

Собственнические графические драйверы

Графическая система X.Org X11 обеспечивает большое количество драйверов и по крайней мере 2D видеоускорение для большинства видеокарт, однако, если система оснащена графическим процессором (видеокартой) от AMD (ATI) или NVIDIA, собственнические драйверы можно загрузить с веб-сайтов обоих этих производителей и установить.

Целью данной статьи является описание различных методов установки и настройки этих собственнических драйверов в Slackware.

Установка и настройка драйвера AMD (ATI) Catalyst

В AMD отказались от поддержки старых карт (4xxx и ниже) в текущем (12.6) и в будущих выпусках собственнических драйверов, поэтому убедитесь, что ваша карта все еще поддерживается. Если это не так, то `xf86-video-ati` с открытым исходным кодом должно быть достаточно, хотя оно несколько медленнее собственнических версий.

Получение драйвера

Обратите внимание, что программа установки будет компилировать драйвера (`Fglrx`) нуждается в пакетах `kernel-modules` и `kernel-source` для установки. Следующие шаги действительны для графической и [CLI](#) версии установщика.

Установка

Следующие инструкции предназначены для создания установочного пакета для Slackware. Он создается программой установки автоматически и, кажется, работает достаточно хорошо в большинстве случаев. Существует также «автоматизированный» метод установки.

Катализатор и набор драйверов для Linux можно загрузить по этой [ссылке](#). Файл подходит для 32 и 64-разрядных систем.

После скачивания, распакуйте файл:

```
$ unzip amd-driver-installer-VERSION-x86.x86_64.zip
```

Далее получите привилегии root:

```
$ su -
```

Переместитесь в каталог с файлом 'run' и запустите установщик:

```
# sh amd-driver-installer-VERSION-x86.x86_64.run
```

Программа установки будет собирать данные о вашей системе и подскажет как выбрать

автоматическую установку или создать пакет для конкретного дистрибутива. Выберите создание пакета для конкретного дистрибутива, так как позднее он может быть установлен при помощи инструмента для работы с пакетами Slackware.

Когда будет предложено выбрать дистрибутив, выберите «Detected OS: Slackware/Slackware».

После его завершения сборки, установщик создаст инсталляционный пакет в том же каталоге, где находится файл `run` и теперь вы можете сделать:

```
# installpkg fglrx-VERSION-x86-1.tgz
```

Если были обнаружены ошибки во время установки вы можете увидеть их в `/usr/share/ati/fglrx-install.log`.

Настройка

Вам нужно будет перезагрузить систему для того, чтобы воспользоваться новыми драйверами. Прежде чем сделать это вам нужно отредактировать `/etc/modprobe.d/blacklist.conf` (или создать новый: `/etc/modprobe.d/radeon_blacklist.conf`) и добавить следующие строки в файл:

```
blacklist radeon  
blacklist radeonhd
```

Затем вы можете создать новый конфигурационный файл X.Org, запустив:

```
# aticonfig --initial -f
```

X.Org обычно не требует изменения файла `/etc/X11/xorg.conf`, но в некоторых случаях (как правило, старые карты) необходимо добавить следующую строку в разделе «Device»:

```
Option "SWCursor" "true"
```

Если бы не было никаких ошибок и драйвер был установлен успешно, вы должны перезагрузить систему для использования драйверов.

Проверка

После перезагрузки, вы можете проверить работу драйвера, запустив:

```
$ fgl_glxgears
```

Должны отобразиться шестерни, вращающиеся плавно и без сбоев.

Можно выполнить:

```
$ glxinfo
```

Посмотрите в первых строках, включен ли DRI. Если нет, потребуются дополнительные действия, поскольку встроенный в установщик построитель пакетов довольно стар (старше официальной Slackware64), потребуется переместить некоторые файлы для работы на Slackware версии 14.1. Пакет для slackbuilds.org с устранением проблемы пока в разработке, а пока можно попробовать исправить вручную как описано [здесь \(сообщение на форуме LQ\)](#), а если не вышло, должна помочь автоматическая установка.

