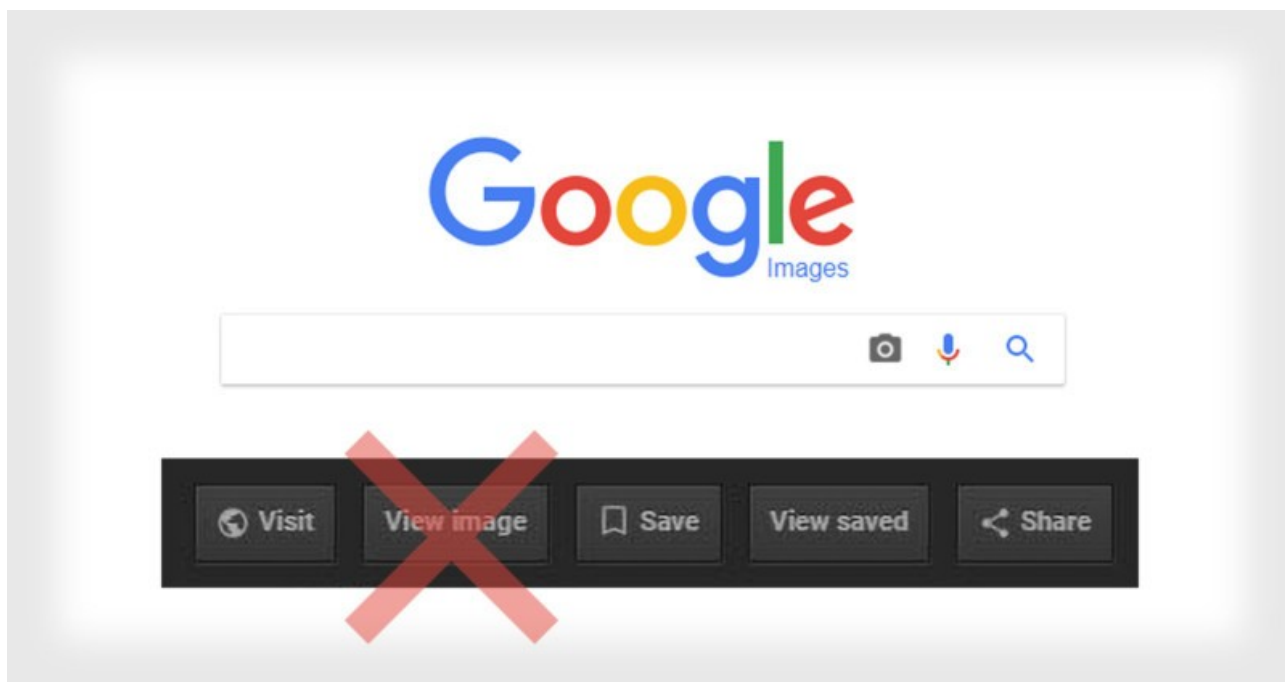
Когда вы нажимаете на ссылку на странице поиска Google, вы не сразу открываете нужный сайт. Вместо этого Google сначала перенаправляет вас на специальную страницу для отслеживания посещаемости сайтов.

Faster Google Search отключает эту функцию, так что вы сможете мгновенно открывать сайты без перенаправления на страницы Google. Кроме того, с этим расширением намного удобнее копировать адреса найденных сайтов прямо на странице с результатами.

Google Chrome:

<https://chrome.google.com/webstore/detail/faster-google-search/fmhiafgdfaglfbaaefbbolaholfhkmjc?hl=ru>

6) View Image

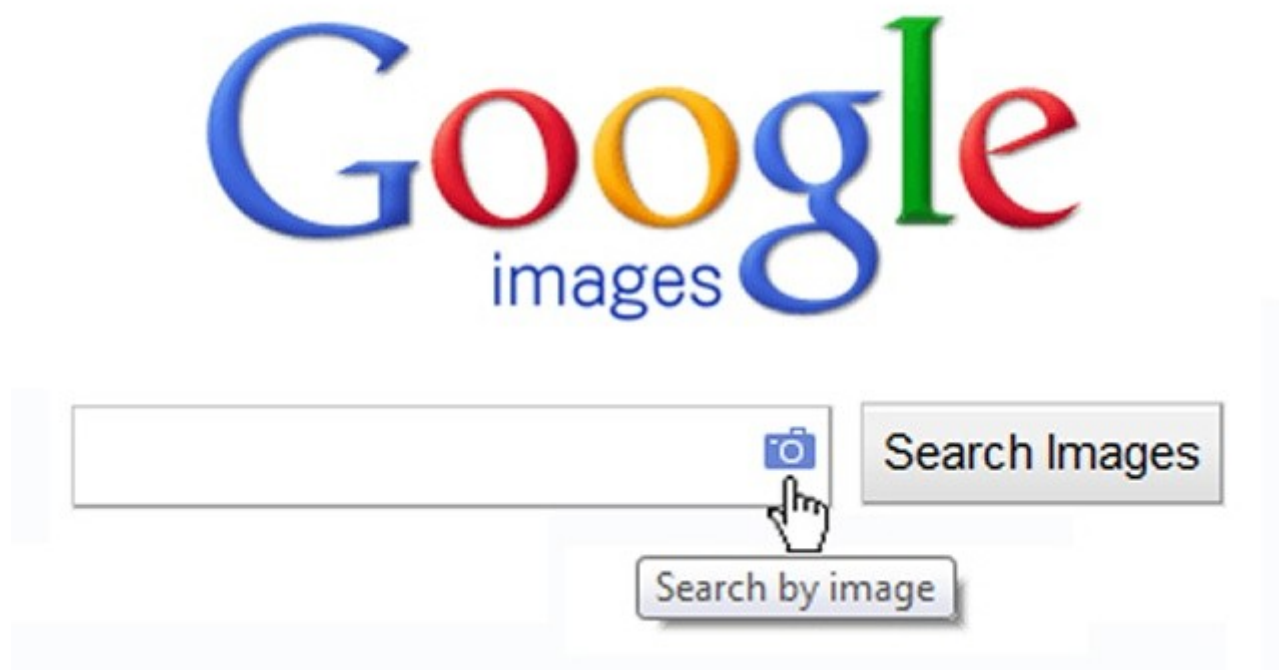


Google недавно убрала кнопки «Открыть в полном размере» и «Поиск по картинке» со своей поисковой страницы. Если раньше вы могли сразу открыть найденную картинку прямо в браузере, то теперь вам придётся каждый раз переходить на сайт, содержащий изображение, и уже оттуда открывать картинку. Эту проблему легко обойти, установив View Image. Это расширение вернёт удалённые кнопки на место.

Mozilla Firefox: <https://addons.mozilla.org/ru/firefox/addon/view-image/>

Google Chrome:

<https://chrome.google.com/webstore/detail/view-image/jpcmhcelnjdmblfmjabdeclccemkghjk?hl=ru>

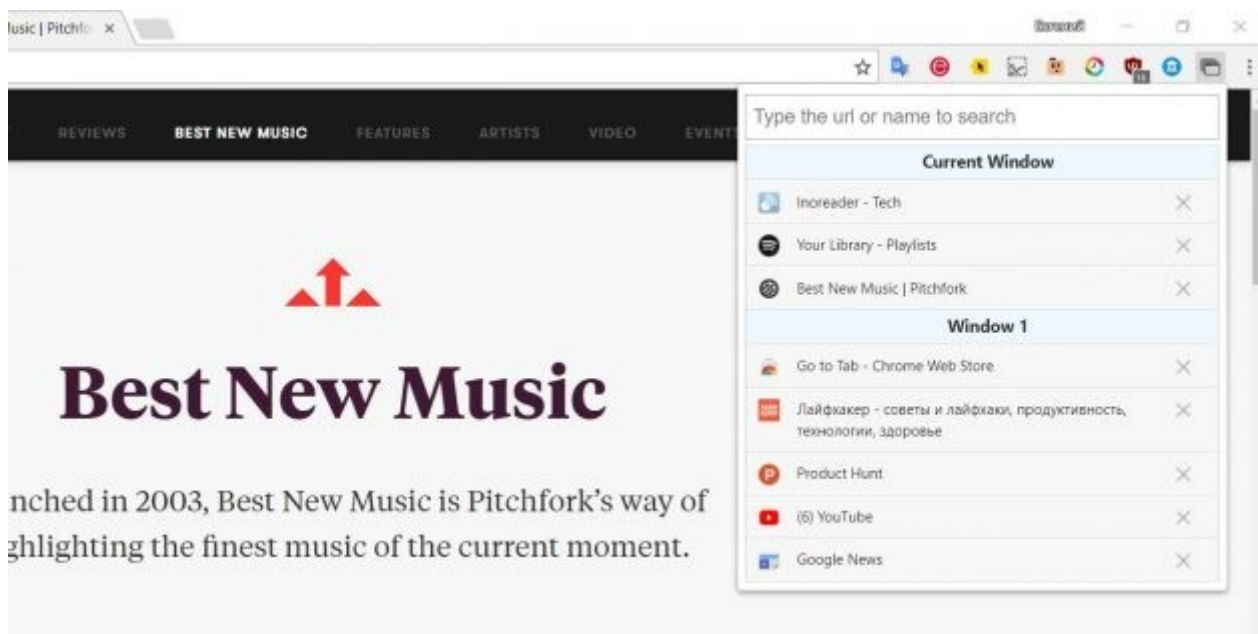
7) Search by Image (by Google)

Функция поиска изображений Google замечательна, с ней не нужно никакого TinEye. Просто скормите поисковику нужную картинку, и он отыщет все изображения, похожие на неё.

Расширение Search by Image делает поиск картинок ещё проще. Щёлкните на изображении правой кнопкой мыши и выберите «Поиск в Google».

Google Chrome: <https://chrome.google.com/webstore/detail/search-by-image-by-google/dajedkncpodkkgklbegccjpmnglmnflm?hl=ru>

8) Go to Tab



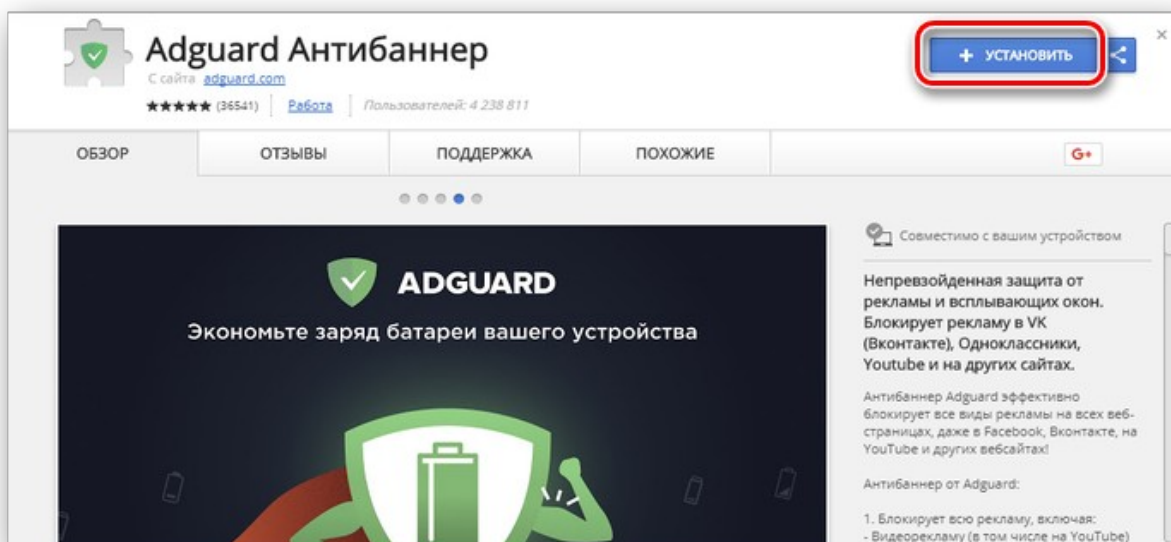
Если вы привыкли работать с огромным количеством вкладок, то это расширение поможет вам ими управлять. Оно показывает горизонтальный список открытых в браузере веб-страниц с их полными названиями.

Вы можете быстро переключаться между вкладками, закрывать выбранные или искать нужные страницы в списке по заголовку или URL. На фоне конкурентов Go to Tab выделяется простотой — в расширении нет ничего лишнего.

Google Chrome:

<https://chrome.google.com/webstore/detail/go-to-tab/hjfkaoibgkmaeomgdhmmhipdbjdhhjkoi?hl=ru>

9) Adguard Антибаннер от Adguard Software Ltd



Непревзойденная защита от рекламы и всплывающих окон. Блокирует рекламу в VK (Вконтакте), Одноклассники, Youtube и на других сайтах.

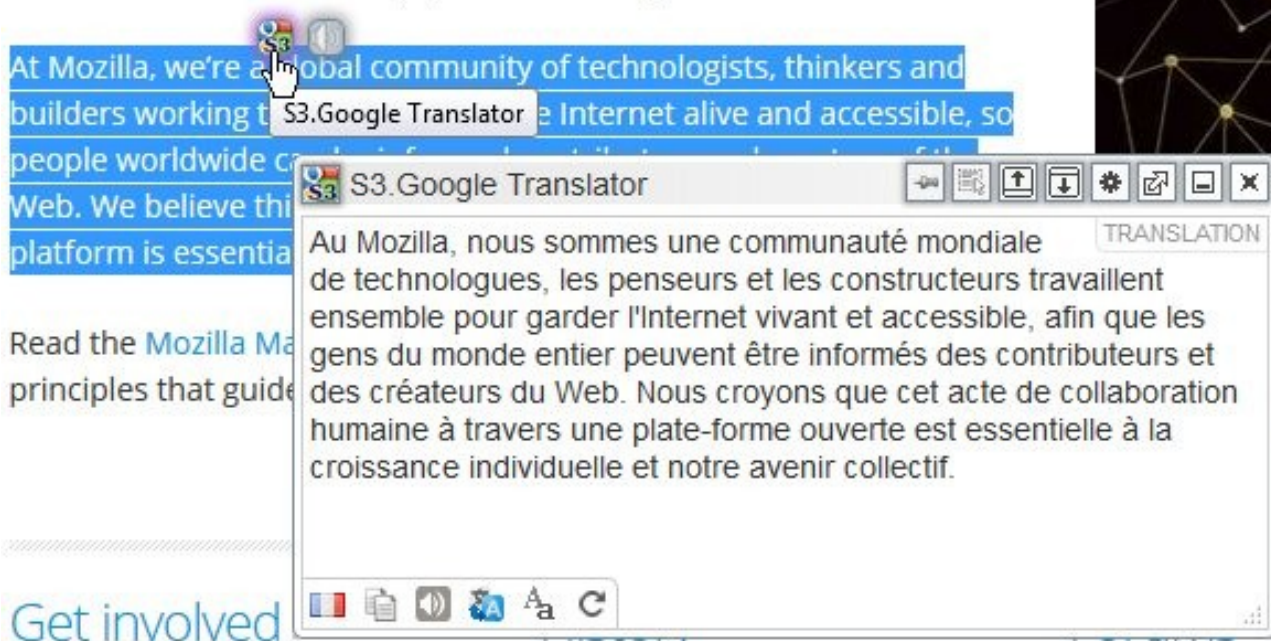
Mozilla Firefox: <https://addons.mozilla.org/ru/firefox/addon/adguard-adblocker/?src=search>

Google Chrome:

<https://chrome.google.com/webstore/detail/adguard-adblocker/bgnkhnnamicmpeenaelnjfhikgbklg?hl=ru>

10) S3.Translator

Our mission is to promote openness,
innovation & opportunity on the Web.



