



Thorsten Leemhuis

Bilderwege

Die Grafiktreiber-Architektur von Linux

Das Einrichten oder Aktualisieren von Grafiktreibern bereitet Linux-Anwendern immer wieder Probleme. Mit Schuld daran: das Verfahren, mit dem Linux-Distributionen die Grafikhardware ansprechen, denn es verteilt den Treibercode auf zahlreiche Stellen.

Ein Linux-„Grafiktreiber“ ist in Wirklichkeit kein einzelner Treiber, sondern ein Verbund aus bis zu zehn Treibern. Sie erledigen ganz unterschiedliche Dinge und werden teilweise unabhängig voneinander entwickelt. Trotzdem müssen sie zusammenpassen, damit alles rund läuft. Gerade bei der Treibereinrichtung oder Systemaktualisierung gerät das Gefüge schnell durcheinander, sodass die grafische Bedienoberfläche nicht mehr startet; in anderen Fällen sind 3D- oder Video-Beschleunigung gestört. Solche Probleme sind oft schnell diagnostiziert und behoben, wenn man weiß, wie Linux-Distributionen die Grafikhardware ansprechen.

Typischerweise besteht ein Verbund aus mindestens vier Treibern. Die Hauptverantwortung hat ein Kernel-Treiber, denn er koordiniert den Zugriff auf die GPU (Graphics Processing Unit) und offeriert anderen Treibern einige Basis-Funktionen. Diese nutzt der X-Treiber für den X-Server von X.Org, über den Desktop-Oberflächen sowie die darüber gestarteten Anwendungen ihre Bedienelemente ausgeben. 3D-Beschleunigung erledigt ein via OpenGL angesprochener Treiber, der sich bei der Bildausgabe mit dem X-Server koordiniert. Ähnlich funktioniert auch die Hardware-beschleunigte Wiedergabe moderner Video-Formate. Der dafür

zuständige Video-Treiber arbeitet wie der OpenGL-Treiber auch unabhängig vom X-Server; ein wichtiges Detail, durch das diese Treiber auch unter dem designierten X11-Erben Wayland funktionieren (siehe S. 170).

Kernel-, X-, 3D- und Video-Treiber gibt es bei allen fünf Treiberfamilien, die man häufiger bei PCs antrifft. Eine davon ist „Intel“, der das Gros der GPUs anspricht, die Intel in den letzten Jahren in Mainboard-Chipsätze und Prozessoren verbaut hat. Er unterliegt einer Open-Source-Lizenz, daher liefern Linux-Distributionen ihn gleich mit und aktivieren ihn vollautomatisch. Das gilt auch für die Treiberfamilie „Radeon“, die AMDs gleichnamige Grafikchips unterstützt, und „Nouveau“ für GeForce-Chips von Nvidia. Außerdem bieten AMD und Nvidia für ihre Grafikprozessoren proprietäre Linux-Treiberfamilien an, die als „Fglrx“ und „Nvidia“ bezeichnet werden; AMD nennt Fglrx auch „Catalyst“.

Im Sprachgebrauch meinen die Bezeichnungen Intel, Radeon, Nouveau, Nvidia und Fglrx meist die ganze Treiberfamilie, obwohl es jeweils nur die Bezeichnung des X-Treibers ist; die anderen Treiber tragen vielfach ganz andere Namen. Einige der fünf Familien bestehen aus mehr als vier verschiedenen Treibern. Auch bei Arbeitsverteilung und Ansatz gibt es feine, aber entscheidende Unterschiede; bei den drei quelloffenen Treiber-

```

cttest@localhost:~$
# Grafik-Hardware =
01:00.0 VGA compatible controller [0300]: Advanced Micro Devices, Inc. [AMD/ATI] Caicos [Radeon HD 6450/7450/8450 / R5 230 OEM] [1002:6779]
01:00.1 Audio device [0403]: Advanced Micro Devices, Inc. [AMD/ATI] Caicos HDMI Audio [Radeon HD 6400 Series] [1002:aa98]

# Kernel-Treiber & Firmware im Initramfs =
-rw-r--r-- 3 root root 0 Sep 13 00:53 usr/lib/firmware/radeon/CAICOS_mc.bin
-rw-r--r-- 3 root root 0 Sep 13 00:53 usr/lib/firmware/radeon/CAICOS_me.bin
-rw-r--r-- 1 root root 4480 Sep 13 00:53 usr/lib/firmware/radeon/CAICOS_pfp.bin
-rw-r--r-- 1 root root 24512 Sep 13 00:53 usr/lib/firmware/radeon/CAICOS_smc.bin
-rw-r--r-- 1 root root 467396 Sep 26 16:57 usr/lib/modules/3.16.3-302.fc21.x86_64/kernel/drivers/gpu/drm/radeon/radeon.ko.xz

# X-Treiber =
-rwxr-xr-x 1 root root 430K 3. Jul 16:10 /usr/lib64/xorg/modules/drivers/radeon_drv.so

# OpenGL-Treiber (32- und 64-Bit) =
-rwxr-xr-x 7 root root 7,9M 27. Sep 13:57 /usr/lib64/dri/r600_dri.so
-rwxr-xr-x 7 root root 8,1M 27. Sep 13:59 /usr/lib64/dri/r600_dri.so

# VDPAU-Video-Beschleuniger-Treiber (32- und 64-Bit) =
-rwxr-xr-x 3 root root 3,8M 27. Sep 13:57 /usr/lib64/vdpau/libvdpau_r600.so.1.0.0
-rwxr-xr-x 3 root root 3,8M 27. Sep 13:59 /usr/lib64/vdpau/libvdpau_r600.so.1.0.0

# Von OpenGL- und Video-Treiber verwendete Bibliotheken mit Treibercode (32- und 64-Bit) =
-rwxr-xr-x 1 root root 56K 17. Aug 11:26 /usr/lib64/libdrm_radeon.so.1.0.1
-rwxr-xr-x 1 root root 25M 14. Aug 19:40 /usr/lib64/llvm/libLLVM-3.4.so
-rwxr-xr-x 1 root root 56K 17. Aug 11:46 /usr/lib64/libdrm_radeon.so.1.0.1
-rwxr-xr-x 1 root root 26M 14. Aug 19:40 /usr/lib64/llvm/libLLVM-3.4.so

# HDMI-Audio-Treiber =
-rw-r--r-- 1 root root 21K 26. Sep 16:55 /lib/modules/3.16.3-302.fc21.x86_64/kernel/sound/pci/hda/snd-hda-codec-hdmi.ko.xz
cttest@localhost ~$

```

Der Treibercode zur Unterstützung einer AMD Radeon HD 6450 verteilt sich auf mehr als ein Dutzend Dateien.

familien erledigt etwa der Kernel-Treiber einige Aufgaben, die bei den beiden proprietären Treiberfamilien andere Komponenten wahrnehmen. Ihren Ansatz erläutern wir im hinteren Artikeldrittel. Android bleibt hier außen vor, denn es arbeitet grundverschieden; auch andere Embedded-Linuxe arbeiten häufig etwas anders als PC-Linuxe.

