

Grafiktreiber im Linuxkernel – die Außenseiter

Lucas Stach
18. Januar 2011

Abstract

Moderne Grafikkarten integrieren vielfältige Funktionen und entsprechend komplex sind auch die zugehörigen Treiber. Essentielle Teile dieser Treiber residieren gar im Userspace und haben mit dem Kernel nur über verschiedene Interfaces zu tun.

Wie ist der Linux Grafikstack aufgebaut und wieso ist er so komplex? Wie sehen die aktuellen Entwicklungen aus? Wo wird es in Zukunft hin gehen? Ist die Entwicklung dieser Treiber nur etwas für Gurus, oder kann auch ich etwas beitragen?

Auf all diese Fragen versucht der Vortrag eine umfangreiche Antwort zu geben.

Zielgruppe und Umfang

Dieser Vortrag richtet sich an Linux Anwender, welche sich für neue Entwicklungen im Kernel interessieren. Kernelentwicklungseinsteigern wird ein Überblick über ein Betätigungsfeld geboten, zu dem die Informationen meist sehr verstreut sind. Es sollen Einstiegspunkte aufgezeigt werden, von denen aus sich der interessierte Teilnehmer weiter informieren kann.

Inhalt

Der Linuxgrafikstack besteht aus mehreren getrennten Komponenten, die meist in bekannter Unixmanier von verschiedenen Entwicklerteams bearbeitet werden. Dazu zählen in erster Linie die Direct Rendering Manager (DRM) Kernelmodule, der X Server und dessen Device Dependent X (DDX) Treiber. Dazu kommt die Mesa3D Bibliothek für OpenGL Unterstützung.

Seit 2006 hat die Entwicklung in diesem Bereich wieder an Geschwindigkeit gewonnen. Seit dem machen Schlagworte wie Kernel Mode Setting (KMS), Translation Table Maps (TTM), Graphics Execution Manager (GEM) und Gallium3D die Runde.

Der Verbund all dieser Komponenten mit ihren klar definierten Aufgaben bildet zusammen einen Stack, der nach Ansicht der Entwickler den Anforderungen der nächsten Jahre gewachsen ist.

Die Entwicklung schreitet schnell voran, wie man an den zahlreichen Commits ablesen kann, die in jede neue Kernelversion einfließen. Trotzdem hinken die freien Treiber den closed-source Versionen, welche von den Chipherstellern bereit gestellt werden, meist weit hinter her, da der Arbeitsaufwand für die wenigen beteiligten Entwickler einfach zu groß ist.

Eine der großen Herausforderungen, die es noch zu meistern gilt, ist die Umsetzung des General-Purpose computation on Graphics Processing Units (GPGPU) Konzepts. Dabei werden Teile der Berechnungen, welche bisher von der CPU ausgeführt werden auf den Grafikprozessor verlagert, da dieser bei der Berechnung von großen Datenströmen effektiver arbeiten kann. Dazu wurde die plattformübergreifende Schnittstelle OpenCL geschaffen, welche unter Linux bisher aber nur ansatzweise implementiert ist.

Weitere Informationen

- <http://www.x.org/wiki/>
- <http://mesa3d.sourceforge.net/>
- <http://nouveau.freedesktop.org/wiki/>
- <http://www.phoronix.com/>