S3.Переводчик - перевод выделенного текста, любой введенной фразы или полностью всего web-сайта с любого языка на любой язык

Новая функция: Изучение языка!

Mozilla Firefox: <https://addons.mozilla.org/addon/s3google-translator/>

Google Chrome:

https://chrome.google.com/webstore/detail/s3translator/debnnjfbneojbmioajinefnflop_dohjk



Учимся искать в интернете

Google-Функциональный поиск

Базовый примеры:

бикини Италия — будет произведён поиск по словам "бикини" и "Италия"

переработка металла OR пластика — будет искать "переработка металла" или "переработка пластика"

"у меня есть мечта" — поиск именно "этой" фразы

сальса -танец — искать слово "сальса", но без слова "танец"

Людовик "I" Франция — искать Людовика Первого (I), короля именно Франции

замок ~термин — искать слово "замок", а также все синонимы слова "термин"

Калькулятор:

+ - * / — арифметика(обычная)

% of — выведет количество процентов от указанного числа(прим. *10% of 50*)

[^] или ** — возведение в степень

из <значение> in в <значение> — конвертер (прим. валют - 300 евро in рубли)

Поиск узкой направленности:

город1 город2 — расписание, билеты из город1 в город2

site:<сайт> — поиск на "сайте"

<число>..<число> — поиск в указанном диапазоне чисел

filetype:(или ext:<тип_файла> — искать по "типу файла"

link:<сайт> — искать ссылку на сайтах на "сайт"

Специальный информативный поиск:

book <книга> — искать "книгу"

phonebook:<тел.номер> — поиск "телефонного номера"

bphonebook:<тел.номер> — поиск "телефонного номера" компании

define, what is, what are — искать определение слова или фразы

define:<слово> — вывести определение "слова"

movie:<слово> — искать "слово" в описании к фильму, включая обзоры и прочее

stocks:<товар> — искать "товар"

weather <локация> — выводит погоду в "локации" (город или zip код)

Альтернативные типы поиска:

cache:<ссылка> — вывести кэшированную версию для "ссылки"

info:(или *id*:)<сайт> — искать информацию о "сайте"

related:<сайт> — вывести список похожих(аналогичных) "сайтов"

Узкий поиск по сайтам:

allinanchor:<слова> — поиск всех указанных "слов" в ссылках сайта

inanchor:<слово> — поиск "слова" в ссылках сайта

allintext:<слова> — искать все указанные "слова" в тексте сайта

intext:<слово> — искать "слово" в тексте сайта

allintitle:<слова> — поиск всех "слов" в заголовках сайтов

intitle:<слово> — поиск "слова" в заголовках сайтов

allinurl:<слова> — искать все указанные "слова" в ссылке сайта

inurl:<слово> — искать "слово" в ссылке сайта

Узкий поиск по Google Groups:

author:<автор> — поиск сообщения по группам Гугл от "автора"

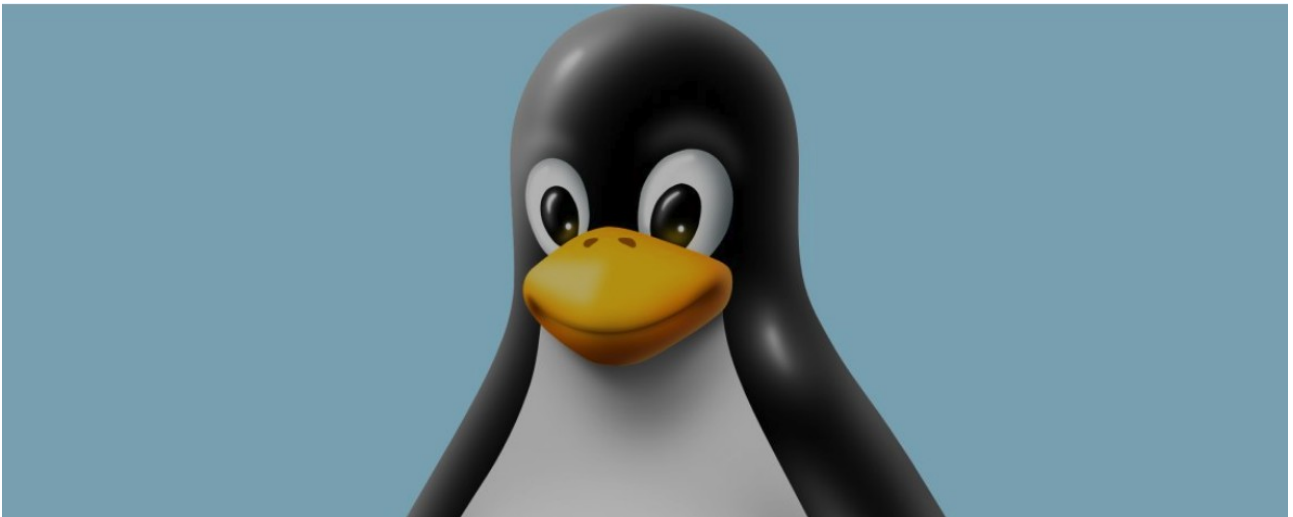
group:<группа> — поиск сообщения по группам Гугл в "группе"

insubject:<тема> — поиск сообщения по группам Гугл в "теме"

Узкий поиск по Google News:

location:<локация> — поиск новости от Гугл по данной "локации"

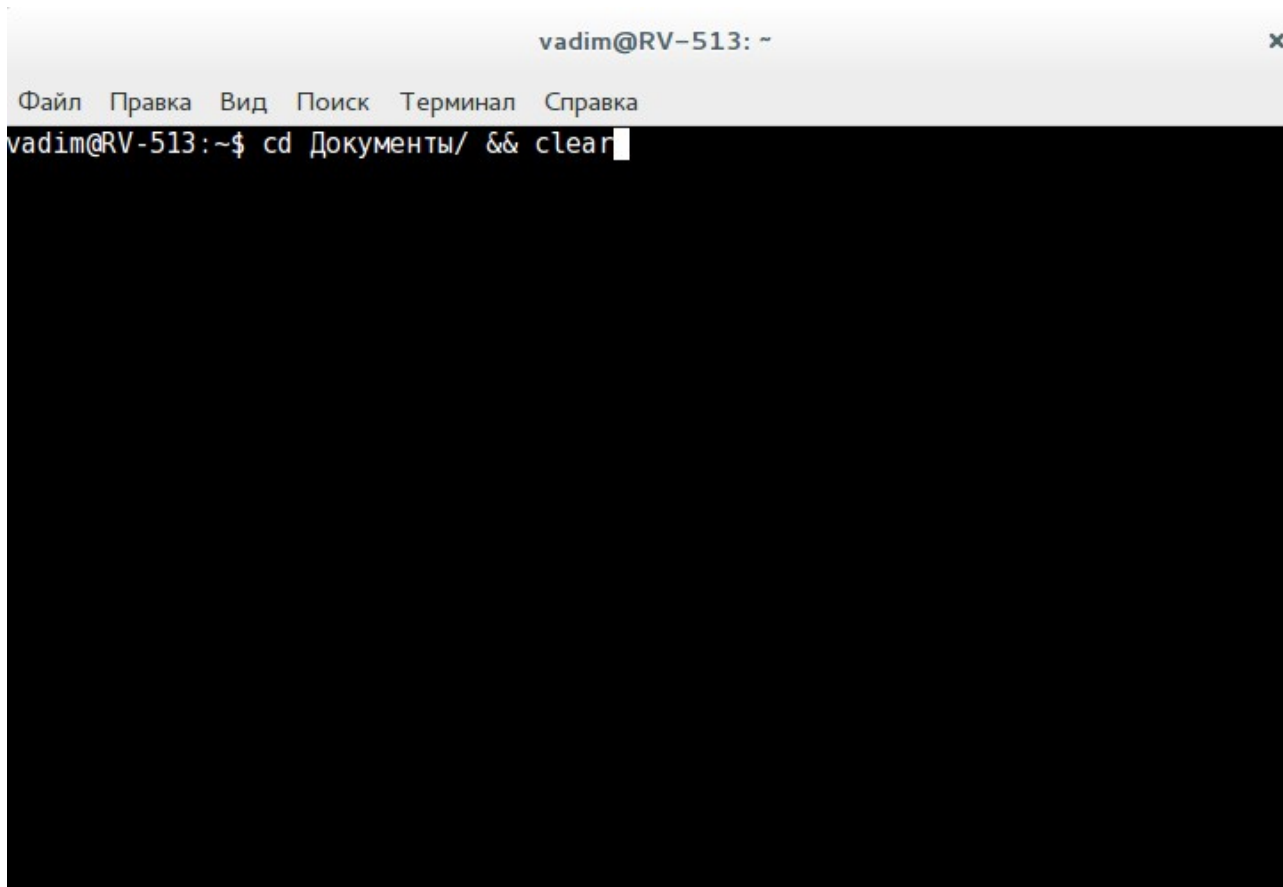
source:<источник> — поиск новости от Гугл из данных "источников"



Команды Linux на все случаи жизни!

В терминале Linux можно делать практически что угодно: настраивать систему, устанавливать и удалять приложения, управлять дисками и файлами.

Команды Linux для навигации в терминале



1. `&&`. Строго говоря, это не команда. Если вы хотите выполнить сразу несколько команд, поставьте между ними двойной амперсанд вот так: `первая_команда && вторая_команда`. Терминал выполнит команды по порядку. Вы можете ввести столько команд, сколько захотите.
2. `alias`. Присваивает созданные вами названия длинным командам, которые вы не можете запомнить. Введите `alias длинная_команда короткая_команда`.
3. `cd`. Изменяет текущую терминальную папку. Когда вы запускаете терминал, он использует вашу домашнюю папку. Введите `cd адрес_папки`, и терминал будет работать с файлами, которые там находятся.
4. `clear`. Очищает окно терминала от всех сообщений.

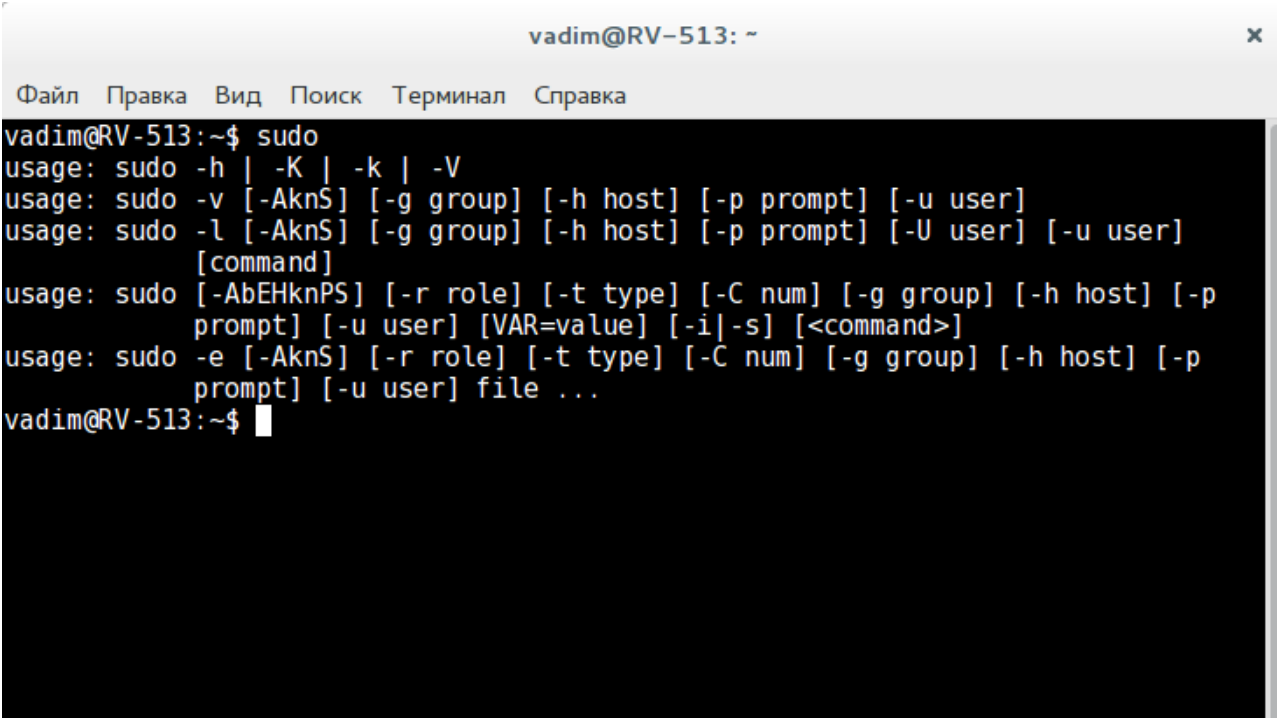
5. `history`. Отображает все недавно введенные вами команды. Кроме того, вы можете переключаться между недавними командами с помощью клавиш «Вверх» и «Вниз». Если вы не хотите, чтобы введенная вами команда была записана, поставьте перед ней пробел так: `ваша_команда`.

6. `man`. Отображает руководство по программам и командам `Linux`. Введите `man имя_пакета` или `man ваша_команда`.

7. `whatis`. Отображает краткое описание какой-либо программы. Введите команду и название программы `whatis имя_пакета`.