Open-Source-Ansatz

Alle drei Open-Source-Treiberfamilien gehen sehr ähnlich vor. Beim Start einer Linux-Distribution für PCs geht es zuerst ganz ohne Grafikchip-spezifische Treiber los, denn der Bootmanager verwendet zur Bildausgabe generische Mechanismen – also das Graphics Output Protocol (GOP) von UEFI, die VESA BIOS Extensions (VBE) oder den VGA Text Mode, den DOS schon Ende der Achtzigerjahre genutzt hat.

Über VGA oder GOP gibt auch der Kernel seine ersten Meldungen aus. Bei Systemen mit den quelloffenen Treibern aber nur die allerersten, denn typischerweise laden Linux-Distributionen den Kernel-Grafiktreiber bereits im Initramfs – also noch bevor das Root-Dateisystem eingebunden wird. Dieses Kernel-Modul heißt bei Intel i915. Es kümmert sich um Hardware-Initialisierung, Speicher-verwaltung und die Validierung der Kommandos, die etwa OpenGL- oder Video-Treiber an den Grafikchip senden. Der Kernel-Treiber stellt direkt beim Laden auch eine Bildschirmauflösung ein, die optimal zum Ausgabegerät passt; die dazu nötigen Informationen bezieht er aus einer EDID (Extended Display Identification Data) genannten Datenstruktur, die er per DDC (Display Data Channel) beim Bildschirm abfragt.

Das Einstellen der Bildschirmparameter durch den Kernel nennt sich Kernel-based Mode Setting (KMS). Kernel-Grafiktreiber werden daher manchmal auch KMS-Treiber genannt. Treffender ist allerdings die Bezeichnung „DRM-Treiber“, denn Treiber und KMS-Funktion gehören zum Direct Rendering Manager (DRM) des Linux-Kernels. Er

entstand Ende der Neunziger und damit ein Jahrzehnt vor KMS.

Oberflächliches

Beim Einstellen der Auflösung durch KMS wird der Bildschirm kurz dunkel und alle bis dorthin vom Bootmanager oder Kernel ausgegebenen Meldungen verschwinden. Viele Distributionen starten an diesem Punkt eine Boot-Animation, um die Systeminitialisierung zu verschönern. Oft kümmert sich darum das Programm „Plymouth“, das auf KMS zurückgreift, das auch Funktionen zur Bildausgabe bietet.

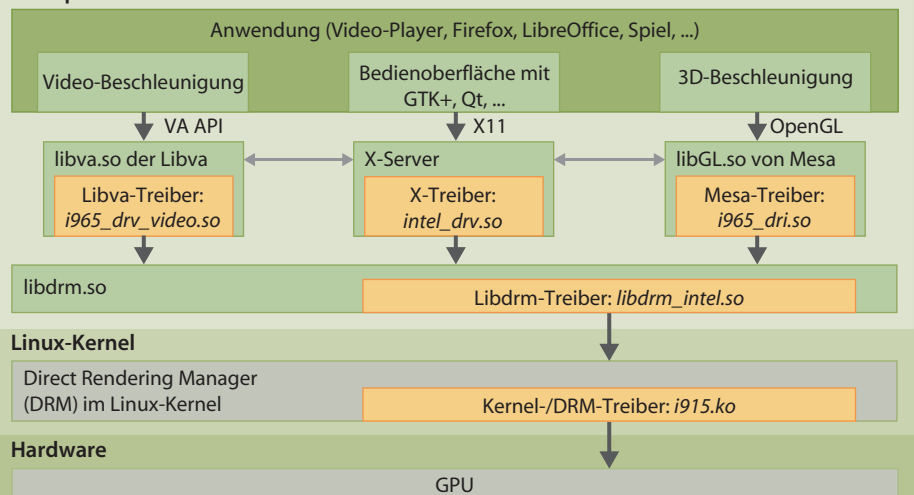
Für Desktop-Oberflächen und grafische Programme ist mehr nötig, daher startet nach der grundlegenden Systeminitialisierung der X-Server von X.Org. Er implementiert das X Window System (X11) und offeriert Funktionen und Schnittstellen, mit denen Anwendungen eine Bedienoberfläche erstellen und an den X-Server übergeben können. Der kümmert sich um die Ausgabe über die Grafikhardware und liefert die Eingaben von Maus, Tastatur und Co. an die Anwendungen.

Linux-Distributionen verwenden bei PCs seit jeher einen X-Server als Unterbau für grafische Bedienoberflächen. Die früher er-

Die quelloffenen Treiber für Intel-Grafik

Eine Linux-Anwendung erzeugt ihre Bedienoberfläche mit Hilfe des X-Servers. Zur Video- und OpenGL-Beschleunigung greift sie auf Bibliotheken zurück, die sich mit dem X-Server koordinieren. Die Hauptarbeit überlassen die Bibliotheken allerdings Treibern, die Daten und Grafikkommandos via Libdrm und Linux-Kernel an die Grafikhardware senden, die das Video dekodiert oder die 3D-Befehle umsetzt.

Userspace



```

cttest@localhost:~$ LIBGL_DEBUG=verbose glxinfo 2>&1 | grep -e '^libGL' -e '^direct' -e '^OpenGL'
libGL: screen 0 does not appear to be DRI3 capable
libGL: pci id for fd 4: 1002:6779, driver r600
libGL: OpenDriver: trying /usr/lib64/dri/tls/r600_dri.so
libGL: OpenDriver: trying /usr/lib64/dri/r600_dri.so
libGL: Can't open configuration file /home/cttest/.dirc: No such file or directory.
libGL: Can't open configuration file /home/cttest/.dirc: No such file or directory.
direct rendering: Yes
OpenGL vendor string: X.Org
OpenGL renderer string: Gallium 0.4 on AMD CAICOS
OpenGL core profile version string: 3.3 (Core Profile) Mesa 10.3.0
OpenGL core profile shading language version string: 3.30
OpenGL core profile context flags: (none)
OpenGL core profile profile mask: core profile
OpenGL core profile extensions:
OpenGL version string: 3.0 Mesa 10.3.0
OpenGL shading language version string: 1.30
OpenGL context flags: (none)
OpenGL extensions:
OpenGL ES profile version string: OpenGL ES 3.0 Mesa 10.3.0
OpenGL ES profile shading language version string: OpenGL ES GLSL ES 3.0
OpenGL ES profile extensions:
cttest@localhost:~$ █

```

Das Programm Glxinfo liefert Informationen zum verwendeten 3D-Treiber, der bei Open-Source-Grafiktibern meist ein Mesa-Bestandteil ist.

forderliche und oft mühsame X-Konfiguration ist heute zumeist unnötig, denn moderne X-Server laden beim Start automatisch alle Treiber, die sie für Eingabegeräte und Grafikkhardware benötigen. Beim Intel ist das etwa der X-Treiber `intel_drv.so`, der bei Ubuntu im Verzeichnis `xorg/modules/drivers/` unterhalb von `/usr/lib/` liegt; bei einem x86-64-Fedora liegt das Verzeichnis in `/usr/lib64/`.