Автоматическая деинсталляция

Если вы выбрали автоматическую установку пакета и вам нужно удалить драйверы, вы можете запустить программу установки с дополнительным аргументом `uninstall`, вот так:

```
# sh amd-driver-installer-VERSION-x86.x86_64.run --uninstall
```

Или вызвать сценария удаления непосредственно:

```
#!/usr/share/ati/amd-uninstall.sh
```

Независимо от того, как он был установлен, удаление драйвера `fglrx` «ломает» `mesa`, так как некоторые файлы перемещаются. В этом случае необходимо переустановить `mesa`. Кроме того, если вы хотите вернуться к собственным `radeon/radeonhd` драйверам, вы должны удалить строки из *черного списка*, добавленные ранее.

Установка и настройка драйвера nVIDIA

Установка из SlackBuilds.org

Эдвард Кениг поддерживает пакеты `libvdpau`, `nvidia-driver` и `nvidia-kernel` в [SlackBuilds.org](#)

Все три пакета должны иметь функционирующий драйвера NVIDIA. Пакет `The nvidia-kernel` собирает модуль ядра, а пакет `nvidia-driver` собирает X.Org драйвер, содержит OpenGL реализацию, а также COMPAT32-библиотеки для использования их на [мультибиблиотечной Slackware64](#) системе. Пакет `libvdpau` требуется пакету `nvidia-driver`.

Деактивация модуля "nouveau"

Первым шагом при установке драйвера является добавление драйвера `nouveau` в черный список. Невыполнение этого требования может привести к ошибке `startx «ERROR: could not insert 'nvidia': No such device» «Ошибка: не удалось загрузить 'nvidia': Нет такого устройства»`.

Добавление `nouveau` в черный список происходит при помощи установки пакета `xf86-video-nouveau-blacklist` из директории «extra» версии Slackware. Это может быть достигнуто с помощью установочного CD или DVD, получением файла с зеркала Slackware, или при использовании утилиты `slackpkg`.

Установка пакета

Существуют два способа установки пакетов из SlackBuilds.org:

- путем загрузки соответствующих сборочных сценариев из [SBo](#), и сборки в соответствии с процедурой установки SlackBuild
- использование утилиты [sbopkg](#) которая автоматизирует загрузку исходных файлов и последовательную сборку/установку нескольких пакетов.

Как только пакеты были установлены с помощью инструкции, сопровождающие каждый SlackBuild, сервер X11 может быть запущен с полной поддержкой графического процессора. Для того чтобы настроить X11 на автоматический запуск, смотрите «Запуск X11 с поддержкой графического процессора NVIDIA» в следующем разделе.

Устранение неполадок



Установка двоичного драйвера nVIDIA

Установка двоичного драйвера nVIDIA включает следующие шаги:

- Загрузка соответствующего установщика драйвера `nvidia`
- Деактивация и выгрузка модуля драйвера `pouveau`
- Установка драйвера `nvidia`
- По желанию, конфигурирование системы для автоматического запуска графического интерфейса

Следующий совет перечисляет основные шаги по установке драйвера `nvidia`. Полный текст статьи следующий.

Для тех, кто просто хочет соответствующие команды, здесь приводится краткая информация из следующих действий:

- Зайдите на [веб-сайт драйверов nVIDIA для Unix](#) и загрузите соответствующий драйвер. Название пакета драйвера что-то вроде «NVIDIA-Linux-**ARCH-VERSION**.run» где ARCH будет архитектурой процессора/вашей системы и VERSION - версией драйвера
- Убедитесь что X11 не запущен; если это так, завершите его работу и перейдите в текстовый режим
- Авторизуйтесь в системе как root, запустите установщик драйвера nVIDIA командой

```
# sh /path/to/NVIDIA-Linux-ARCH-VERSION.run
```

- Если требуется деактивировать `pouveau`, сделайте это и перезагрузите систему
- Снова запустите установщик с привилегиями root
- Как минимум, выберите принятие лицензии и установите драйвер. Пожалуйста, ознакомьтесь с разделом параметров установщика nVIDIA чтобы узнать о вариантах автоматической установки.

Если `nvidia-xconfig` не был запущен установщиком, то после завершения установки и перед запуском сервера X11 файл `«/etc/X11/xorg.conf»` должен быть отредактирован вручную.