Команды Linux для получения прав суперпользователя

Для выполнения множества действий в системе, например для установки и удаления программ, вам понадобятся права администратора, или суперпользователя `root`, как его называют в `Linux`.



```
vadim@RV-513: ~  
Файл  Правка  Вид  Поиск  Терминал  Справка  
vadim@RV-513:~$ sudo  
usage: sudo -h | -K | -k | -V  
usage: sudo -v [-AknS] [-g group] [-h host] [-p prompt] [-u user]  
usage: sudo -l [-AknS] [-g group] [-h host] [-p prompt] [-U user] [-u user]  
[command]  
usage: sudo [-AbEHknPS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p  
prompt] [-u user] [VAR=value] [-i|-s] [<command>]  
usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p  
prompt] [-u user] file ...  
vadim@RV-513:~$
```

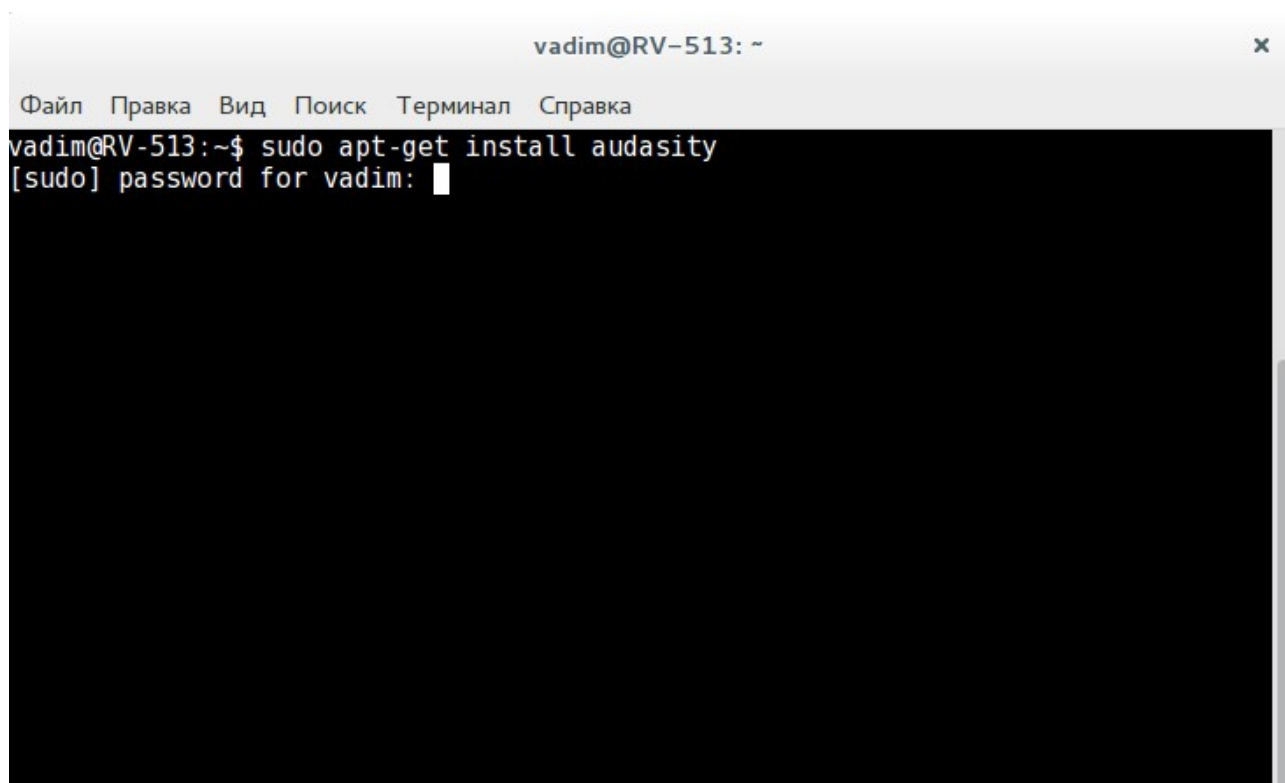
sudo. Эта команда даст вам права суперпользователя. Введите `sudo` перед нужной командой (например, `sudo apt upgrade`), чтобы выполнить её от имени администратора. Система спросит у вас пароль.

`sudo su.` После этой команды все введённые вами команды будут исполняться от имени суперпользователя, пока вы не закроете терминал. Используйте её, если вам нужно выполнить много команд с правами администратора.

`sudo gksudo.` Команда для запуска с правами администратора приложения с графическим интерфейсом. Например, если вы хотите переместить или изменить системные файлы, введите `sudo gksudo nautilus` (укажите тот файловый менеджер, которым пользуетесь).

`sudo !!.` Эта команда запустит ранее введённую команду с правами администратора. Полезно, если вы набрали команду без `sudo`.

Команды Linux для управления пакетным менеджером



```
vadim@RV-513: ~  
Файл Правка Вид Поиск Терминал Справка  
vadim@RV-513:~$ sudo apt-get install audacity  
[sudo] password for vadim: 
```

Установку и удаление приложений в Linux выполняют пакетные менеджеры. В Ubuntu и Debian пакетный менеджер называется apt. Он загружает приложения из сетевых репозитариев, источников пакетов. Давать им команды следует с правами суперпользователя.

apt (Debian/Ubuntu/Mint)

`sudo apt install имя_пакета`. Установить нужный пакет.

`sudo apt-add-repository адрес_репозитария`. Добавить сторонний репозитарий.

`sudo apt update`. Обновить сведения о пакетах.

`sudo apt upgrade` . Обновить все пакеты до самых свежих (выполнять после `apt update`).

`sudo apt remove имя_пакета` . Удалить ненужный пакет.

`sudo apt purge имя_пакета` . Удалить ненужный пакет со всеми зависимостями, если хотите освободить больше места.

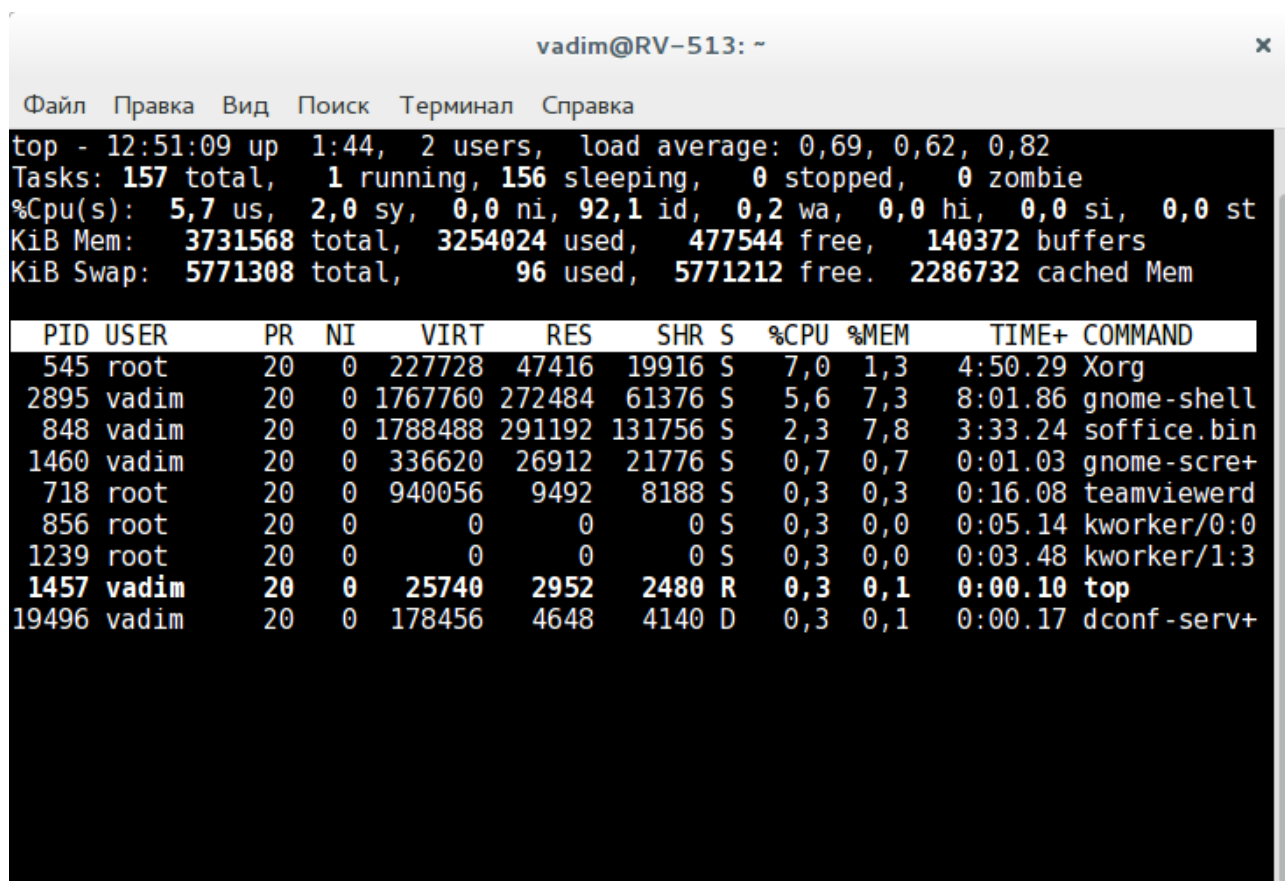
`sudo apt autoremove` . Удалить все ненужные зависимости, бесхозные пакеты и прочий мусор.

Вы можете устанавливать и удалять сразу несколько пакетов, просто перечисляя их через пробел.

```
sudo apt install firefox clementine vlc
```

Если вы хотите установить какой-то пакет, но не знаете его точного названия, введите несколько первых букв имени пакета и дважды нажмите Tab. Пакетный менеджер покажет все пакеты, названия которых начинаются одинаково.

Команды Linux для управления процессами



```
vadim@RV-513: ~  
Файл Правка Вид Поиск Терминал Справка  
top - 12:51:09 up 1:44, 2 users, load average: 0,69, 0,62, 0,82  
Tasks: 157 total, 1 running, 156 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 5,7 us, 2,0 sy, 0,0 ni, 92,1 id, 0,2 wa, 0,0 hi, 0,0 si, 0,0 st  
KiB Mem: 3731568 total, 3254024 used, 477544 free, 140372 buffers  
KiB Swap: 5771308 total, 96 used, 5771212 free. 2286732 cached Mem  


| PID   | USER  | PR | NI | VIRT    | RES    | SHR    | S | %CPU | %MEM | TIME+   | COMMAND     |
|-------|-------|----|----|---------|--------|--------|---|------|------|---------|-------------|
| 545   | root  | 20 | 0  | 227728  | 47416  | 19916  | S | 7,0  | 1,3  | 4:50.29 | Xorg        |
| 2895  | vadim | 20 | 0  | 1767760 | 272484 | 61376  | S | 5,6  | 7,3  | 8:01.86 | gnome-shell |
| 848   | vadim | 20 | 0  | 1788488 | 291192 | 131756 | S | 2,3  | 7,8  | 3:33.24 | soffice.bin |
| 1460  | vadim | 20 | 0  | 336620  | 26912  | 21776  | S | 0,7  | 0,7  | 0:01.03 | gnome-scre+ |
| 718   | root  | 20 | 0  | 940056  | 9492   | 8188   | S | 0,3  | 0,3  | 0:16.08 | teamviewerd |
| 856   | root  | 20 | 0  | 0       | 0      | 0      | S | 0,3  | 0,0  | 0:05.14 | kworker/0:0 |
| 1239  | root  | 20 | 0  | 0       | 0      | 0      | S | 0,3  | 0,0  | 0:03.48 | kworker/1:3 |
| 1457  | vadim | 20 | 0  | 25740   | 2952   | 2480   | R | 0,3  | 0,1  | 0:00.10 | top         |
| 19496 | vadim | 20 | 0  | 178456  | 4648   | 4140   | D | 0,3  | 0,1  | 0:00.17 | dconf-serv+ |


```

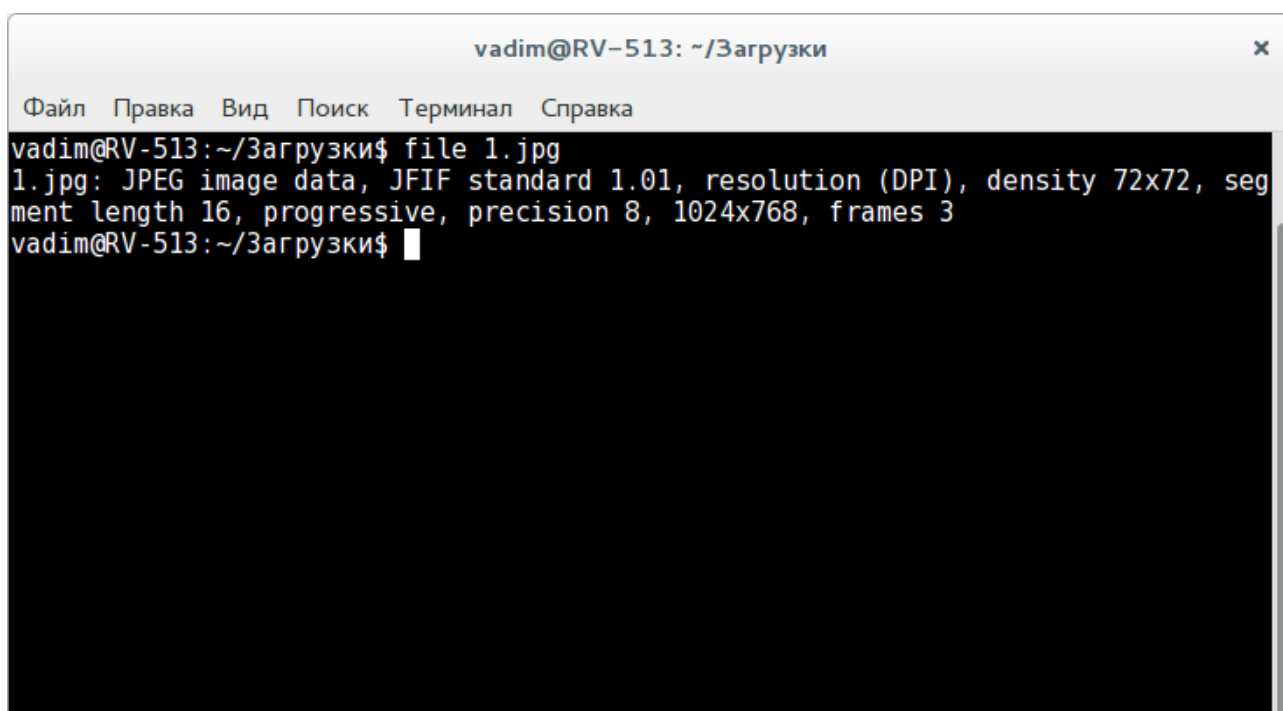
kill. Эта команда служит для принудительного завершения процессов. Нужно ввести **kill PID_процесса**. PID процесса можно узнать, введя **top**.

xkill. Ещё одна команда для завершения процессов. Введите её, затем щёлкните по тому окну, которое нужно закрыть.

`killall`. Убивает процессы с определённым именем. К примеру, `killall firefox`.

`top`. Отображает перечень запущенных процессов, сортируя в зависимости от потребления ресурсов CPU. Своего рода терминальный «Системный монитор».

Команды Linux для управления файлами



```
vadim@RV-513: ~/Загрузки
Файл  Правка  Вид  Поиск  Терминал  Справка
vadim@RV-513:~/Загрузки$ file 1.jpg
1.jpg: JPEG image data, JFIF standard 1.01, resolution (DPI), density 72x72, segment length 16, progressive, precision 8, 1024x768, frames 3
vadim@RV-513:~/Загрузки$
```

Просмотр и изменение файлов

`cat`. Когда команда используется с одним текстовым файлом (вот так: `cat путь_к_файлу`), она отображает его содержимое в окне терминала. Если указать два и больше файлов, `cat путь_к_файлу_1 путь_к_файлу_2`, она

склеит их. Если ввести `cat путь_к_файлу_1 > новый_файл`, она объединит содержимое указанных файлов в новый файл.

`chmod`. Позволяет изменять права доступа к файлу. Может пригодиться, если вы хотите внести изменения в системный файл.

`chown`. Изменяет владельца файла. Следует выполнять с правами суперпользователя.

`file`. Выводит информацию об указанном файле.

`nano`. Открывает простой текстовый редактор. Можно создать новый текстовый файл или открыть существующий: `nano путь_к_файлу`.

`rename`. Переименовывает файл или несколько файлов. Команду можно использовать и для массового переименования файлов по маске.

`touch`. Изменяет дату последнего открытия или модификации указанного файла.

`wget`. Загружает файлы из интернета в терминальную папку.

`zip`. Распаковывает и сжимает архивы.

Создание и удаление файлов и папок

`mkdir`. Создаёт новую папку в текущей терминальной папке или в указанной папке: `mkdir путь_к_папке`.

`rmdir`. Удаляет указанную папку.