Die bei PCs gängigen X-Treiber haben früher auch alles enthalten, um die Bildschirm-auflösung einzustellen. Dieses User-Mode Setting beherrschen sie dieser Tage nicht mehr. Stattdessen müssen sie den DRM-Treiber über KMS-Funktionen anweisen, den Bildschirm zu konfigurieren, wie es der Anwender vorgibt; etwa über die Anzeigeeinstellungen der Desktops oder das Kommandozeilenwerkzeug `xrandr`, die ihre Wünsche über die X Resize, Rotate and Reflect Extension (RandR) an X-Server und Treiber übermitteln.

Der X-Treiber ist auch für die Video-Beschleunigung über die X Video Extension (kurz: XVideo oder XV) und die X-Video Motion Compensation (XvMC) zuständig; diese Funktionen des X-Servers werden aber kaum noch genutzt, denn sie eignen sich nicht für moderne HD-Video-Formate.

Die verbleibende Hauptfunktion des X-Treibers ist die 2D-Beschleunigung, durch die der Treiber etwa das Bewegen von Fensterinhalten an den Grafikchip delegiert. Dazu diente in den letzten Jahren zumeist EXA oder das Intel-spezifische UXA. Jünger ist das vom Radeon-X-Treiber bereits seit einer Weile verwendete Glamor, bei dem der X-Server zur 2D-Beschleunigung auf den 3D-Treiber zurückgreift. Das beraubt X-Treiber ihrer letzten wesentlichen Aufgabe. Es erleichtert dadurch das Design von Hardware-spezifischen Treibern und kann diese sogar überflüssig machen; das lockt auch andere Entwickler, daher wurde Glamor in den X-Server 1.16 integriert, der im Juli erschienen ist.

Beschleunigtes 3D

Zur 3D-Beschleunigung nutzen Linux-Programme typischerweise OpenGL-Befehle, wozu sie auf die Bibliothek `libGL.so` zurückgreifen. Dahinter verbirgt sich standardmäßig die Mesa 3D Graphics Library. Die Bibliothek ist als Software-Implementation von OpenGL gestartet, um Zeichenbefehle mit der CPU auszuführen, als 3D-taugliche GPUs noch rar und teuer waren. Das kann Mesa auch heute noch; typischerweise leitet es die OpenGL-Befehle aber an die GPU weiter, die sie um ein Vielfaches flotter ausführt. Hierfür bringt Mesa 3D-Treiber mit. Jener für moderne Intel-GPUs heißt `i965_dri.so` (kurz: `i965`) und liegt bei einem x86-64-Ubuntu in `/usr/lib/x86_64-linux-gnu/dri/`; bei einem x86-64-Fedora ist es `/usr/lib64/dri/`.

„DRI“ steht dabei für Direct Rendering Infrastructure. Das ist der Oberbegriff für die Infrastruktur zur Nutzung der 2D- und 3D-Beschleunigung von Grafikchips. Neben den Mesa-Treibern gehört zur DRI auch der bereits erwähnte Direct Rendering Manager (DRM) im Kernel sowie einige Funktionen im X-Server. Die wichtigste ist GLX (OpenGL Extension to the X Window System), das OpenGL und X-Server verbindet; es stellt unter anderem das X-Fenster, in dem das per OpenGL gerenderte Bild landet, damit sich 2D- und 3D-Anwendungen nicht auf der Oberfläche ins Gehege kommen.

Mithilfe von GLX können X-Anwendungen ihre OpenGL-Zeichenbefehle an den X-Server übergeben, damit der sie an den 3D-Treiber übergibt und das so berechnete Bild dann ausgibt. Dieses „Indirect Rendering“ ist aber mit viel Overhead verbunden, der Performance kostet. Daher entstand „Direct Rendering“, bei dem Anwendungen die OpenGL-Befehle direkt an den 3D-Treiber übergeben, der sie am X-Server vorbei direkt in dem Bildschirmbereich ausgibt, der zuvor mit dem X-Server ausgehandelt wurde.

Moderne Linux-Desktops stellen die ganze Bedienoberfläche mithilfe von Direct Rendering dar. Das erledigt ein „Compositing Window Manager“. Er ist letztlich eine 3D-Anwendung, die den ganzen Bildschirmbereich füllt, den der X-Server bereitstellt. Er erzeugt typischerweise ein Hintergrundbild und andere Desktop-Elemente; ferner zeichnet er Fensterrahmen und kombiniert diese mit den Fensterinhalten, die Anwendungen erzeugen. Zum Erhalt dieser Inhalte tauscht er sich nicht nur mit dem X-Server aus, sondern bekommt von der GPU bei Bedarf auch Referenzen auf im Grafikspeicher liegende Bilder, die Anwendungen über den X-Server oder mithilfe von 3D- und Video-Beschleunigung erzeugt haben. Diese Referenzen nutzt der Compositing Window Manager dann in OpenGL-Befehlen, mit denen er die GPU zum Bau des Gesamtbildes anweist. Durch diesen Ansatz können die Fenster-Vorschauen beim Taskwechsel mit Alt+Tab auch Bilder enthalten, die mit 3D- und Video-Beschleunigung erzeugt wurden.

Beschleunigtes Video

Zur Nutzung der Video-Beschleunigung haben derzeit drei Programmierschnittstellen größere Bedeutung. Das wichtigste API ist das von Nvidia erfundene VDPAAU (Video Decode and Presentation API for Unix). Programme können es über die von Nvidia unter Open-Source-Lizenz entwickelte `Libvdpau` nutzen. Ähnlich wie bei Mesa erledigt die Bibliothek die Beschleunigung nicht selbst, sondern greift dazu auf Treiber zurück, die sich bei der Bibliothek einklinken.

Ähnlich funktioniert es auch bei dem von Intel erfundenen Video Acceleration API (VA API), bei dem die Bibliothek `Libva` heißt. Auch die Linux-Implementation `Libomxil-Bellagio` des Plattform-übergreifenden OpenMAX (Open Media Acceleration) geht so vor. Das manchmal als OMX abgekürzte Video-API stammt von der Khronos Group, die auch OpenGL vorantreibt.

Über VDPAAU lassen sich nur Decoder ansprechen, daher beschleunigt es lediglich die Video-Wiedergabe. VA API und OpenMAX hingegen stellen auch Schnittstellen bereit, um Video-Encoder für MPEG2 oder H.264/AVC zu nutzen, die in vielen modernen GPUs stecken.