Загрузка соответствующего установщика драйвера nVIDIA

Первый шаг заключается в том, чтобы загрузить соответствующий драйвер с [вебсайта драйверов nVIDIA для Unix](#). Для 64-разрядной версии Slackware (включая [мультибиблиотечные системы](#)) вам необходимо загрузить пакет драйвера `«Linux x86_64/AMD64/EM64T»`, в то время как 32-разрядной версии Slackware необходим драйвер `«Linux x86/IA32»`. Самый безопасный выбор для подбора правильной версии заключается в использовании *«Актуальной версии с долговременной поддержкой»* но в некоторых случаях (очень новая видеокарта или сопутствующие проблемы, которые вы пытаетесь решить), вы можете выбрать вместо этого *«Актуальную версию с кратковременной поддержкой»*.

Устаревшие драйверы также доступны для старых видеокарт. При выборе драйвера, отобразится список совместимых графических процессоров. Как только вы определили нужный драйвер, необходимо принять условия лицензии, и сохранить файл в доступном месте. Перед запуском программы установки есть некоторые другие действия, которые должны быть приняты.

Деактивация модуля nouveau

В целом, графический процессор nVIDIA сразу же будет обнаружен и открытый графический драйвер `«nouveau»` активируется в системе. Это позволит использовать 3D-ускорение видео на многих картах. Модули `nouveau` и `nvidia` несовместимы; Таким образом, прежде чем приступить к установке модуля `nvidia`, необходимо выгрузить `nouveau` и запретить его автоматическую загрузку.

Для этого должен быть создан файл в каталоге `«/etc/modprobe.d/»` и содержать текст

```
blacklist nouveau
options nouveau modeset=0
```

и соответствующе назван, например, `«disable_nouveau.conf»`

Первая строка будет блокировать модуль `nouveau` от автоматической загрузки во время запуска. Модуль по-прежнему может быть загружен вручную пользователем или сервером X. Таким образом, вторая строка предотвратит `modeset` ядра, что позволяет выгружать модуль `nouveau`.¹⁾

As mentioned in the `modprobe.d` README file, the monolithic module blacklist file has been split into smaller files and stored under `/lib/modprobe.d/` However, a `.conf` file in the `/etc/modprobe.d/` directory will override one in `/lib/modprobe.d/` This is thus a more sure way to block the loading of the `nouveau` driver.

After displaying an error to the effect that the `nouveau` and nVIDIA proprietary drivers are incompatible, newer nVIDIA installers will prompt for permission to create a `.conf` file to blacklist `nouveau` in `/etc/modprobe.d/` which will be named `nvidia-installer-disable-nouveau.conf` By default, the contents of this file will be:

```
# generated by nvidia-installer  
blacklist nouveau  
options nouveau modeset=0
```

The installer will then state that it has failed and request that the system be restarted to put changes into effect. Once the system has been restarted, the nouveau driver will no longer be loaded, and installation of the binary driver can continue.

Alternately, the package `xf86-video-nouveau-blacklist` from the «/extra» directory of the Slackware installation media should be installed. Once installed, the system should be restarted, at which point driver installation can continue.

nVIDIA Binary Driver Installation

nVIDIA binary driver installation **can not be completed** while the X11 Window System is active.

The default behavior of Slackware is to boot into a text-only terminal. In this case, the nVIDIA installer can simply be run. However, if the graphical environment was already launched, it must be exited. If lunched with the «startx» command, closing the graphical environment can be accomplished by logging out or, and only if necessary, pressing `Ctrl+Alt+Bksp`.

If using a graphical login manager, such as KDM, simply exiting via the menu or pressing `Ctrl+Alt+Bksp` **will not work** because the graphical login manager will be restarted automatically. The best way to shut down X gracefully is by logging off, switching to a console terminal by pressing `Alt+F2`, logging into the console as root and running:

```
telinit 3
```

Another method to return to the console, crude and thus not recommended but effective if necessary, is to stop/kill the DM and then exit the X Server by pressing `Ctrl+Alt+Bksp`. For example, KDM can be killed by issuing the command

```
killall -9 kdm
```

Again, this should only be used if other methods fail.