`rm`. Удаляет файлы. Может удалить как отдельный файл, так и группу, соответствующую определённым признакам.

Копирование и перемещение файлов

`cp`. Создаёт копию указанного файла в папке терминала: `cp путь_к_файлу`. Или вы можете указать назначение `cp путь_к_файлу путь_для_копии`.

`mv`. Перемещает файл из одной папки в другую. Вы можете указать имя для перемещаемого файла. Забавно, но в Linux эта команда может использоваться и для переименования файлов. Просто укажите ту же папку, где находится файл, и другое имя.

Поиск файлов

`find`. Поиск файлов по определённым критериям, таким как имя, тип, размер, владелец, дата создания и модификации.

`grep`. Поиск текстовых файлов, содержащих определённые строки. Критерии очень гибко настраиваются.

`locate`. Поиск файлов и папок, чьи названия подходят запросу, и отображение их путей в файловой системе.

Команды Linux для работы с разделами

```
vadim@RV-513: ~  
Файл Правка Вид Поиск Терминал Справка  
vadim@RV-513:~$ lsblk  
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT  
sda      8:0    0 298,1G  0 disk  
├─sda1    8:1    0  46,7G  0 part /  
├─sda2    8:2    0    1K    0 part  
├─sda5    8:5    0   3,7G  0 part [SWAP]  
└─sda6    8:6    0 247,7G  0 part /home  
sr0      11:0    1 1024M  0 rom  
zram0    254:0   0   911M  0 disk [SWAP]  
zram1    254:1   0   911M  0 disk [SWAP]  
vadim@RV-513:~$
```

lsblk. Эта команда демонстрирует, какие диски есть в вашей системе и на какие разделы они поделены. Также команда отображает имена ваших разделов и накопителей, в формате `sda1`, `sda2` и так далее.

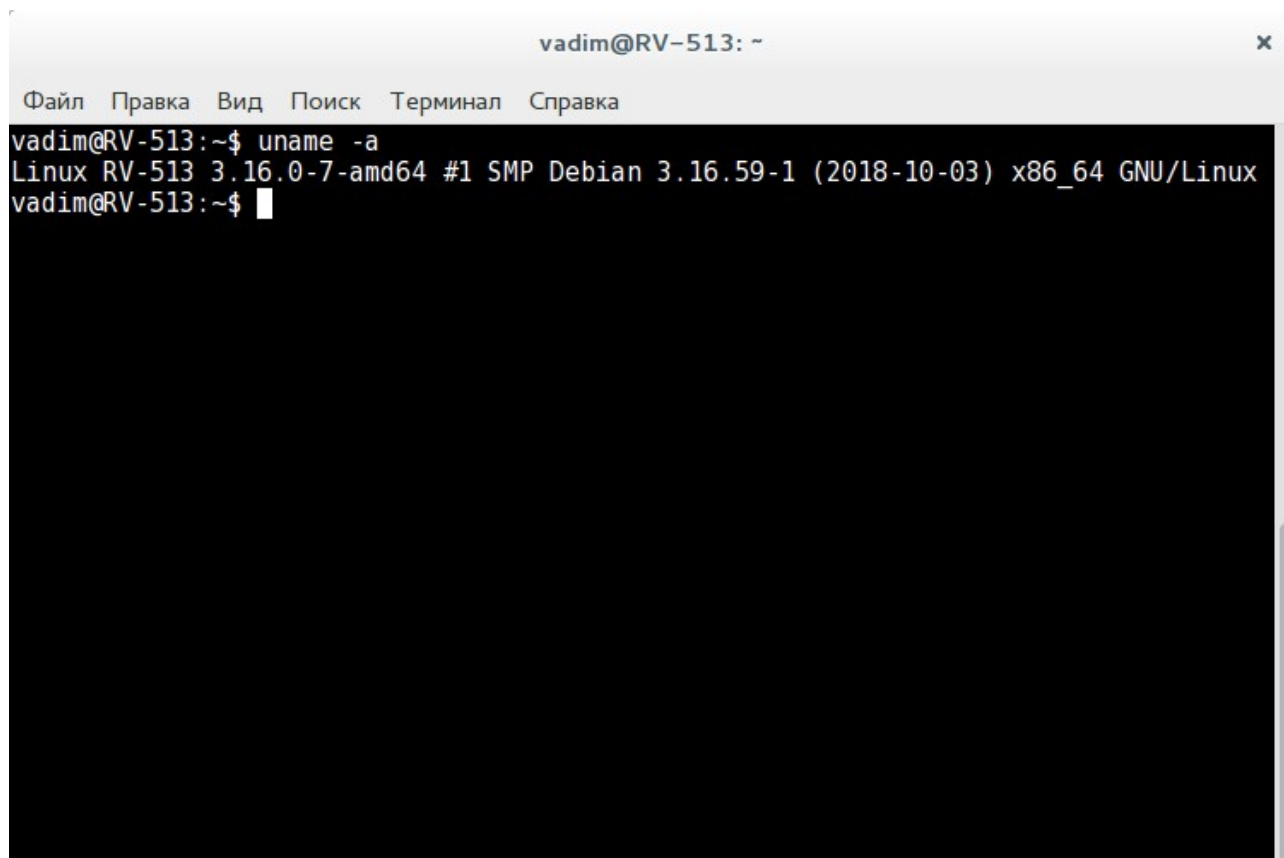
mount. Монтирует накопители, устройства или файловые системы Linux, чтобы вы могли с ними работать. Обычно устройства подключаются автоматически, как только вы щёлкнете по ним в файловом менеджере. Но иногда может понадобиться примонтировать что-то вручную. Вы можете подключать что угодно: диски, внешние накопители, разделы и даже ISO-образы. Эту команду нужно выполнять с правами суперпользователя. Чтобы примонтировать имеющийся диск или раздел, введите `mount sdX`.

umount. Демонтирует файловые системы. Команда `umount sdX` отключит файловую систему внешнего носителя, чтобы вы могли извлечь его.

dd. Эта команда копирует и преобразовывает файлы и разделы. У неё множество различных применений. Например, `dd if=/dev/sda of=/dev/sdb` сделает точную копию раздела `sda` на разделе `sdb`. `dd if=/dev/zero of=/dev/sdX` затрёт содержимое указанного носителя нулями,

чтобы информацию было невозможно восстановить. А `dd if=~Downloads/ubuntu.iso of=/dev/sdX bs=4M` сделает загрузочный носитель из скачанного вами образа с дистрибутивом.

Команды Linux для управления системой



```
vadim@RV-513: ~  
Файл Правка Вид Поиск Терминал Справка  
vadim@RV-513:~$ uname -a  
Linux RV-513 3.16.0-7-amd64 #1 SMP Debian 3.16.59-1 (2018-10-03) x86_64 GNU/Linux  
vadim@RV-513:~$
```

`df`. Отображает объём вашего диска, и сколько на нём осталось свободного места.

`free`. Отображает объём доступной и занятой оперативной памяти.

`uname`. Отображает сведения о системе. Если ввести `uname`, терминал сообщит только Linux. Но команда `uname -a` выводит сведения об имени компьютера и версии ядра.

`uptime`. Сообщает, как долго запущена ваша система.

`whereis`. Отображает расположение исполняемого файла нужной программы.

`whoami`. Называет имя пользователя.

Команды Linux для управления пользователями

`useradd`. Регистрирует нового пользователя. Введите `useradd` имя_пользователя, и пользователь будет создан.

`userdel`. Удаляет учётную запись и файлы пользователя.

`usermod`. Изменяет учётную запись пользователя. Может переместить домашнюю папку пользователя или назначить дату, когда учётная запись будет заблокирована.

`passwd`. Изменяет пароли учётных записей. Обычный пользователь может изменить пароль только своей учётной записи, суперпользователь может изменить пароль любой учётной записи.

Команды Linux для управления сетью

`ip`. Многофункциональная команда для работы с сетью. Команда `ip address show` выводит сведения о сетевых адресах, `ip route` управляет маршрутизацией и так далее. Давая команды `ip link set ethX up`, `ip link set ethX down`, можно включать и выключать соединения. У команды `ip` много применений, так что перед её использованием лучше ознакомиться с руководством или ввести `ip --help`

`ping`. Показывает, подключены ли вы к сети, и помогает определить качество связи.

Это далеко не все команды Linux. Если вам нужно узнать в деталях параметры и способы применения команд Linux, вы можете воспользоваться встроенным руководством.

Наберите `man ваша_команда` или `ваша_команда --help`.

Устраняем тормоза системы при операциях ввода-вывода

| Дескриптор | Название | Описание |
|------------|---------------------|--------------------------|
| 0 | <code>stdin</code> | Стандартный ввод |
| 1 | <code>stdout</code> | Стандартный вывод |
| 2 | <code>stderr</code> | Стандартный вывод ошибок |

Баг #12309 - самый знаменитый баг в ядре Linux, довольно долго досаждавший пользователям Linux на десктопах. Сам баг исправлен в ядре Linux 3.3, но симптомы, похожие на таковые при 12309, могут проявляться на некоторых конфигурациях до сих пор. В сети можно найти много инструкций по лечению этих симптомов.

На самом деле 12309 — это не один, а несколько багов, смешанных в кучу. Можно выделить следующие случаи появления:

- при копировании больших объемов данных с диска на диск (или с раздела на раздел одного диска);
- при нехватке ОЗУ (и, соответственно, диком своппинге);
- при копировании на USB-девайсы;
- при использовании зашифрованных разделов;

Соответственно, фиксы тоже будут разные.

Сначала тест: восприимчива ли ваша система к 12309? введите в терминале:

```
dd if=/dev/zero of=/tmp/test bs=1M count=1M
```

и понаблюдайте за отзывчивостью системы.

Оптимистическое выделение памяти

Возможно, в научных программах какого-нибудь толка позволить выделить терабайт ОЗУ при наличии 3 Гб физической памяти и считается приемлемым, но на десктопе, где много процессов должны спокойно сосуществовать, такой расклад неприемлем — зажавшаяся программа спокойно вытеснит все остальное, после чего система практически остановится. Хуже всего то, что суть бага 12309 в том, что ядро принимает решения о том, какие страницы вытеснять, мягко говоря, не оптимально, а чинить это долго, мучительно, и не в каждой ситуации решение будет приемлемым.

То есть стратегии вытеснения страниц нужно делать переключаемыми, а это работы на много времени не столько реализации, сколько тестирования.

Пока что же можно сделать так, чтобы на явно дикие запросы `malloc()` отвечал решительным отказом, и чтобы зажатость программы определялась на этапе выделения памяти, а не тогда, когда программа радостно вывалит туда пару гигабайт данных.

Для этого нужно прописать в `/etc/sysctl.conf`

`vm.overcommit_memory = 2`

Максимум памяти, который можно будет выделить, будет равен в сумме объему свопа + некоторому проценту физической памяти. Этот процент по умолчанию равен 50, но можно его несколько увеличить. Во всяком случае, я выставил его в 80 и пока что катастроф нет.

`vm.overcommit_ratio = 80`

Подкачка нужна

Некоторые люди полагают, что если отключить своп, то 12309 исчезнет. А вот как бы не так. Своп (swap) — это хранилище анонимных страниц памяти. Код исполняемых программ и всяких библиотек не анонимен и по умолчанию не изменяем. В то время как на 32-битных системах исполняемый код зачастую зависит от позиции (начального адреса), что приводит к тому, что, во-первых, динамический линковщик проводит вычисление смещений каждый раз при загрузке и, соответственно, страницы кода анонимны (это несет с собой недостаток в виде наличия нескольких копий одной и той же библиотеки, но и преимущество в виде невозможности вытеснить страницы кода для освобождения памяти), то на 64-битных системах практически весь код линкуется в независимом от позиции виде (PIC).

Это означает, что, во-первых, загрузка такого кода — это фактически всего лишь mmap() на исполняемый файл, во-вторых, можно держать только одну копию страниц кода на каждый процесс, сколько раз его ни загружай. Это достоинства. Недостаток в том, что даже при отсутствии свопа ядро может в случае нехватки освободить память за счет страниц со спящим кодом. Когда код надо будет выполнять, поднимать его надо будет с диска, и хорошо бы тогда иметь место в очереди ввода-вывода, а то система встанет в неудобную позу, причем надолго.

Так как ядро, вообще-то, достаточно умное, чтобы сначала при возможности сбросить в своп анонимные страницы спящих процессов, то своп лучше все же иметь. Чаще всего доставать этих страниц надо будет меньше (особенно в случае Xorg и его драйверов, не путать с драйверами ядра).

Уменьшение размеров дисковых буферов

С одной стороны, отдавать под дисковые буферы практически всю свободную память — здравая идея. А с другой стороны, чем больше ОЗУ, на самом деле, тем сильнее это способно ударить в критической ситуации.

Это работает вот как. У ядра есть буфер файловой системы. Мы пишем много данных. Этот буфер заполняется грязными страницами, а потом выполняется

системный вызов `sync()` и буфер сбрасывается на носитель. Чем больше буфер, тем больше данных надо будет сбрасывать. Все бы ничего, да вот когда кому-то вдруг вздумается выделить себе памяти, в первую очередь будут сбрасываться все эти буферы, и если при этом вдруг надо будет закачать страницы с исполняемым кодом, им опять-таки придется ждать в очереди. Опять слайдшоу, с возможной цепной реакцией.

То есть, кеш на чтение — это ничего так, а слишком большой кеш на запись способен встать поперек горла в критических случаях.

Есть еще одна неприятная особенность, связанная трудно сказать, с чем — возможно, с реализацией DMA, но вполне возможно, что не с ней, или не только с ней. Берем какой-нибудь медленный для записи носитель, типа той же USB-флешки, и пробуем записать на него данных побольше, фильм какой или что-то вроде. Мы увидим, что происходит это рывками — сначала заполняется буфер, сколько влезет, а потом весь сбрасывается, потом весь заполняется... и так далее. При этом суммарно потраченное время почему-то ощутимо больше, чем как если бы мы примонтировали носитель с -o `sync`, а скорость записи на, собственно, носитель невообразимо мала.

Но если уменьшить порог количества грязных блоков, после которого начнется их сброс на носитель, не до сверхмалых величин, но все же — это позволит проводить зачитку данных из источника и запись на носитель параллельными DMA-трансферами. Я у себя выставил этот объем равным 2 мегабайтам, что, с одной стороны, уменьшает количество перезаписей в случае частой смены маленьких файлов и значительно увеличивает скорость переноса больших объемов данных. Возможно, если поиграться размером, можно найти оптимальное быстродействие, но не думаю, что буфер больше 16 мегабайт будет эффективным.

```
echo 2097152 >/proc/sys/vm/dirty_bytes
```

```
echo 2097152 >/proc/sys/vm/dirty_background_bytes
```

Для сохранения после перезагрузки, прописать в ***/etc/sysctl.conf***

```
vm.dirty_bytes = 2097152
```


vm.dirty_background_bytes = 2097152

Стоит учесть, что кеши чтения файловой системы будут все так же занимать почти все свободное ОЗУ, но при этом запись будет осуществляться, как только блоков, помеченных на запись, наберется на 2 мегабайта.

Значение *dirty_bytes* должно делиться на 4096 нацело.

В результате, даже при переваривании больших объемов данных, система не заикается. Может затормозить сам процесс, который выделяет память, но отзывчивость системы не теряется.