AMDs Radeon-Treiberfamilie soll bald alle drei APIs unterstützen, umfasst derzeit aber nur Treiber für OpenMAX und VDPAAU. Zu Nouveau gehört ein Treiber für VDPAAU; zu Intels-Treiberfamilie einer, der das VA API implementiert.

Über Plug-ins kann das von verschiedenen Linux-Anwendungen genutzte Gstreamer-Framework alle drei APIs nutzen. Auch MythTV, VLC und XBMC unterstützen alle drei. Der Mplayer beherrscht nur VA API und VDPAAU. Das Flash-Plug-in unterstützt nur VDPAAU und kann daher die Video-Beschleunigung von Intel-Chips nicht direkt nutzen. Dieses Manko beseitigt `Libvdpau-VA-Gl` – ein VDPAAU-Treiber, der zur Video-


```

cttest@localhost:~/Downloads
[cttest@localhost Downloads]$ vdpauinfo | head -n 25
display: :0 screen: 0
API version: 1
Information string: G3DVL VDPAU Driver Shared Library version 1.0

Video surface:

name      width height types
-----
420       16384 16384  NV12 YV12
422       16384 16384  UYVY YUYV
444       16384 16384  Y8U8V8A8 V8U8Y8A8

Decoder capabilities:

name      level macbs width height
-----
MPEG1      0  9216  2048  1152
MPEG2_SIMPLE 3  9216  2048  1152
MPEG2_MAIN 3  9216  2048  1152
H264_BASELINE 41 9216  2048  1152
H264_MAIN 41  9216  2048  1152
H264_HIGH 41  9216  2048  1152
VC1_ADVANCED 4  9216  2048  1152
MPEG4_PART2_SP 3  9216  2048  1152
MPEG4_PART2_ASP 5  9216  2048  1152

[cttest@localhost Downloads]$ vlc big_buck_bunny_720p_h264.mov 2>&1 | grep 'avcodec'
[0x7fa488da0cb8] avcodec decoder: Using G3DVL VDPAU Driver Shared Library version 1.0
for hardware decoding.
[0x7fa488da0cb8] avcodec decoder: Using G3DVL VDPAU Driver Shared Library version 1.0
for hardware decoding.

```

Drei Video-Beschleunigungs-API haben in der Linux-Welt größere Verbreitung; am häufigsten trifft man auf VDPAU, das auch VLC unterstützt.

enthalten aber ein wenig GPU-spezifische Informationen, daher könnte man sie auch als Treiber bezeichnen. Bei Intel heißt die Datei `libdrm_intel.so` und liegt im Verzeichnis mit den Systembibliotheken; bei x86-64-Ubuntu also in `/usr/lib/x86_64-linux-gnu/` und bei x86-64-Fedora in `/usr/lib64/`.

Direkt in diesen Verzeichnissen oder unterhalb davon liegen auch die erwähnten Bibliotheken und Treiber, auf die Anwendungen zur 3D- und Video-Beschleunigung zurückgreifen. 32-Bit-x86-Programme können dort liegende Libraries nicht verwenden. Man muss daher 32-Bit-Fassungen von `Libdrm` und `Mesa` samt der dort einklinkenden Treiber einrichten, damit beispielsweise ein x86-32-Spiel die 3D-Beschleunigung einer x86-64-Distributionen nutzen kann. Diese x86-32-Bibliotheken bleiben bei der Standardinstallation häufig außen vor, lassen sich aber oft über die Paket-Repositories nachinstallieren. Längst nicht bei allen Distributionen gibt es x86-32-Pakete der Video-Bibliotheken und -Treiber, um auch die

Beschleunigung auf VA API und OpenGL zurückgreift.

Vermittler

Die für 2D-, 3D- und Video-Beschleunigung zuständigen Treiber arbeiten wie normale

Programme im Userspace und sind auf den DRM-Treiber des Kernels angewiesen. Auf ihn greifen die Userspace-Treiber mithilfe der `Libdrm` zu. Ähnlich wie bei `Mesa` und Video-Beschleunigungs-Bibliotheken stecken die Hardware-spezifischen Funktionen in eigenen Dateien. Die sind recht simpel,

Anzeige

Grafiktreiber-Familien für Linux

gängige Kurzbezeichnung	unterstützte Hardware ¹	treibende Entwicklerkraft	Kernel-Treiber	X-Treiber	3D-Beschleunigung ²	Video-Beschleunigung ²
quelloffene Treiber						
Intel	HD-Graphics- und Iris-Serien	Intel	i915	intel	Mesa (i965)	VA API: intel
Nouveau	GeForce- und Quadro-GPUs	Open-Source-Community	nouveau	nouveau	Mesa (nouveau)	VDPAU: Mesa (nouveau)
Radeon	Radeon HD 2400 bis 7670	AMD	radeon	radeon	Mesa (r600)	VDPAU & OMX: Mesa (r600)
	Radeon HD 7750 und neuer	AMD	radeon	radeon	Mesa (radeonsi)	VDPAU & OMX: Mesa (radeonsi)
proprietäre Treiber						
Fglrx aka Catalyst	Radeon HD 5000 und neuer ³	AMD	fglrx	fglrx, fglrx-libglx.so	fglrx_dri.so, libGL.so	XvBA: libAMDvBA.so
Nvidia	GeForce 8000, 9000, 100 bis 980 ^{4,5}	Nvidia	nvidia	nvidia, libglx.so	libnvidia-glcore.so, libGL.so	VDPAU: libvdpau_nvidia.so

¹ Faustregel; Details siehe Dokumentation

² Treibernamen teilweise gekürzt

³ AMD unterstützt die Serien 2000, 3000 und 4000 mit einem Legacy-Treiber, der inkompatibel zu einigen aktuellen Linux-Distributionen ist

⁴ Ferner verwandte Quadro-Chips

⁵ ältere GPUs unterstützt Nvidia mit Legacy-Treibern; auch die GeForce-GPUs der Serien 8000, 9000, 100, 200 und 300 werden bald nur noch durch einen solchen unterstützt

Video-Beschleunigungsfunktionen mit 32-Bit-Software nutzen zu können.

Aktualisieren

Die verschiedenen Treiber einer Familie interagieren stark miteinander. Man kann sie aber in der Regel unabhängig voneinander aktualisieren, denn die Aufgabengebiete sind klar voneinander abgegrenzt und die Entwickler achten auf Abwärtskompatibilität. Hin und wieder muss man aber mehrere Treiber aktualisieren, um neue Funktionen nutzen zu können. Zudem muss man in der Regel die ganze Treiberfamilie aktualisieren, um volle Unterstützung für eine neue GPU-Generation nachzurüsten.