Once X11 is no longer running, the nVIDIA installer can be launched as root. Either log in as root or issue the «su -» command, then run the installer. Sometimes, it is advantageous to run the installer using some of the available command-line options. For more information, consult «nVIDIA Installer Options» at the end of this section.

```
$ su -  
Password:  
# sh /path/to/NVIDIA-Linux-ARCH-VERSION.run
```

When the installer is launched without any command line options and excepting any errors, the installation will proceed as follows:

- The installer will extract itself and start an ncurses interface. The first step is to accept the license terms.
- When installing the module for a system that has no previous version, the installer will begin to build the module. If, however, a previous nVIDIA driver was found, the installer will prompt for permission to remove the previous driver as part of the new installation. To proceed with the installation, permission should be granted to remove the previous installation.
- After building the module on x86_64 systems, the installer will prompt about installing nVIDIA's 32-bit compatibility OpenGL libraries. You will only need the 32-bit compatibility libraries if your Slackware is 64-bit [multilib](#).
- Next, in the case of a previous version, the installer will uninstall the module.
- Once no conflicting X and (should this option be chosen) OpenGL files are found, the module will be installed.
- The program will then offer to use the nvidia-xconfig utility to modify the xorg.conf file to reflect the change in video drivers. This utility often works, but could possibly change other values in xorg.conf. It does, however, back-up the original file to `«/etc/X11/xorg.conf.nvidia-xconfig-original»`
- A prompt confirms that the driver installation is complete and was a success, at which point the program exits to the console.

The nvidia module should now be installed for the currently running kernel.

You must create an X.Org configuration file which loads the binary Nvidia driver if you decided *not* to let the nvidia-xconfig utility modify your computer's xorg.conf file. The X.Org of Slackware supports individual `«*.conf»` files in a directory `/etc/X11/xorg.conf.d`. Any file with a `.conf` extension will be included together with the main `/etc/X11/xorg.conf` file.

You could create for instance a file named `/etc/X11/xorg.conf.d/10-nvidia.conf` with the following content:

```
Section "Device"
    Identifier "Device0"
    Driver "nvidia"
    VendorName "Nvidia Corporation"
    BoardName ""
EndSection
```

Without this definition, you will not get accelerated Nvidia GPU support! X.Org would fall-back to VESA mode because the nouveau driver has been blacklisted. *The kernel will not auto-detect the binary driver as opposed to the nouveau driver.*

Starting X11 with nvidia GPU Support

All that remains is to start the X.org server. This can be accomplished by configuring Slackware to start in *runlevel 4*, which will start a graphical login manager, such as KDM or XDM, on boot. If you want this, then you need to edit the file `«/etc/inittab»` and change the line

```
id:3:initdefault:
```

to

```
id:4:initdefault:
```

Otherwise you can login to a user account and issue the «startx» command to start an X session.

By default, startx will start the window manager which was chosen during installation. To change this behavior, the file «.xinitrc» in the user's home directory can be edited to start a different WM. Alternatively, the default WM can be altered on a per-user basis by using the command «xwmconfig» and selecting one of the available WM's.

nVIDIA Installer Options

The nVIDIA Installer has many options available to the system maintainer which can be accessed by running the installer with the -A option:

```
# sh ./NVIDIA-Linux-ARCH-VERSION.run -A
```

Some common options are

- -a, --accept-license : bypasses the nVIDIA license screen. By doing so, the license is accepted.
- --update : check for an updated driver on the nVIDIA website; if it exists, automatically download and install the new version.
- --uninstall : removes the nVIDIA driver and other files installed previously.
- -q, --no-questions : assumes default answers to all questions. Note: does not automatically accept the license.
- -s, --silent : runs the binary installer without a curses UI, automatically accepts the license and uses the default answers for all questions.

Troubleshooting



Sources

- nVidia section originally written by [rinias](#)

Translation in progress. (*drhouse123*)

[howtos](#), [software](#), [nvidia](#), [author rinias](#)

¹⁾

ftp://download.nvidia.com/XFree86/Linux-x86_64/256.44/README/commonproblems.html

From:

<https://docs.slackware.com/> - **SlackDocs**

Permanent link:

https://docs.slackware.com/ru:howtos:hardware:proprietary_graphics_drivers

Last update: **2016/06/16 08:43 (UTC)**