Кеш файловых систем

Иногда бывает такая проблема: у вас относительно новое ядро, тесты с *dd* проходят на ура, а вот когда надо много маленьких файловиков создавать/менять/убивать, например, командой *dpkg*, система встает колом и не может даже регистрами пошевелить.

Проблема в том, что мета информация файловых систем тоже особым образом кешируется на уровне VFS. И это как раз хороший, годный кеш — мета информация разбросана по диску очень в случайном порядке. Поэтому надо сказать, что если кеши и нужно сбрасывать, то кеш файловой системы - в последнюю очередь.

vm.vfs_cache_pressure = 50

По умолчанию оно почему-то 100, что просто безумно на десктопных нагрузках.

Прочие решения:

Установить параметр *swappiness* равным 10 или 5, чтобы подкачка задействовалась только при исчерпании 90 и 95% памяти соответственно:

sudo nano /etc/sysctl.conf

Добавить в конец

vm.swappiness = 10

Сохранить и выполнить

sudo sysctl -p

Сменить планировщик ввода-вывода. В такой ситуации рекомендуют использовать BFQ, но он не в основной ветке ядра, и его нужно добавлять в ядро самому (в некоторых дистрибутивах, например Calculate Linux, BFQ по умолчанию). В остальных же случаях можно сменить планировщик на лету.

Перевесить системные прерывания на одно ядро (на многоядерном процессоре) скриптом:

#!/bin/sh

for interruption in `grep usb /proc/interrupts | awk '{print \$1}' | sed 's/://g'` ; do

echo 1 > /proc/irq/\${interruption}/smp_affinity;

done

При самостоятельной сборке ядра, задействовать 100Hz таймер ядра и опцию No Force Preemption (Server) mode.

Выставить приоритет ionice для ядра 1 (realtime) для пространства пользователя (userspace) - 3.

Установить ядро с патчами для реализации режима реального времени (linux-lowlatency, есть по умолчанию в репозиториях большинства дистрибутивов).

Подводя итог, можно отметить: самого бага давно нет, но на некоторых конфигурациях могут возникнуть его симптомы, по совершенно разным причинам, от нехватки памяти до аппаратных проблем с дисками. Пугаться этого не стоит, ибо например в Windows, эта проблема ещё более серьёзна, а

самое главное - никак не решаемая. Система может с лёгкостью встать колом при копировании больших объёмов данных, и её придётся перезагружать кнопкой Reset.

Браузер



Как бы не была защищена система, остаётся уязвимым то, через что вы сёрфите по интернету. Это браузер. Опишу настройку стандартного для большинства дистрибутивов, браузера **Mozilla Firefox**

1. Установите расширение [uBlock](#) (или AdBlock, если вы привыкли к нему). Это отсеет множество рекламных баннеров, которые очень часто бывают вредоносными. В настройках расширения включите подписки RuList+EasyList, а также фильтры вредоносных программ (Malware).
2. Установите расширение NoScript, которое будет отключать расширения JavaScript, которые являются самыми опасными в плане уязвимостей. Это может нарушить функционал сайта, поэтому вы можете включить только нужные JavaScript'ы, для работы сайта.
3. В адресной строке, наберите about:config. Откроется редактор расширенных настроек браузера. Наверху будет поисковая строка. Делаем так:

browser.safebrowsing.enabled = false, browser.safebrowsing.malware.enabled = false - Отключают передачу информации о посещаемых веб-сайтах (система

для борьбы с фишингом).

browser.search.suggest.enabled = false - Отключает передачу текста, набираемого в окне поиска, поисковой системе без явного подтверждения со стороны пользователя.

dom.enable_performance = false - Отключает передачу браузером информации о времени начала и окончания загрузки страницы. Анализ этих данных позволяет определить факт использования прокси-сервера.

dom.storage.enabled = false - Отключает возможность сайтов хранить некоторые настройки (нечто похожее на Cookies).

general.useragent.override = "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1.7) Gecko/200091221 Firefox/3.5.7" - Подмена User Agent. Если выставить наиболее распространённое значение, то это поможет снизить уникальность пользователя.

network.dns.disablePrefetch = true - Запрещает предварительное разрешение имён DNS для всех ссылок на веб-странице (пока пользователь сам не нажмёт на ссылку). Это может привести к утечке DNS-трафика при работе через анонимизирующий прокси-сервер.

network.cookie.cookieBehavior = 1 - Запрещает Cookies для сторонних сайтов.

network.http.sendRefererHeader = 0 network.http.sendSecureXSiteReferrer = false - Отключают передачу заголовка *HTTP referer* для обычного и зашифрованного соединения соответственно.

FlashPlayer - самое уязвимое и опасное, что может находиться в вашем браузере. Если вы используете его только для просмотра видео на **YouTube**, то он вам не нужен. Можно переключиться на **HTML 5** проигрыватель (который, правда, пока не позволяет смотреть в Firefox видео в 1080p).

Для этого в ***about:config***

- ***media.mediasource.enabled;true***
- ***media.mediasource.mp4.enabled;true***

- media.mediasource.webm.enabled;true**
- media.gstreamer.enabled;true**
- media.fragmented-mp4.exposed;true**
- media.fragmented-mp4.ffmpeg;true**
- media.fragmented-mp4.gmp;true**
- media.fragmented-mp4.use-blank-decoder>false**

Далее перейдите по [этому адресу](#) и переключите проигрыватель.

Если флеш всё-таки нужен, то делаем так:

Заходим на сайт Adobe на страницу [диспетчера параметров](#). На вкладке «Глобальные параметры хранения» сократите до минимума разрешённый объём пространства на диске для хранения информации и запретите стороннему flash содержимому сохранять данные на компьютер.

Руткиты

Если у вас есть подозрение на наличие руткита, воспользуйтесь утилитой **rkhunter**:

```
sudo apt-get install rkhunter  
sudo rkhunter -c --sk
```

Утилита проверит систему на наличие руткитов, и если найдёт - укажет где он сидит, и его можно будет удалить. Подробный лог проверки находится в **/var/log/rkhunter.log**. Утилиту нужно периодически обновлять. Можно создать задание для cron, которое будет ежедневно выполнять обновление утилиты и сканирование на руткиты:

```
sudo nano /etc/cron.daily/rkhunter.sh
```

```
#!/bin/bash
```

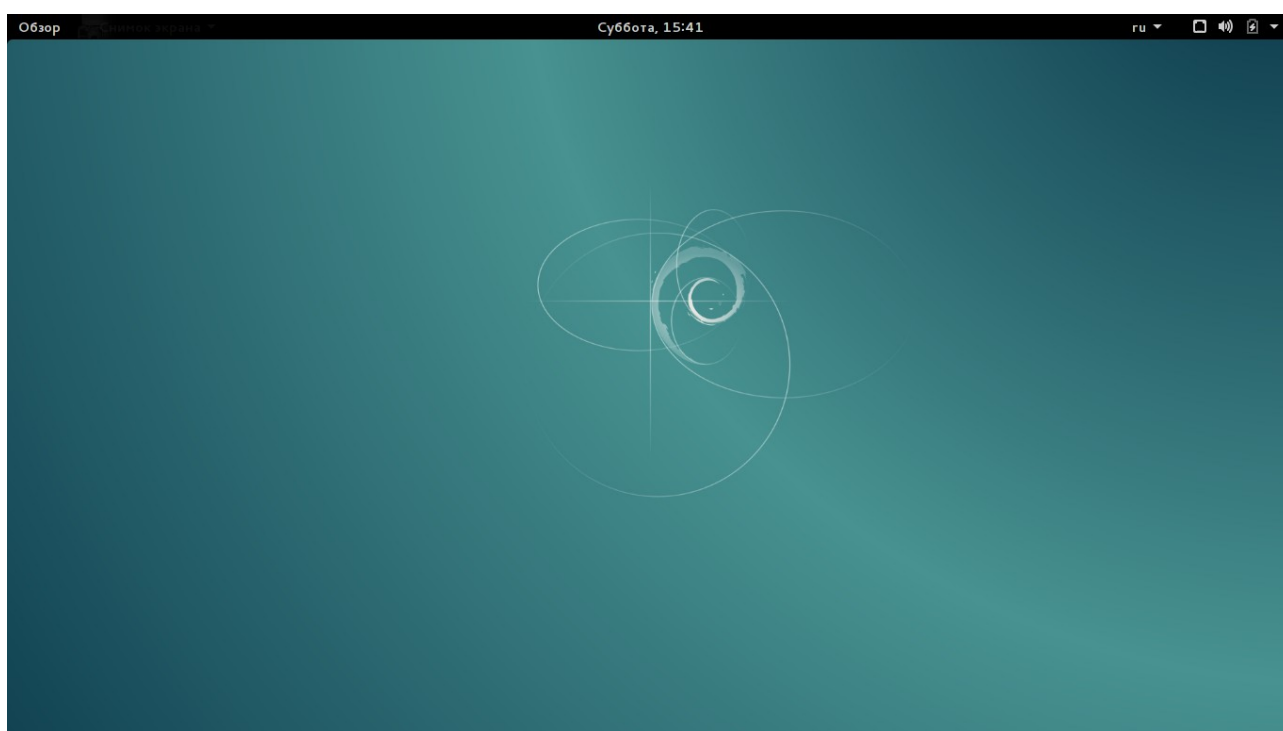
```
/usr/bin/rkhunter --update
```

```
/usr/bin/rkhunter -c -cronjob 2>&1 | mail -s "RKHunter Scan Results" ваш.e-mail@.ru
```

```
sudo chmod +x /etc/cron.daily/rkhunter.sh
```

После проверки, результат сканирования отправится на адрес электронной почты, который вы указали.

Настройка рабочего стола Gnome

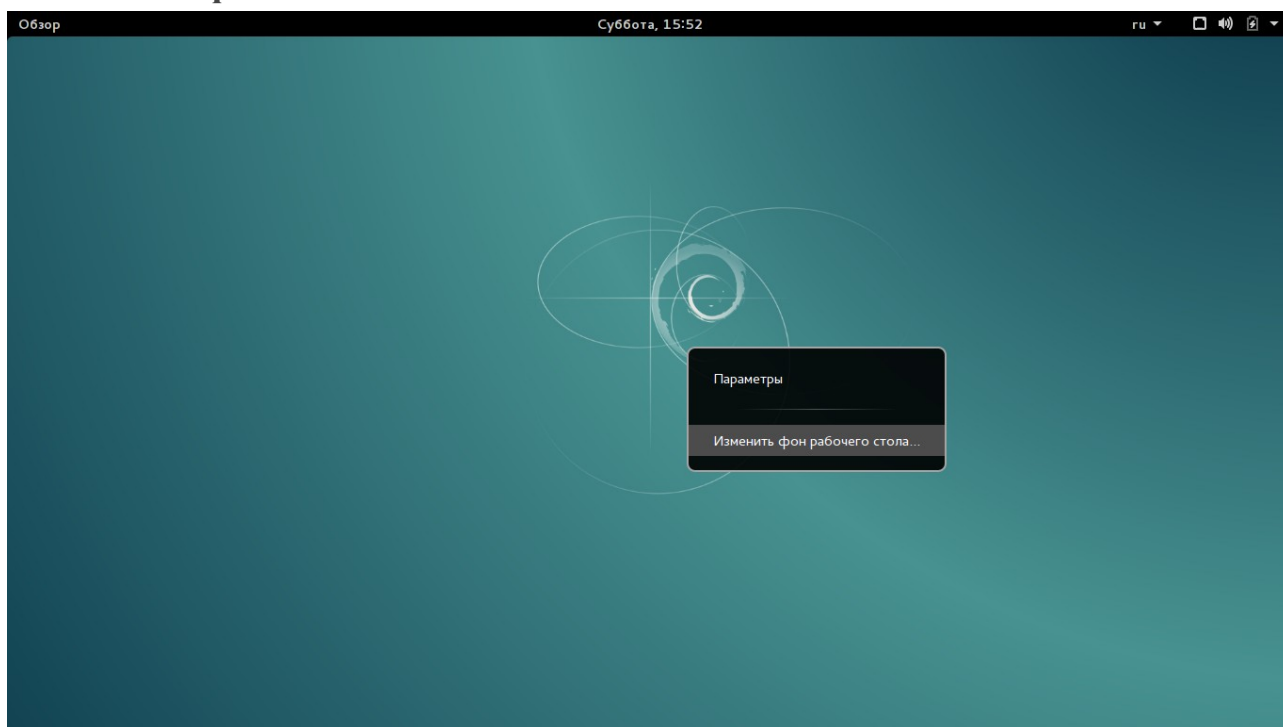


Окружение рабочего стола **Gnome** используется очень в различных дистрибутивах по умолчанию и очень популярно среди многих пользователей. После установки система выглядит очень красиво, так что в большинстве случаев сразу после установки окружения вы можете с ним работать.

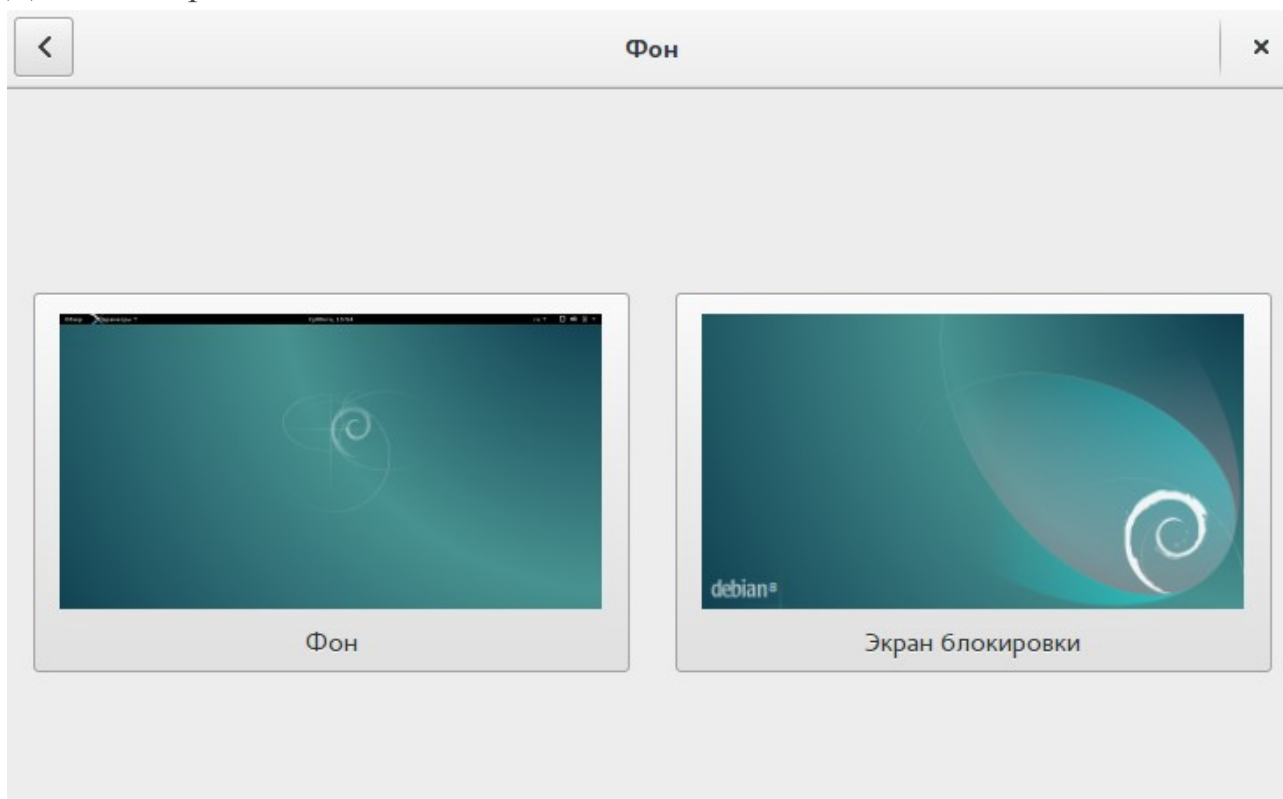
Мы рассмотрим как выполняется настройка Gnome 3, какие утилиты для этого нужны, а также что можно изменить в этом окружении. Начнем с самых простых задач.