Das ist enorm aufwendig, denn es gibt keinen Treiber-Installer, der eine komplette Treiberfamilie unter den gängigen Linux-Distributionen einrichtet. Intel offeriert immerhin einen Installer zur Einrichtung von Paket-Repositories, die aktuelle Versionen von Fedora und Ubuntu mit neuen Treibern versorgen. Die Repositories zeigen aber auch, warum ein universelles Installationsprogramm keineswegs trivial wäre, denn sie tauschen zentrale Komponenten von Linux-Distributionen aus. Das lässt sich nicht vermeiden, weil DRM-, Libdrm- und OpenGL-Treiber mehr oder weniger feste Bestandteile von Linux-Kernel, Libdrm und Mesa sind. Einen neuen OpenGL-Treiber gibt es daher nicht separat, sondern nur zusammen mit einer neuen Mesa-Version; eine solche selbst zu kompilieren erfordert mehr als nur fortgeschrittene Linux-Kenntnisse. Ähnlich ist es beim Kernel, der wie Mesa ein zentraler Baustein ist, der Stabilität und Funktionsumfang einer Linux-Distribution stark beeinflusst. Zwar gibt es mit dem Linux-Backport-Projekt einen Weg, um einen aktuellen DRM zusammen mit seinen Treibern unter älteren Kernen einzurichten; das Ganze ist aber kompliziert und schwer zu pflegen [1]. Typischerweise ist es erheblich einfacher, eine neue Kernel-Version einzuspielen, um an neuere DRM-Treiber zu gelangen.

Die Libdrm ist recht klein und leicht aktualisierbar. Die X- und Video-Treiber lassen sich normalerweise unabhängig von X-Server und Video-Bibliotheken erneuern. Gelegentlich erfordern die Treiber oder eine neu unterstützte Funktion aber auch eine frischere Umgebung, daher muss man X-Server

oder Video-Bibliotheken manchmal doch aktualisieren. Letztlich gilt daher für das Gros der Anwender: Wer frischere Open-Source-Treiber braucht, installiert am besten eine Distribution, die diese bereits mitbringt.

Intel

Die drei Open-Source-Treiberfamilien arbeiten zwar ganz ähnlich, gehen an wichtigen Stellen aber leicht anders vor. Die wenigsten Besonderheiten gibt es bei Intels Treiberfamilie, die vorwiegend von Mitarbeitern des Unternehmens entwickelt wird. Bei der Video-Beschleunigung lauert aber eine Hürde: Weil der Libva-Treiber möglicherweise lizenzpflichtige Techniken nutzt, meiden ihn manche Distributionen. Daher muss man ihn bei Fedora etwa über das Paket-Repository des RPM-Fusion-Projekts einrichten.

Vorsicht auch: Intels Treiberfamilie unterstützt den PowerVR-Grafikkern nicht, den Intel in einigen Atom-Prozessoren oder zu Atom-CPU's passenden Chipsätzen verbaut hat. Bei den allerersten Bausteinen für Atom-Netbooks oder der neuesten Atom-Generation mit Codenamen „Bay Trail“ gilt das nicht, denn sie enthalten Intel-GPUs.

Nouveau

Die Nouveau-Treiberfamilie, die nahezu alle GeForce- und Quadro-GPUs von Nvidia unterstützt, hat die Open-Source-Community entwickelt. Da Nvidia so gut wie keine Informationen zur Arbeitsweise der GPUs bereitgestellt hat, mussten die Entwickler dabei auf Reverse Engineering zurückgreifen; sie haben also die Linux- und Windows-Treiber von Nvidia analysiert, um mit den so gewonnenen Informationen die Treiber zu programmieren.

So gehen die Nouveau-Entwickler im Wesentlichen auch heute noch vor. Vor knapp einem Jahr hat sich Nvidia allerdings etwas geöffnet und den Entwicklern ein wenig Dokumentation zu Basisfunktionen seiner GPUs zur Verfügung gestellt. Wenig später hat das Unternehmen dann auch angefangen, selbst ein klein wenig an Nouveau mitzuarbeiten. Dieses Engagement beschränkt sich aber größtenteils auf Treiberteile, die zur Unterstützung der GPU im Tegra K1 nötig sind; einige

GPU-Funktionen dieses mit ARM-Kern ausgestatteten System-on-Chip (SoC) sind aber mit denen aktueller Desktop-Grafikchips verwandt, daher profitiert davon auch die Unterstützung für Grafikkarten. Nouveau hat aber eine Reihe von Qualitätsmängeln und liefert oft nur dürftige 3D-Performance; für manche Einsatzzwecke ist es aber gut genug.

DRM-, Libdrm-, Mesa- und X-Treiber heißen alle „nouveau“. Anders als Intels Mesa-Treiber greift der des Nouveau-Projekts auf die in Mesa enthaltene Bibliothek Gallium3D zurück. Gallium3D-Treiber bestehen aus „State Tracker“, „Pipe Driver“ und „Winsys“. Der State Tracker nimmt die Grafikbefehle der Anwendungen an. Typischerweise sind das OpenGL-Kommandos. Derzeit wird auch an einer Erweiterung gearbeitet, durch die der State Tracker auch DirectX-9-Befehle versteht. Bei einem so erweiterten Gallium3D bräuchte Wine die DirectX-Befehle von Windows-Anwendungen dann nicht mehr auf OpenGL umzusetzen, sondern könnte sie direkt an den Mesa-Treiber übergeben; das vermeidet Overhead und steigert die Grafikperformance.

Der State Tracker erzeugt aus eingehenden Grafikbefehlen einen Zwischencode. Er geht an den Pipe Driver, der die Aufgaben dann mit dem Grafikchip ausführt – letztlich ist er also der eigentliche Grafiktreiber. State Tracker und Pipe Driver müssen dabei mit dem Betriebssystem interagieren; alle dafür nötigen Funktionen stellt Winsys. Der State Tracker gilt auch als Frontend, Pipe Driver und Winsys als Backend. Das Frontend enthält keinen Hardware-spezifischen Code und lässt sich daher von mehreren Treibern verwenden. Das gilt auch für den meisten Code von Winsys. Gallium3D erleichtert so die Entwicklung von Grafiktreibern, da es nach oben (Anwendungen) und unten (Betriebssystem) abstrahiert. Ein sauber programmierter Pipe Driver kann daher unter anderen Betriebssystemen als Linux arbeiten.

Auch der VDPAU-Treiber von Nouveau ist ein Bestandteil von Mesa und basiert auf Gallium3D; er nutzt allerdings einen State Tracker, der VDPAU unterstützt. Dieser Treiber steckt in der Datei libvdpau_nouveau.so, bleibt bei der Standardinstallation vieler Linux-Distributionen außen vor, weil er noch sehr neu ist und bekannte Schwächen aufweist. Eine davon: Bei moderneren GPUs erfordert er eine Nvidia-Firmware, die Linux-Distributoren nicht mitliefern dürfen. Nvidia

will das ändern; bis dahin muss man die Firmware aus der Grafikkarte auslesen, nachdem man diese mit den proprietären Nvidia-Treibern initialisiert hat.

