Tux bringt Spiele auf Trab

Linux-Kernel 5.16 optimiert die Speicherverwaltung

Eine neue Thread-Synchronisierung im Linux-Kernel vereinfacht es, Windows-Spiele unter Linux performant auszuführen. Linux 5.16 widmet sich auch der Mammutaufgabe, durch Folio-Pages eine effiziente Speicherverwaltung zu schaffen.

Von Oliver Müller

Mitte Januar hat Linus Torvalds die jüngste Version des Linux-Kernels veröffentlicht. Linux 5.16 gibt den Startschuss für den Locking-Mechanismus Futex2 und verspricht insbesondere bei der Ausführung von Windows-Spielen unter Linux bessere Performance.

Futex steht für „Fast User Mutex“. Es erlaubt Sperren (Mutexes) für den konkurrierenden Zugriff nicht nur im Kernel-Space, sondern auch im Userspace zu verwenden. Ein Aufruf von `futex()` blockiert den Prozess beziehungsweise dessen Thread, bis die betreffende Ressource frei wird. Futex2 soll auf lange Sicht den in die Jahre gekommenen und historisch gewachsenen System-Call `futex()` ablösen. Von einem umfassenden Ersatz für das alte `futex()` ist Futex2 in Linux 5.16 noch entfernt. Der eingeflossene Ansatz ist eine Light-Variante, die ein spezifisches Problem adressiert.

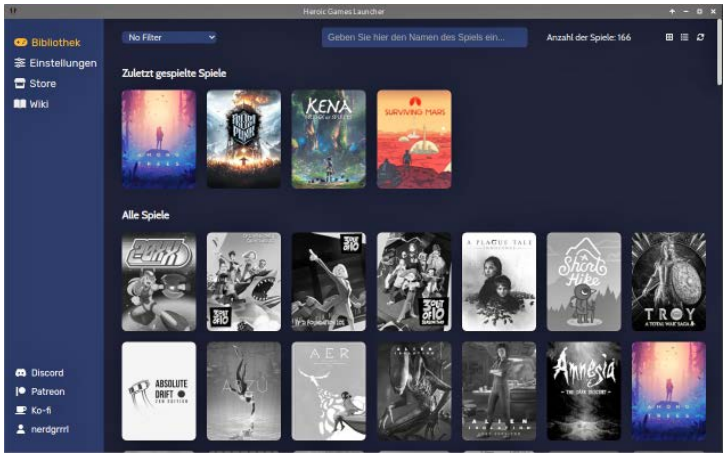
Die bisherige Funktion kann lediglich einen Futex abfragen. Sind mehrere zu berücksichtigen, geht das nur hintereinander, mit dem Manko, in mehrere blockierende Aufrufe von `futex()` in Serie zu laufen. Das hat gerade bei Windows-Spielen Nachteile, die mit einer Kompatibilitätsschicht wie WINE oder Proton ausgeführt werden.

Mimikry der Windows-API

Denn das Windows-API bietet die Funktion `WaitForMultipleObjects()`, die das Abfragen mehrerer Objekte erlaubt. Diese Objekte lassen sich analog zu den Ressourcen und Futexes in Linux verwenden. Windows kann die Abfrage aber in einem Aufruf parallel durchführen, Linux bislang nicht. An dieser Stelle setzt Linux 5.16 mit dem ersten Futex2-Patch an. Der Kernel führt den System-Call `futex_waitv()` ein, der auf mehrere Mutexes beziehungsweise deren Freigabe gleichzeitig warten kann. Damit kann Linux 5.16 für Kompatibilitätsschichten die Funktion `WaitForMultipleObjects()` eleganter und effizienter nachbilden. Windows-Spiele unter Linux profitieren von einem Geschwindigkeitsgewinn.

Speicherverwaltung optimiert

Linux 5.16 krempelt die Speicherverwaltung gehörig um, indem es die sogenannten Folio-Pages einführt. Die machen aufwendige Prüfungen beim Adressieren von Speicherseiten überflüssig, erfordern aber auch umfassende Umbauten in unzähligen Speicherfunktionen.



Unter Linux ausgeführte Windows-Spiele sollen von Futex2 profitieren, das eine Thread-Funktion von Windows nachbildet.

Zum Verständnis: Virtuellen Arbeitsspeicher verwaltet Linux mit Speicherseiten (Pages). Die Größe der Pages spielt für die Leistung des Systems eine Rolle, aber die optimale Größe hängt von der genutzten CPU-Architektur ab. Da der Linux-Kernel auf einer Vielzahl unterschiedlicher Architekturen läuft, nutzt es intern zu Verwaltungszwecken eine (logische) Basisgröße für Pages.

Um die Hardware-implizierte Page intern abzubilden, verwendet der Linux-Kernel Verbünde von Pages (Compound-Pages). In diesen Verbünden hängt der Kernel logische Basis-Pages aneinander, um auf die Größe der physischen Seiten zu kommen. Den Anfang des Verbundes repräsentiert eine Head-Page; alle folgenden Pages im Verbund sind die Tail-Pages.

Wird einer Kernel-Funktion eine Tail-Page als Parameter übergeben, muss zunächst die Head-Page ermittelt werden. Die dafür verwendete Funktion `compound_head()` ist, um wenigstens ein paar Taktzyklen zu sparen, als Inline-Funktion umgesetzt. Im ausführbaren Kernel findet kein echter Funktionsaufruf statt, sondern der Code wird stattdessen an die Stelle hineinkopiert. Das bläht den Kernel schließlich in Summe auf.

Das Folio-Konzept beseitigt diese Ineffizienz. Eine Folio-Page ist nichts anderes als eine Page, die garantiert eine Head-Page ist. Prüfungen und Konvertierungen mit `compound_head()` fallen weg. Gerade bei vielen Funktionen des Kerns der Speicherverwaltung, aber auch bei Dateisystemen verschlankt das den Code und hilft die Leistung zu steigern.

Der Ansatz wurde lange kontrovers diskutiert, denn er zieht einen Gewaltakt nach sich. Einerseits wird eine Parallelwelt für die Page-Verwaltung geschaffen. Neben den bestehenden Page-Funktionen sind zusätzliche Folio-Funktionen notwendig. Zudem müssen Unmengen von Kernel-Funktionen auf das API angepasst werden. Die potenzielle Leistungssteigerung überzeugte schließlich und ließ das Kernel-Team in Linux 5.16 den Schritt wagen. *(ktn@ct.de)* c't

© Copyright by Heise Medien.
Persönliches PDF für alle Leser der c't aus 30625 Hannover