ФОН РАБОЧЕГО СТОЛА

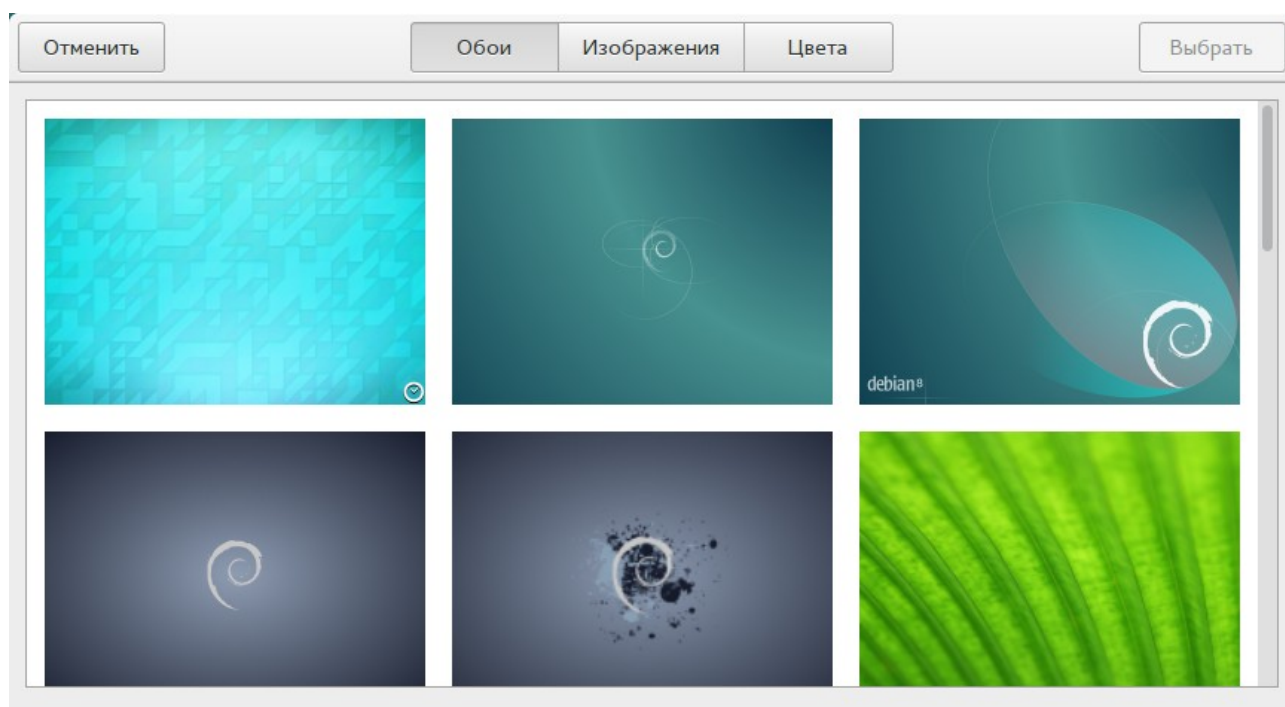
Первое что мы сделаем в новой системе - это изменим фон рабочего стола. Для этого кликните павой кнопкой по любому месту на рабочем столе и выберите **"Изменить фон"**.



Далее выберите **"Фон"**



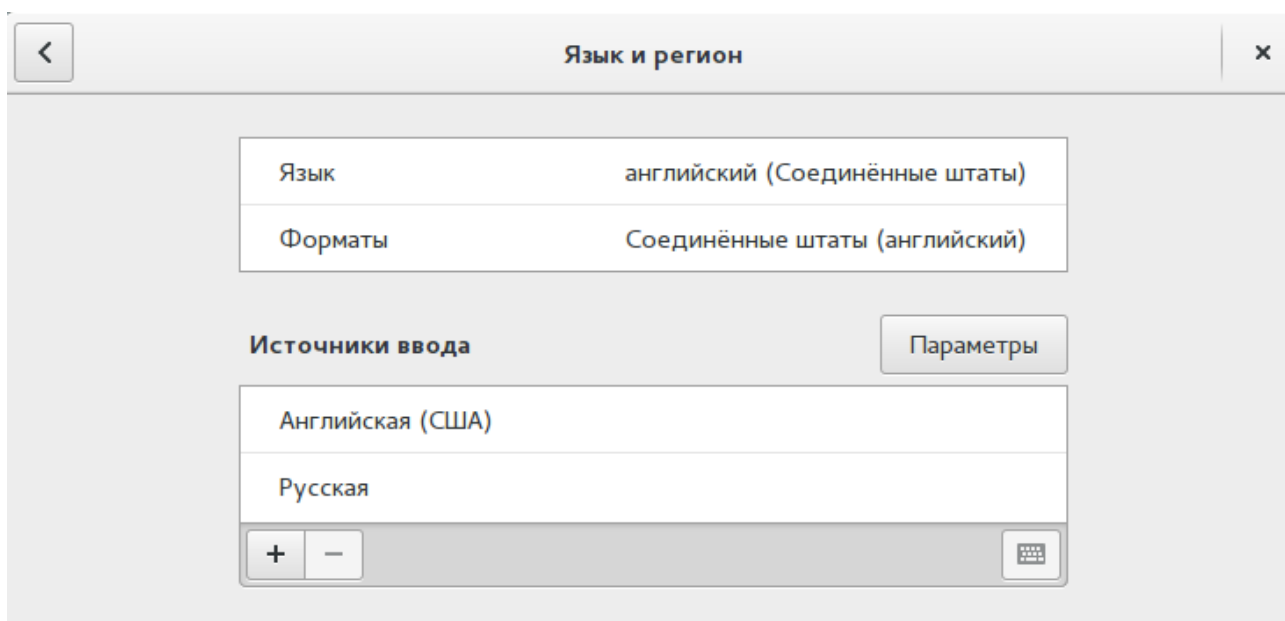
Тут вы можете выбрать одну из понравившихся картинок или загрузить свою на вкладке "**Изображение**"



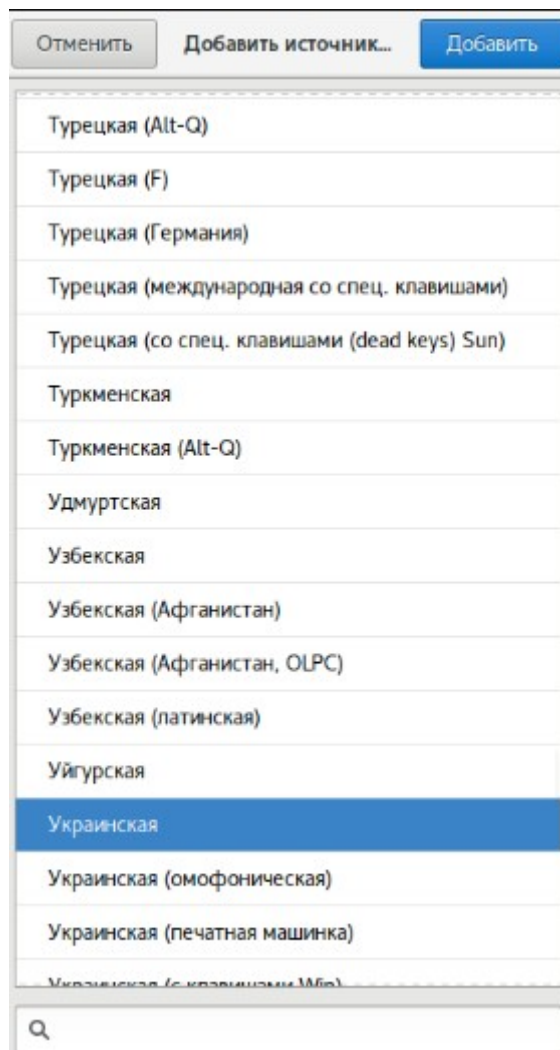
Дальше вам осталось нажать выбрать и фон будет изменен. Также можно настроить картинку для экрана блокировки в этом же окне.

РАСКЛАДКА КЛАВИАТУРЫ

По умолчанию в Gnome для русского языка есть русская и английская раскладки, они переключаются сочетанием клавиш **Super+Пробел**. Но иногда нужно добавить новую раскладку, например, украинский язык. Откройте утилиту "**Параметры**", затем выберите "**Язык и ввод**"

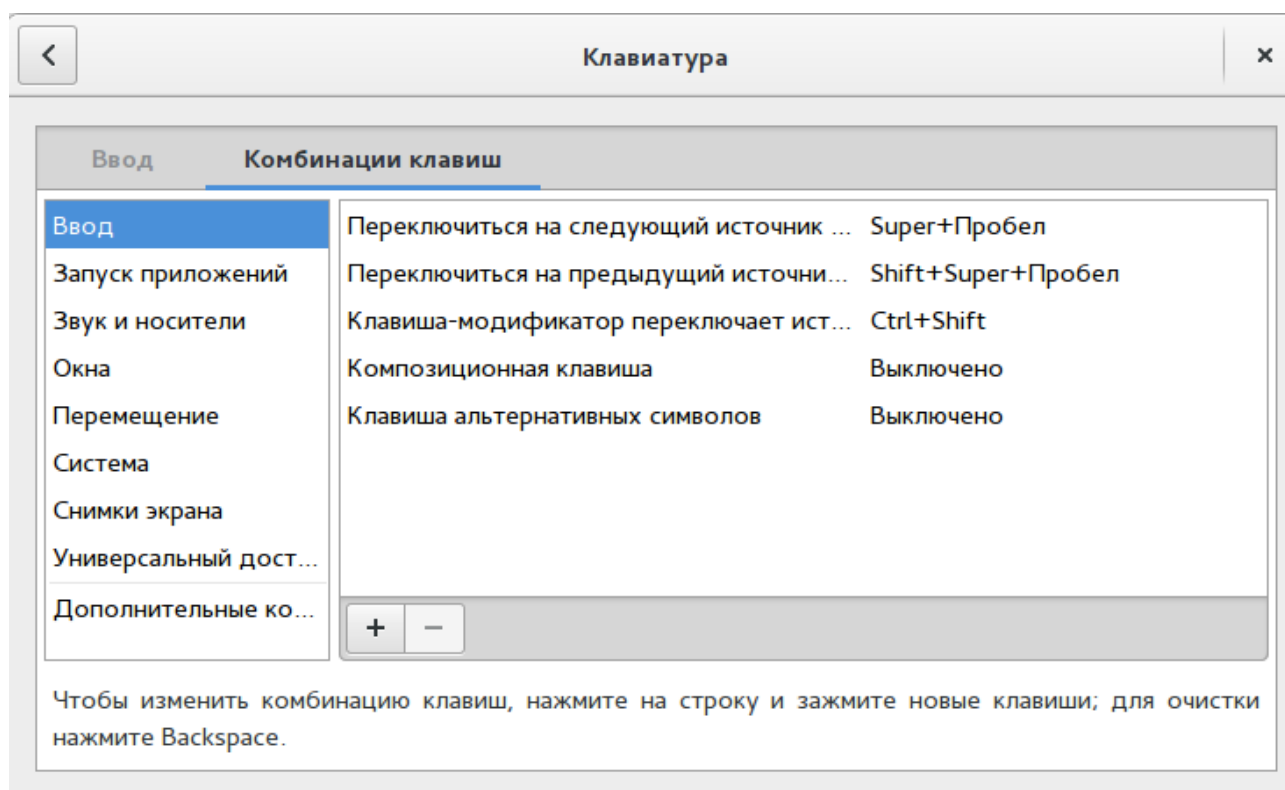


В разделе **"Источник ввода"** вы можете добавить новый язык с помощью кнопки **"+"**



После добавления он появится в списке.

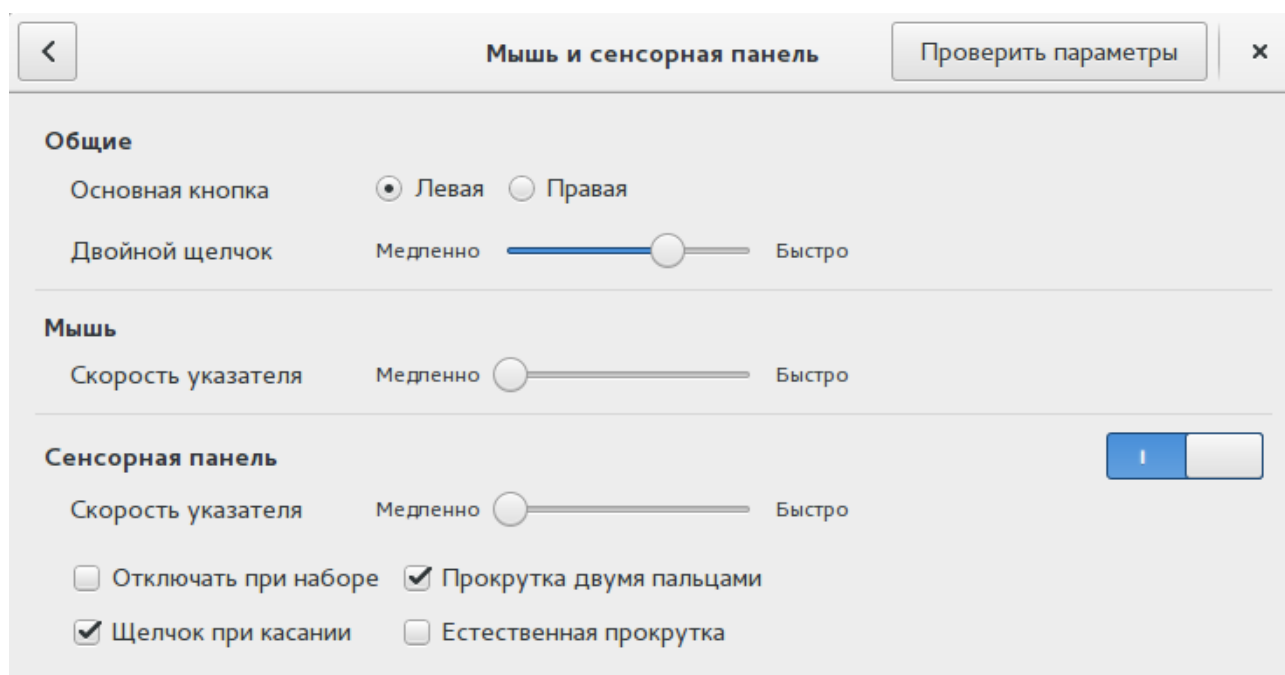
Не все привыкли переключать язык по **Super+Пробел**, я до сих пор пользуюсь **Ctrl+Shift**. Сменить клавишу для переключения очень просто. Для этого откройте утилиту "**Параметры**", затем "**Клавиатура**"