Auch für andere Beschleunigungsfunktionen ist der DRM-Treiber auf eine Firmware angewiesen, die er für die meisten GPUs aber selbst erzeugt; mittelfristig wird diese Firmware vielleicht erweitert, um auch die Video-Beschleuniger aktueller GPUs zu unterstützen. Für manche Grafikkarten kann der Treiber die Firmware nicht erzeugen, sodass man sie bei Nvidias proprietärer Treiberfamilie abgreifen muss; das ist aber meist nur bei gerade eingeführten und von Nouveau ohnehin nur rudimentär unterstützten GPUs der Fall.

Radeon

Die Grafiktreiberfamilie Radeon unterstützt die gleichnamigen Grafikkarten von AMD und wird vornehmlich von Mitarbeitern des Unternehmens vorangetrieben. Die DRM- und Libdrm-Treiber heißen wie der X-Treiber „radeon“. Der DRM-Treiber benötigt eine GPU-spezifische Firmware, die in der Firmware-Dateien-Sammlung für Linux steckt. Viele Distributionen richten die standardmäßig ein; Debian allerdings nicht.

Es gibt auch einen Mesa-Treiber namens „radeon“; er spricht allerdings nur GPUs an, die vor mehr als zehn Jahren eingeführt wurden. Für moderne Radeon-Chips sind andere Mesa-Treiber zuständig: „r600“ für GPUs mit den Radeon-HD-Modellnummern 2400 bis 7670 sowie „radeonsi“ für neuere Radeon-GPUs bis hin zur aktuellen R-Serie.

Auch diese zwei Mesa-Treiber verwenden Gallium3D. Das war bei einer früheren Fassung des R600-Treibers nicht der Fall, die Mesa-intern r600c hieß; der neuere Treiber heißt dort r600g.

Im Unterschied zu anderen Mesa-Treibern wandelt der Radeonsi-Treiber die Grafikbe-

fehle nicht selbst in Maschinencode für den Grafikchip um. Stattdessen überlässt er diese Aufgabe einem Backend der LLVM Compiler Infrastructure, das dabei Hardware-spezifische Optimierungen vornimmt. In diesem Backend steckt viel Logik, daher beeinflusst sein Funktionsumfang auch den des Mesa-Treibers; der OpenGL-3.3-Support im Radeonsi-Treiber erfordert daher LLVM 3.5 oder höher.

Auf dieses R600 genannte LLVM-Backend kann auch der R600-Mesa-Treiber für einige Aufgaben zurückgreifen. Bei neueren Distributionen ist das typischerweise der Fall. Manche Distributionen bauen das LLVM-Backend statisch in die beiden Mesa-Treiber ein. Andere nutzen LLVM als Shared-Library, daher macht ein unbedachtes Entfernen oder Aktualisieren von LLVM den Treiber kaputt.

Wie bei der Nouveau-Treiberfamilie bringt Mesa einen auf Gallium3D basierenden VDPAU-Treiber zur beschleunigten Video-Wiedergabe mit. Seit der im März veröffentlichten Version 10.2 sind auch OpenMAX-Treiber dabei. Über sie lässt sich auch der Video-Encoder ansprechen, was aber den DRM-Treiber des Linux-Kernels 3.15 oder neuer erfordert. Der wiederum benötigt dazu aktualisierte Fassungen der Radeon-Firmware und zwar schon bei der Initialisierung. Die erfolgt oft im Initramfs, daher muss man nach dem Einspielen einer neuen Firmware auch das Initramfs aktualisieren.

Noch mehr Treiber

Durch die vielen verschiedenen Treiber ist beim Aktualisieren der Radeon-Treiberfamilie am meisten Vorsicht geboten, denn der Hardware-spezifische Code verteilt sich auf acht Dateien: DRM-, Libdrm-, X-, OpenGL-, VDPAU-, OpenMAX-Treiber sowie Firmware und LLVM. Streng genommen kommen noch 32-Bit-Treiber für 3D und Video sowie OpenGL-Treiber und der Sound-Treiber snd-

hda-codec-hdmi dazu. Über Letzteren spricht der Kernel die Audio-Codex moderner Grafikkarten an, die die Audio-Wiedergabe via HDMI oder DisplayPort ermöglichen. Er unterstützt auch Intel- und Nvidia-GPUs. Oft braucht er keine oder nur minimale Änderungen, um Codex neu vorgestellter GPUs zu unterstützen.

Probleme mit einzelnen Treibern einer Familie zeigen sich gelegentlich erst bei näherem Hinsehen, weil generische Treiber einspringen, wenn ein Treiber der Familie fehlt oder nicht funktioniert. Bei Problemen mit dem passenden OpenGL-Treiber greift Mesa möglicherweise auf den Treiber LlvmPipe zurück, der 3D-Befehle mithilfe von LLVM auf dem Hauptprozessor ausführt. Für 3D-Desktops wie die Gnome-Shell und simple 3D-Anwendungen ist das schnell genug, für moderne 3D-Shooter vollkommen unzureichend.

LlvmPipe bietet sogar 3D-Beschleunigung, wenn der DRM-Treiber bockt. Die Bedienoberfläche ist dann aber oft sehr träge, weil Kernel oder X-Server zu Treibern greifen, die VESA BIOS Extensions (VBE) zur Bildschirmausgabe nutzen. Häufig passt dann auch die Bildschirmauflösung nicht zum Ausgabegerät und lässt sich nicht oder nur eingeschränkt ändern. Oft liegt dann ein Problem mit dem DRM-Treiber des Kernels vor. Um dieses zu beseitigen, muss man das System meist neu starten, denn der DRM-Treiber verlangt volle Kontrolle über den Grafikchip; die kann er aber nicht bekommen, wenn bereits ein anderer Kernel-Treiber aktiv ist. Die häufigste Ursache für so ein Problem: Ein Konflikt zwischen dem DRM-Treiber und den Kernel-Treibern der proprietären Grafiktreiber-Familien.

Proprietäre Treiber

Auch die proprietären Treiberfamilien Fglrx und Nvidia bestehen aus mehreren Treibern.

Rechnen mit der Grafikkarte

In der Standardinstallation vieler Linux-Distributionen bleiben die Treiber zum Rechnen auf Grafikkarte außen vor. Dieses „General-purpose Computing on Graphics Processing Units“ (GPGPU) ist vor allem für leicht parallelisierbare Berechnungen interessant, denn solche führen GPUs um ein Vielfaches schneller aus als CPUs; Letztere sind dafür flexibler.

Große Bedeutung hat GPGPU bei wissenschaftlichen Berechnungen, denn die lassen sich oft mit wenig Aufwand parallelisieren. Einige aufwendige 3D-Spiele verwenden die Technik zur Fluid-Simulation oder für Partikeleffekte. Auf GPGPU kann auch die Bildverarbeitungs-Bibliothek GEGL zurückgreifen, die sich in aktuelle Gimp-Versionen einbinden lässt; das derzeit entwickelte Gimp 2.10 soll sie mitbringen. Das Gros der Software für Desktop-PCs lässt GPGPU aber links liegen.