На вкладке "Ввод" можно изменить клавишу для переключения раскладки. Только не пытайтесь менять **Super+Пробел**, так оно не сработает. Вам нужен пункт "**Клавиша модификатор переключает источник ввода**". Нажмите напротив этого пункта, затем нажмите нужное сочетание

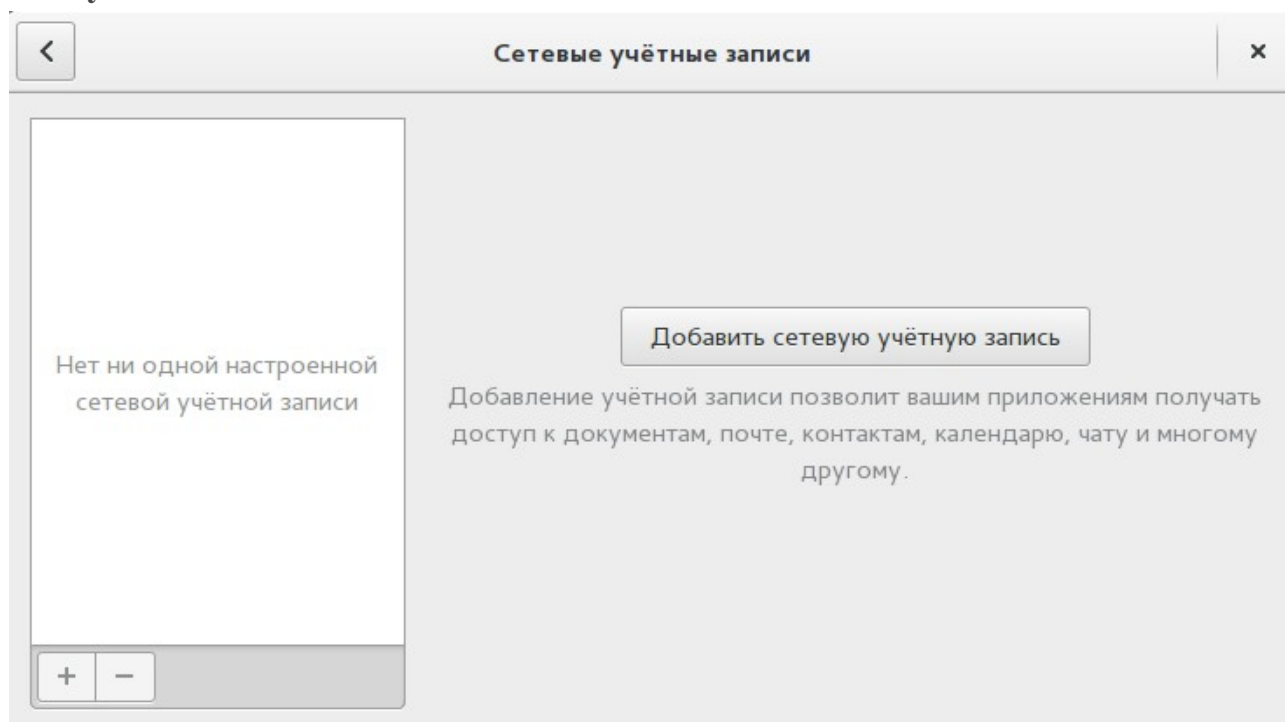
Мышь и сенсорная панель

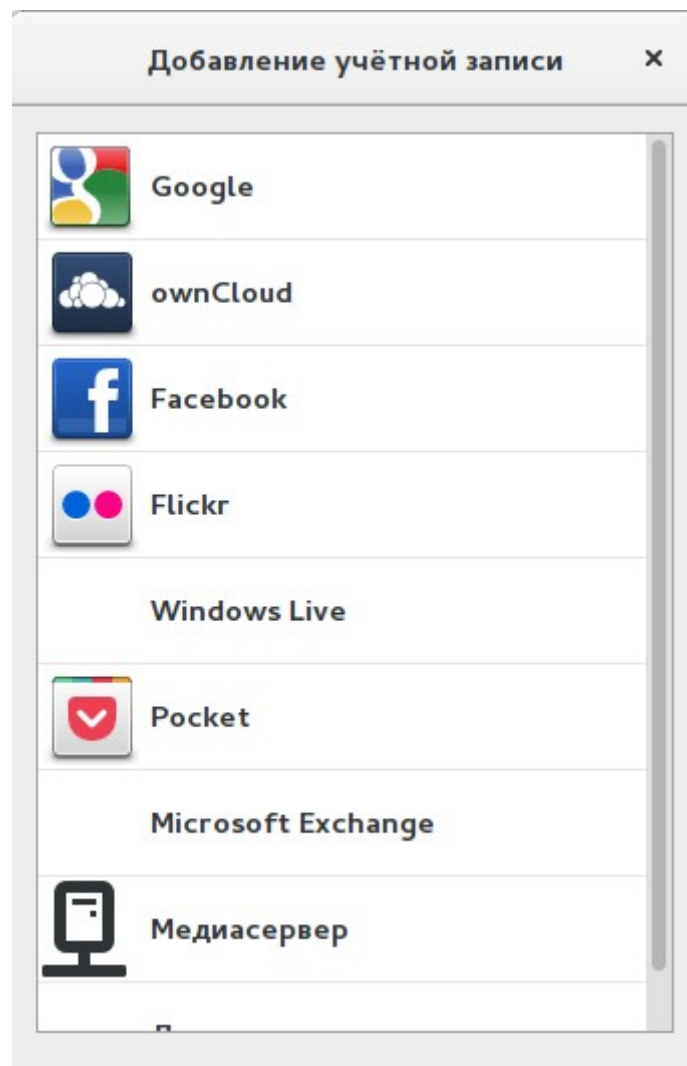
В некоторых дистрибутивах по умолчанию прокрутка двумя пальцами на тачпаде работает наоборот, т.е вы крутите вниз, а содержимое подымается в верх. Это не привычно и неудобно. Вы можете изменить такое поведение в утилите "Параметры", "**Мышь и сенсорная панель**". Найдите и отключите пункт "**Естественная прокрутка**" для сенсорной панели.

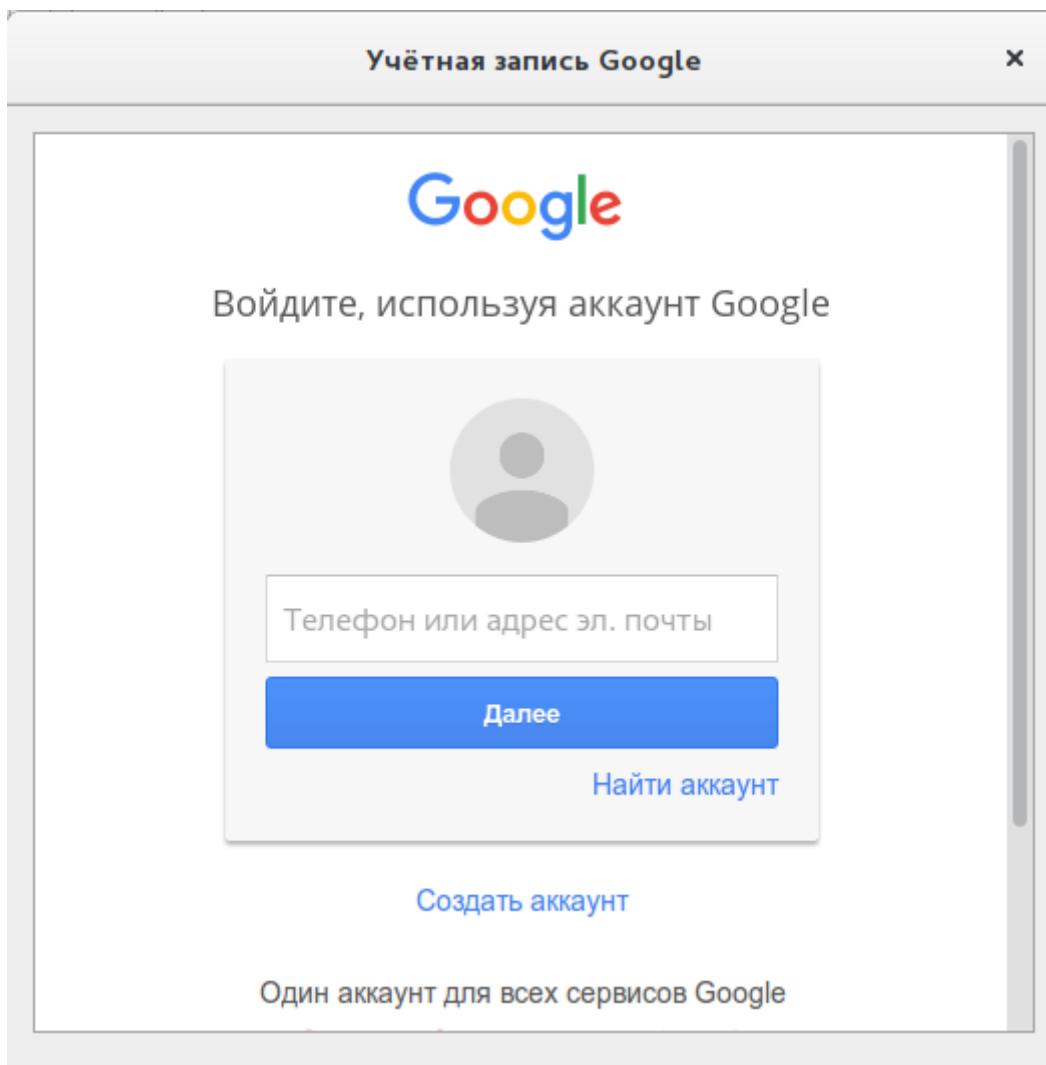


СЕТЕВЫЕ АККАУНТЫ GNOME

Gnome позволяет интегрировать рабочее окружение с различными сетевыми службами. Вы сможете работать прямо с документами, сохраненными на Google диск, получать уведомления от Facebook, получать почту Gmail, Outlook и многое другое. Чтобы связать систему с нужными аккаунтами откройте параметры и в разделе **"Сетевые учетные записи"** выберите **"Добавить аккаунт"**







Точно так вы можете добавить любой другой поддерживаемый аккаунт к системе.

ТЕМА GNOME

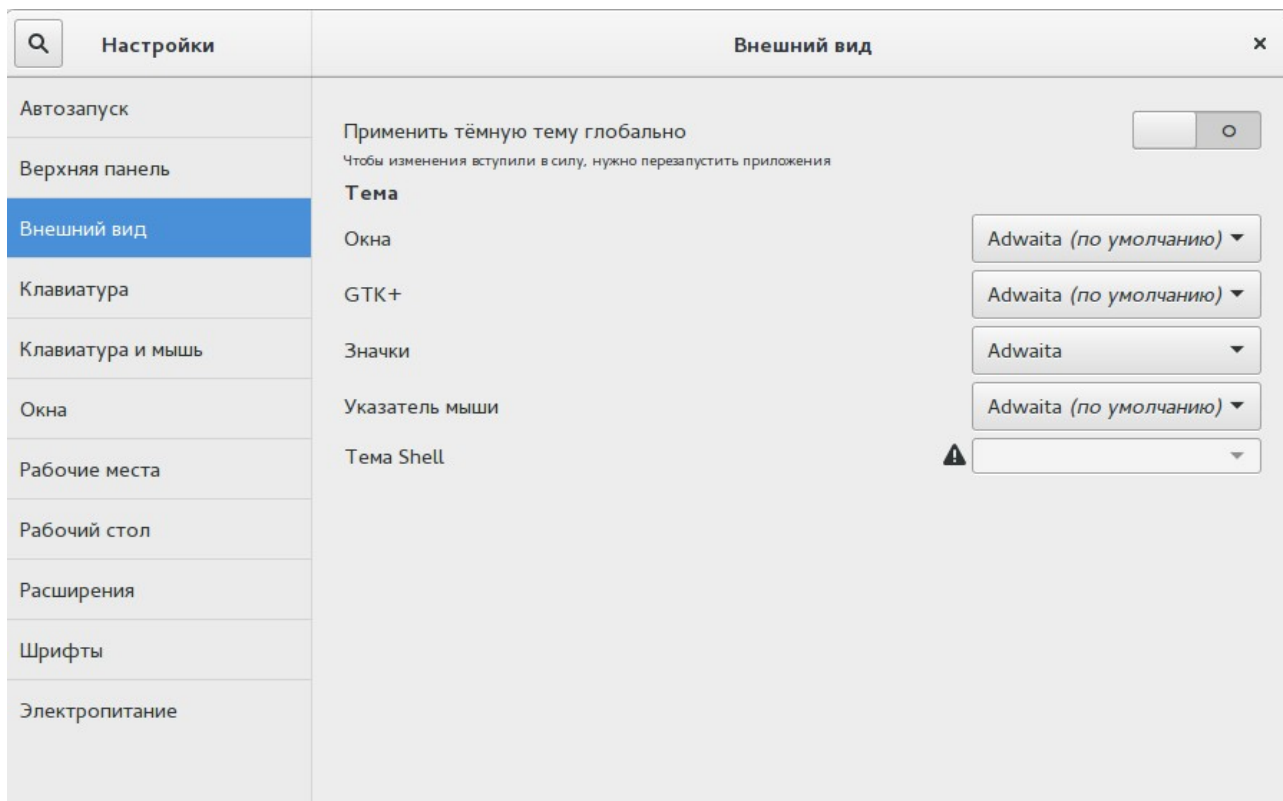
Не всем нравится тема Gnome по умолчанию, вы можете заменить ее на любую, доступную в интернете. Но только для этого нам понадобится стороннее приложение Gnome Tweak Tool. Если программа еще не установлена, вы можете ее установить:

```
sudo apt install gnome-tweak-tool
```

Теперь программу можно запустить из главного меню, она называется дополнительные настройки:



Для настройки темы перейдите на вкладку "Внешний вид". Тут вы можете настроить несколько тем, это тема GTK, которая отвечает за поведение окон, их цвет, грани и так далее, тема иконок, а также тема оболочки, которая отвечает за внешний вид оболочки в целом, панелей, и так далее.



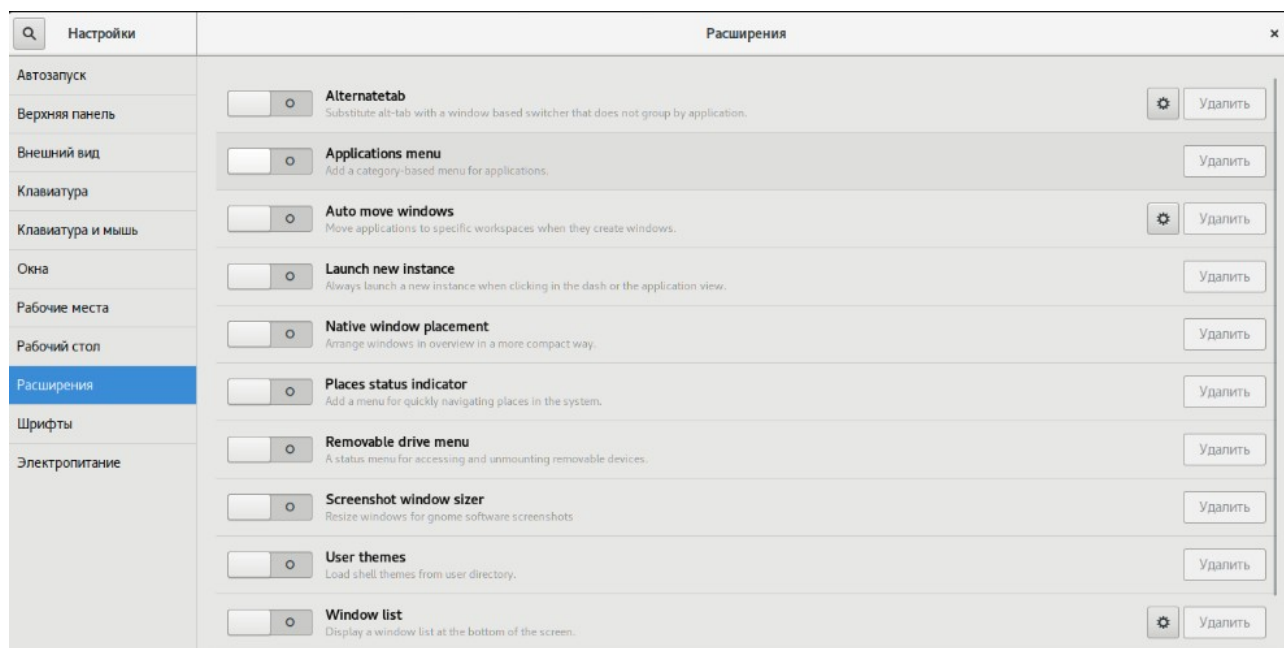
Устанавливать новые темы можно с помощью пакетного менеджера:

sudo apt install numix-gtk-theme

Также вы можете просто скопировать папку с темой, загруженной из интернета в `~/.themes/`. Добавленная тема появится в списке. Для ее активации достаточно выбрать тему.

РАСШИРЕНИЯ ОБОЛОЧКИ GNOME

В Gnome есть поддержка расширений, которые могут улучшать оболочку и дополнять ее новыми функциями. Настройка расширений выполняется тоже с помощью Gnome Tweak Tool. Просто перейдите на вкладку "Расширения". Тут вы можете выбрать и активировать некоторые из стандартных расширений:



Вот основные предустановленные расширения Gnome 3:

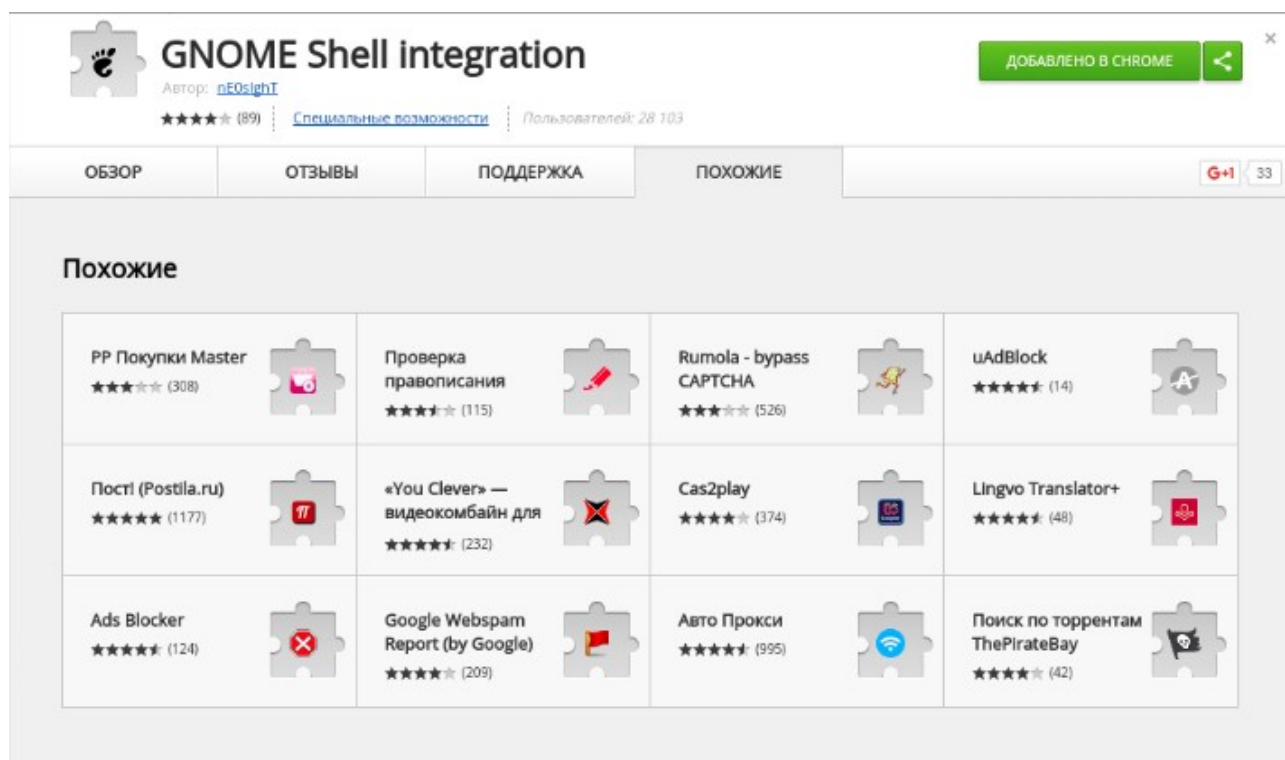
- **Alternatetab** - переключение окон по Alt+Tab в красивом стиле;
- **Appmenu** - классическое меню приложений в верхнем правом углу;
- **Places** - добавляет индикатор "Места" на панель;

- **User Themes** - позволяет использовать пользовательские темы для Shell. Вы, наверное видели, что пункт "Тема Shell" отключен, это расширение включает его;
- **WindowList** - добавляет список открытых окон на рабочий стол Gnome 3 внизу экрана.

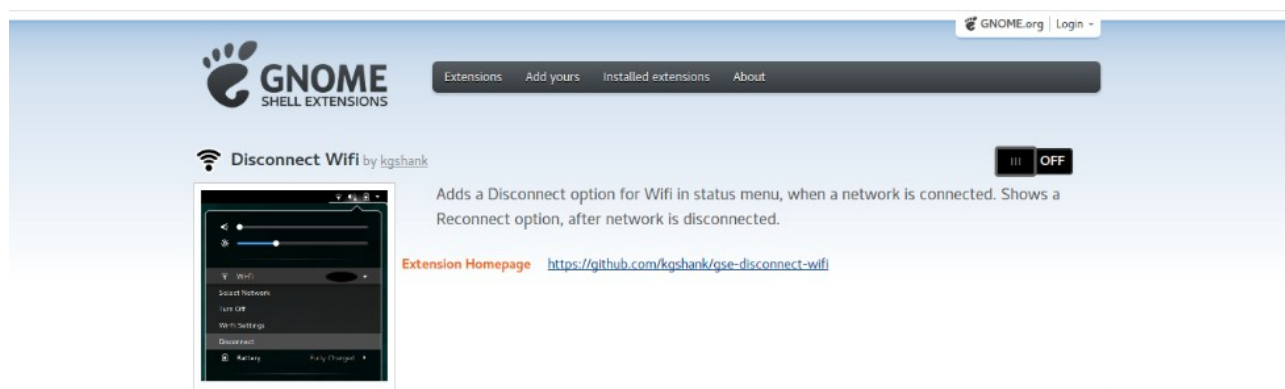
Вы можете установить множество других расширений из интернета. Все они размещены на сайте extensions.gnome.org. Но для работы с ними сначала нужно установить программу для интеграции браузера со средой.

sudo apt-get install chrome-gnome-shell

Также нужно установить расширение для браузера **GNOME Shell Integration**



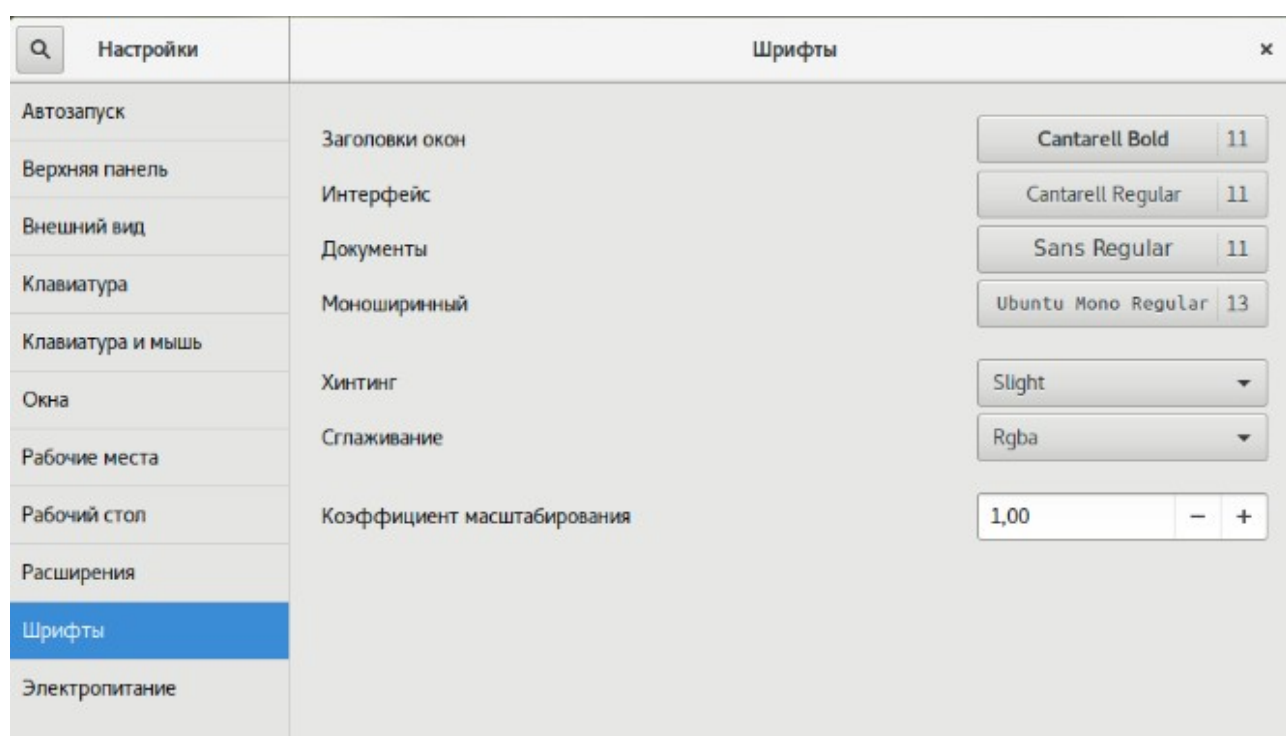
После этого вы можете открыть сайт и установить любое расширение, просто переключив нужный включатель:



Все установленные расширения появятся в списке **Gnome Tweak Tool**.

ШРИФТЫ GNOME

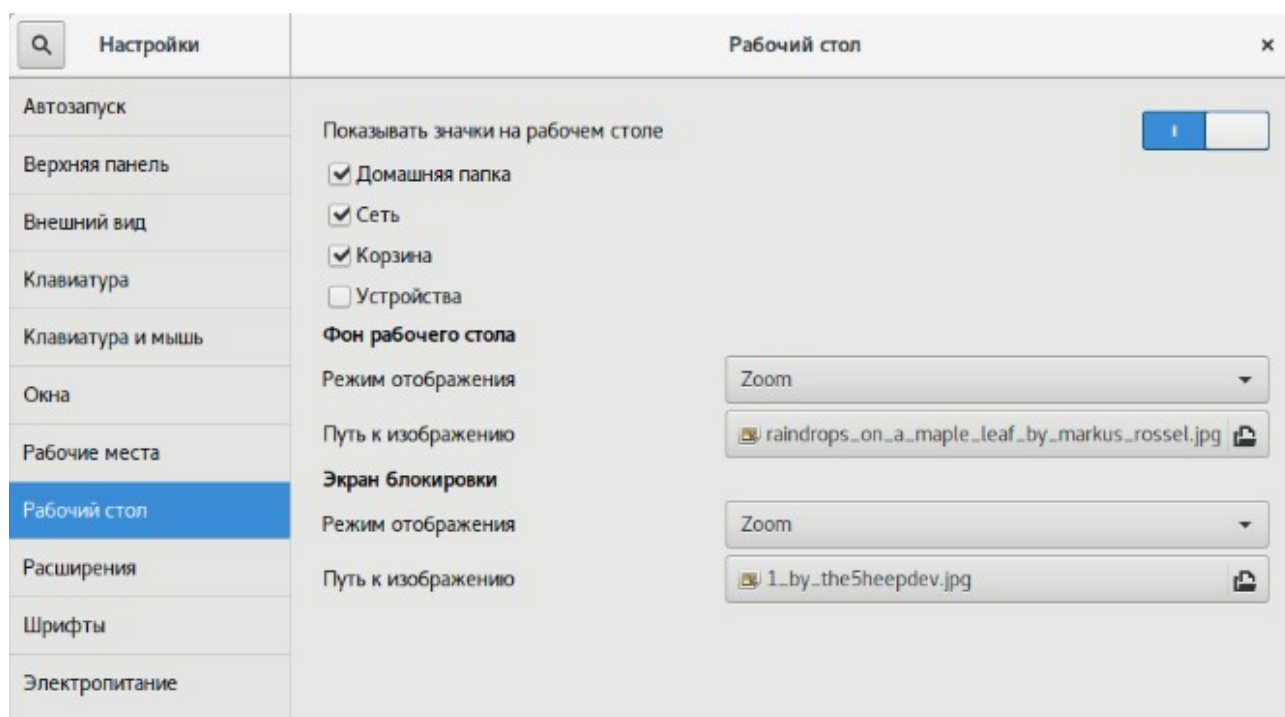
В отличие от других окружений, даже по умолчанию шрифты в Gnome выглядят превосходно. Но вы можете захотеть поменять шрифты на другие или изменить их размер. Все это делается с помощью утилиты Gnome Tweak Tool. Откройте утилиту, затем перейдите на вкладку "Шрифты"



Здесь вы можете настроить шрифты для основных компонентов окружения, настроить сглаживание, а также изменить коэффициент масштабирования шрифта для всей системы, это очень полезная настройка.

ЗНАЧКИ РАБОЧЕГО СТОЛА

Многие пользователи привыкли, что рабочий стол должен содержать значки компьютера и так далее. Их можно очень просто добавить. Для этого в той же утилите перейдите на вкладку **"Рабочий стол"**, включите переключатель **"Показывать значки на рабочем столе"** и отметьте нужное галочками.



Настройка интерфейса **Gnome** завершена, как видите, здесь есть не так много параметров, конечно, вы могли бы пойти и дальше, редактировать конфигурационные файлы, менять композитные менеджеры и менеджеры окон, но зачем? И так все отлично работает.

Все материалы книги взяты с свободных источников сети!

Приятного чтения друзья!

© Кучеренко Вадим Сергеевич