Die wichtigsten Programmierschnittstellen zur Verwendung von GPGPU sind CUDA und OpenCL. Ersteres ist eine Nvidia-eigene Technik, die im professionellen Bereich allerlei Anwender hat und von Nvidias proprietären Grafiktreiber für Linux und Windows un-

terstützt wird. OpenCL ist ein offener Standard, den die Khronos Group vorantreibt, die sich auch um OpenGL kümmert.

Linux-Anwendungen nutzen OpenCL typischerweise über den Loader ocl-icd, der lediglich den Weg zu verschiedenen OpenCL-Implementationen bahnt. Mesa bringt eine solche mit, die unter dem Schlagwort Gallium Compute entwickelt wurde. Sie besteht im Wesentlichen aus dem State-Tracker Clover, mit dem Gallium3D-Treiber eine OpenCL-Unterstützung anbieten können. Darüber offeriert Nouveau eine experimentelle und unfertige OpenCL-Unterstützung. Der OpenCL-Support von R600 und Radeonsi gilt auch noch nicht als stabil, soll aber schon ganz ordentlich arbeiten. Noch etwas besser soll Intels unabhängig von Mesa entwickelte Open-Source-OpenCL-Implementation Beignet funktionieren.

Zu Nvidias proprietärer Treiberfamilie gehört nicht nur ein Treiber für CUDA, sondern auch einer für OpenCL. AMDs proprietäre Treiberfamilie unterstützt nur OpenCL. Beide Unternehmen offerieren genau wie Intel jeweils ein Software Development Kit, das Programmierern die Entwicklung von GPGPU-Software erleichtern soll.

Grafikchip- und Treiberwahl

Auf 3D-Performance bedachte Anwender greifen zu GeForce-Grafikchips und Nvidias proprietären Linux-Treibern, alle anderen kaufen Intel – Aussagen dieser Art hört man häufiger, wenn es um die Frage nach der besten Grafikkarte für Linux geht. So simpel war es aber nie; zudem sind Radeon-GPUs interessanter geworden, denn die quelloffenen AMD-Treiber haben aufgeholt. Ohnehin hat jeder der fünf bei PC-Linux gängigen Grafiktreiber-Familien einige Fehler und Unzulänglichkeiten; alle Ansprüche kann daher keiner zufriedenstellen.

Zur Nutzung mit Unity, Gnome oder anderen 3D-Desktops sind alle fünf gut genug. Bei Nouveau können sich aber schon dabei Schwächen zeigen, denn die Treiberfamilie unterstützt die Lüfterregelung vieler Grafikkarten nicht oder nur rudimentär; die Lüfter laufen dann manchmal mit Höchstgeschwindigkeit, was Krach macht und den Lagerverschleiß erhöht. Das und eine dürftige 3D-Performance sind keineswegs die einzigen Mankos, daher greifen viele zu Nvidias proprietärer Treiberfamilie, die alle wichtigen Grafikchip-Funktionen unterstützt und vielfach gut läuft – aber eben nicht immer.

Intel-Grafik macht weniger Probleme, denn Linux-Distributionen richten die zugehörige Treiberfamilie automatisch ein und aktualisieren diese auch. Die Treiber unterstützen auch alle wichtigen Funktionen. Allerdings liefern selbst Intels leistungsfähigste GPUs nur eine 3D-Leistung auf dem Niveau von GeForce- und

Radeon-Grafikkarten der 60-Euro-Klasse [2]. Vielen Spielern reicht das nicht.

Genau da punktet AMD neuerdings, denn viele Radeon-Karten und die Radeon-GPUs einiger AMD-Prozessoren liefern mehr 3D-Performance. Die passende Open-Source-Treiberfamilie stammt wie bei Intel vom Hersteller selbst, denn AMD ist vor sechs Jahren in die Entwicklung quelloffener Treiber eingestiegen. Seit einigen Monaten lassen sich mit ihnen nun auch die Video-Beschleuniger verwenden. Zudem werden neue Radeon-GPUs nun zügiger unterstützt als noch vor ein oder zwei Jahren – ähnlich wie bei Intel dauert es aber manchmal ein Weilchen, bis alle Mainstream-Distributionen die bessere Hardware-Unterstützung mitliefern. Auch der quelloffene 3D-Treiber hat aufgeholt, denn er holt aus einigen zwei oder drei Jahre alten GPUs fast so viel Performance heraus wie AMDs proprietäre 3D-Treiber; bei neueren GPUs ist der Unterschied größer, aber vielfach verschmerzbar.

Wenn dann doch das Maximum an 3D-Performance gefragt ist oder irgendwo was klemmt, dann kann man zu AMDs proprietärer Treiberfamilie greifen. Die hat zwar viel mehr Ecken und Kanten als Nvidias, läuft aber dennoch oft akzeptabel. Bei der OpenGL-Unterstützung sind die proprietären 3D-Treiber indes viel weiter und implementieren bereits aktuelle OpenGL-Versionen der 4er-Reihe; bei den quelloffenen 3D-Treibern ist es maximal OpenGL 3.3.

Die von Nvidia arbeiten mit denen von Nouveau aber ebenso wenig zusammen wie umgekehrt. Ebenso verhält es sich zwischen den quelloffenen und proprietären Teilen der Treiberfamilien für AMD-GPUs – das soll sich aber vielleicht bald ändern.

Die proprietären Treiber docken an ähnlichen Stellen an wie die Open-Source-Treiber, gehen in wichtigen Bereichen aber anders vor. So gibt es auch bei Fglrx und Nvidia einen Kernel-Treiber, der letztlich der Vermittler zur Grafikkarte ist, auf den andere Teile der Treiberfamilie angewiesen sind. Der Kernel-Treiber klinkt sich allerdings nicht beim DRM des Kernels ein und beherrscht auch kein KMS. Bei Systemen mit proprietären Grafiktreibern greift der Kernel daher auf generische Mechanismen wie VGA und VBE zurück, um Statusmeldungen oder Text-Log-In anzuzeigen.

Der Kernel-Treiber wird typischerweise erst beim Anlaufen des X-Servers geladen, weil der proprietäre X-Treiber ihn anfordert. Die Konfigurationsautomaten des X-Servers bevorzugen bei den allermeisten Distributionen allerdings die quelloffenen X-Treiber. Der Einsatz proprietärer X-Treiber erfordert daher oft eine Konfigurationsdatei. Sie kann man mit Kommandozeilentools anlegen, die AMD und Nvidia mitliefern.

Mangels KMS stellt der X-Treiber selbst eine Bildschirmkonfiguration ein, die zu den Ausgabegeräten passt. Auch die X-Treiber der proprietären Treiberfamilien unterstützen RandR, daher funktionieren mit ihnen auch xrandr oder die Anzeige-Einstellungen von Gnome und KDE. AMD und Nvidia liefern zudem grafische Konfigurationsprogramme mit. Sie bieten deutlich mehr Einstellmöglichkeiten, modifizieren dafür

aber manchmal die X-Konfigurationsdatei, die der X-Server nur bei Neustarts verarbeitet.

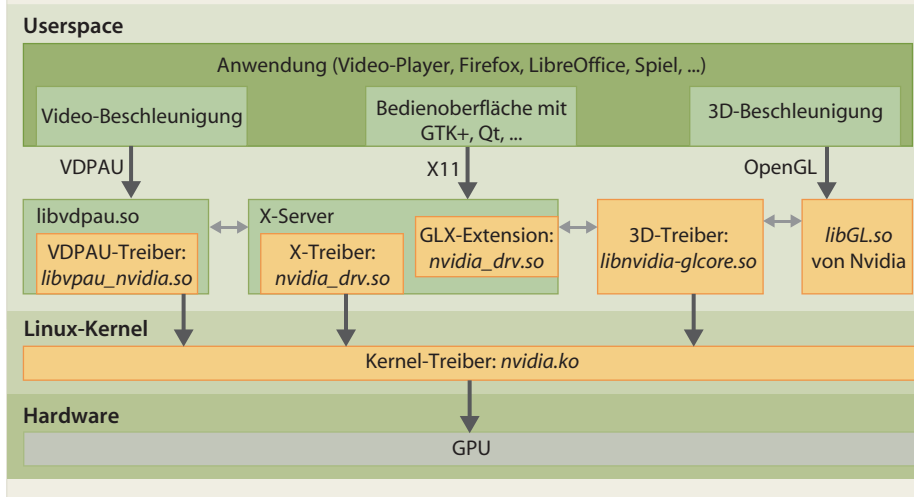
Die proprietären 3D-Treiber docken nicht bei Mesa an. Da Anwendungen normalerweise auf die LibGL von Mesa zurückgreifen, wenn sie OpenGL-Beschleunigung nutzen, richten die Installer eine eigene LibGL ein. Über sie finden Anwendungen den Weg zu einer Bibliothek, die den 3D-Treiber von AMD und Nvidia enthält. Der muss sich bei der Bildausgabe mit dem X-Server koordinieren, was wie bei den quelloffenen Treiberfamilien per GLX erfolgt. Den proprietären Treibern reicht allerdings das zum X-Server gehörende GLX-Modul nicht, daher richten die Installer ein eigenes ein.

Zur beschleunigten Video-Wiedergabe legt Nvidia einen VDPAU-Treiber bei. Die von AMD als Catalyst bezeichnete Fglrx-

Anzeige

Die proprietären Treiber für Nvidia-Grafik

Bei Nvidias proprietärer Treiberfamilie müssen Programme zur Video-Beschleunigung auf VDPAU statt VA API zurückgreifen. Damit Anwendungen den richtigen 3D-Treiber verwenden, ersetzt Nvidias Installer die GLX-Extension des X-Servers und die LibGL-Bibliotheken der Distribution durch eigene. Der Kernel-Treiber ist kein Bestandteil des Distributions-Kernels, sondern muss passend erzeugt werden.



Treiberfamilie unterstützt keines der drei erwähnten Video-APIs, sondern setzt auf XvBA (X-Video Bitstream Acceleration). Keine der bekannten Multimedia-Anwendungen unterstützt diese von AMD selbst erfundene X-Video-Erweiterung. Es gibt allerdings einen VA-API-Treiber, der auf XvBA zurückgreift. Dieser von der Open-Source-Community erfundene Treiber wird aber schon seit drei Jahren nicht mehr weiterentwickelt; Treiber und XvBA selbst gelten als instabil.

Abhängigkeiten

Die verschiedenen Treiber der proprietären Treiberfamilien müssen exakt zusammenpassen; der X-Treiber des Nvidia-Treibers 331.89 arbeitet daher nicht mit dem Kernel-Modul zusammen, das zur Version 331.79

gehört. Das ist eine von vielen Stolperfallen, die Nutzern immer wieder Schwierigkeiten bereiten. Eine weitere: Das Kernel-Modul von Fglrx und Nvidia muss genau zum verwendeten Kernel passen, daher übersetzt der Treiber-Installer oder die Pakete der Distributionen zumeist ein passendes Modul; dieses Prozedere muss jedes Mal auf Neue durchlaufen, wenn die Systemaktualisierung einen neuen Kernel mitbringt.


Auch die Libgl- und GLX-Dateien der proprietären Treiberfamilien sorgen immer wieder für Probleme, denn die Installer der Hersteller ersetzen die gleichnamigen Dateien der Distribution; bei einem Sicherheitsupdate für Mesa oder X-Server landen dann möglicherweise wieder zu Open-Source-Treibern passende Dateien auf der Platte, woraufhin die proprietären

Treiber nur noch eingeschränkt oder gar nicht mehr arbeiten.

Die Installation der Nvidia- und Fglrx-Treiberfamilien über die Software-Verwaltung von Ubuntu nutzt Tricks, um solche Probleme zu vermeiden und es Anwendern möglichst einfach zu machen; das gilt auch für die meisten Wege, über die sich die proprietären Treiber bei anderen Distributionen einrichten lassen. Vielfach legen sie auch gleich eine X-Konfigurationsdatei an, damit der X-Server den proprietären X-Treiber nutzt. Häufig deaktivieren sie zudem die DRM-Treiber von Radeon und Nouveau, damit die sich den Grafikchip nicht krallen, bevor es der Kernel-Treiber von Nvidia oder Fglrx tut.

Ein anderes Problem können die Distributions-spezifischen Wege nur mindern, aber nicht aus der Welt schaffen: Wenn die Kernel-Entwickler mal wieder die internen Treiber-APIs des Linux-Kernels verändert haben, schlägt das Kompilieren des proprietären Kernel-Treibers womöglich fehl. Dann muss man auf Korrekturen durch die Open-Source-Community oder AMD und Nvidia warten. Gerade AMD ist da träge, daher ist der Kernel-Treiber von Fglrx häufig inkompatibel zu aktuellen Kernel-Versionen. Auch mit neuen X-Servern arbeiten Fglrx häufig nicht oder erst nach Monaten zusammen, denn mit fast jedem größeren Versions-schritt des X-Servers ändert sich dessen ABI (Application Binary Interface), über das Treiber andocken. Irgendwann hören die Hersteller auf, die proprietären Treiber für ältere Grafikchips an neue Kernel-APIs und X-ABIs anzupassen; AMD meist früher als Nvidia. Mit Distributionen, die ein Jahr später erscheinen, arbeiten die Treiber dann oft schon nicht mehr zusammen. (thl)

Literatur

- [1] Thorsten Leemhuis, Treiberhilfe, Aktuelle Kernel-Treiber mit älteren Kernen verwenden, c't 21/13, S. 160
- [2] Martin Fischer, GPU-Shopping, Die richtige Grafikkarte finden, c't 10/2014, S. 110 

Anzeige