

Kernel 2.6 - Ein Ausblick

Daniel Mahrenholz, MDLUG e.V.



Vortrag zum Themenabend
19. August 2003

- Überblick der Neuerungen
- Altlasten – Was fliegt raus?
- Nutzen – Wer hat welche Vorteile?

Was bringt der 2.6er Kernel neues mit sich?

- Unterstützung neuer Plattformen
 - ★ Sehr grosse und sehr kleine Systeme
 - ★ Subarchitekturen
- Technologische Verbesserungen
 - ★ Skalierbarkeit (CPUs, Dateisysteme, Benutzer, Prozesse, Speicherausbau, ...)
 - ★ Scheduler (CPU, I/O), Hyperthreading, POSIX-Threads
 - ★ Modulsystem, vereinheitlichte Behandlung von Geräten
 - ★ Netzwerkunterstützung, Dateisysteme
 - ★ Sicherheit

- Integration der *uCLinux*-Entwicklung in den Standard-Kernel
- Unterstützung für Systeme ohne MMU (*Memory Management Unit*)
 - ★ Kein Speicherschutz
 - ★ Keine Multi-User-Unterstützung
 - ★ Keine Prozesse, nur Threads

~> entsprechender Code komplett entfernt
- Möglichkeit, Systeme komplett ohne Swap- oder Display-Unterstützung zu bauen

- Unterstützung Opteron ohne Limitierungen des 2.4er Kernels
- Unterstützung von 64bit-PowerPC-Systemen
- Unterstützung für NUMA-Systeme (*Non-Uniform Memory Access*)
 - ★ Effizienteres Design für Mehrprozessorsysteme
 - ★ Jede CPU kann auf kompletten Speicher zugreifen, **aber** je CPU existiert ein Speicher, auf den besonders effizient zugegriffen werden kann
 - ★ Analog kann Zugriff auf I/O-Geräte unterschiedlich effizient sein
 - ★ Scheduler kennt Systemkonfiguration und ordnet Prozesse geeignet zu
- Erhöhung der Grenzen für Anzahl der Prozesse, Nutzer-/Gruppen-IDs, Dateisysteme
- O(1)-Scheduler
- 64GB-RAM auf moderner 32bit-Intel-Hardware

- Unterscheidung verschiedener Systeme innerhalb einer Prozessorfamilie
 - ★ i386: generisch, NCR Voyager, PC-9800 (NEC), SGI Visual Workstation
 - ★ m68k: Amiga, Macintosh, ...
- Nur Unterteilung bei grundlegenden Unterschieden der Hardware
 - ★ XBOX ist eine generische i386-Plattform
- Konzept erlaubt klare Trennung gemeinsamer und unterschiedlicher Systemteile

- Prozess-IDs: 2^{30} statt 2^{15}
 - ★ Ermöglicht viel mehr gleichzeitig laufende Prozesse
 - ★ Verhindert, dass PIDs zu schnell wiederverwendet werden
- Nutzer- und Gruppen-IDs: 2^{32} statt 2^{16}
- 64bit-Adressierung für Blockgeräte (auch auf 32bit-Systemen, dort mit Einschränkungen)
- Dateisystemgrenzen:

Feature	Ext3	Reiser4	JFS	XFS
max. Blockgrösse	4 KB	4 KB	4 KB	4 KB
max. Dateisystemgrösse	16384 GB	17592 GB	18000 PB	32 PB
max. Dateigrösse	2048 GB	1 EB	9000 PB	4 PB

- Eher kosmetisch: Endung für Module `.o` → `.ko`
- Verhinderung diverser Nebenläufigkeitsprobleme (*race conditions*)
 - ★ Überlappung des Ladens/Entfernes von Modulen, die gleiches Gerät benutzen
- Entladen von Modulen kann selektiv verhindert werden
- Zusammenlegung des Codes zur Erkennung von Geräten an den Systembussen
 - ★ Treiber muss nicht mehr selber suchen (PCI, ISA PnP, PC Card)
 - ★ Hardware-Erkennungstools können einfacher die Gerätekennungen zentral auslesen
- Module spezifizieren, welche Geräte sie unterstützen
 - ↪ Software kann besser entscheiden, welche Module zu laden sind
- Laden eines Treibers kann aber dennoch erzwungen werden

- *Overcommitment*, d.h. Vergabe von mehr Speicher als vorhanden, nicht mehr möglich
- Speicherverwaltung kann mit Lücken im physischen Adressraum umgehen
 - ★ Oftmals bei NUMA-Systemen zu finden
 - ★ Verwendung für Hochverfügbarkeitssysteme, die Speicherfehler zur Laufzeit feststellen
 - ★ Nutzung teilweiser defekter Speicherbausteine
- Volle Unterstützung für Intels PAE (*Physical Address Extension*) zur Nutzung von bis zu 64GB RAM auf vielen neuen 32bit x86-Systemen
- rmap (*Reverse Mapping Virtual Memory System*)
 - ★ Direkte Abbildung von Speicherseiten auf PTEs (*Page Table Entries*)
 - ★ Beschleunigt Freigabe unbenutzter Speicherseiten deutlich
- Large Page Support: Nutzung grosser Speicherseiten für kleine PTs

- Unterbrechbarer Kernel (feingranulares Sperren von Kernelteilen)
 - ★ Nutzerprozesse können laufen, selbst wenn Kernel auf Ereignisse wartet
 - ★ Schnelleres Reagieren auf externe Ereignisse möglich
 - Futex (Fast User-Space Mutex)
 - ★ Synchronisationsmittel für Prozesse und Threads
 - ★ Kernelunterstützung nur im Konfliktfall benötigt
 - ★ Unterstützung von Prioritäten
 - Verringerung von I/O-Latenzen
 - ★ Verhinderung des „Verhungerns“ von Prozessen bei hoher Last
 - Scheduler arbeitet mit 1000 Hz Umschaltfrequenz
- ⇒ Verbesserung der Reaktionsfreudigkeit des Systems speziell bei hoher Last

- Getrennte Warteschlangen
 - ★ Echtzeit-Prozesse teilen sich eine Warteschlange
 - ★ Aktive, abgelaufene Prozesse in 2 Warteschlangen pro CPU
- Sortierung der Warteschlangen nach Priorität \leadsto konstante Zeit bei der Auswahl des nächsten Prozesses (unabhängig von der Anzahl der Prozesse)
- Skaliert sehr gut auf Mehrprozessorsystemen
- Systeme mit sehr vielen Prozessen/Threads profitieren überdurchschnittlich
- Scheduler versucht, Prozesse möglichst an CPUs zu binden, um Cache-Verhalten zu optimieren
- Hyperthreading wird wie NUMA behandelt

- Komplette Überarbeitung der Thread-Infrastruktur
 - ★ User-Threads benutzen Kernel-Threads (1:1; n:1; n:m, $n \geq m$)
 - ★ 100.000 Kernel-Threads ohne Probleme möglich
- Ermöglicht Nutzung der NTPL (*Native POSIX Thread Library*)
 - ★ Erhebliche Beschleunigung von Applikationen mit vielen Threads
 - ★ Backport für 2.4er Kernel verfügbar
 - ★ Noch keine Nutzung durch aktuelle Java-Implementierungen
- Alternativ: NGPT (*Next Generation POSIX Threads*)
 - ★ Benutzt Erweiterungen für NTPL, deshalb keine separaten Kernel-Änderungen notwendig
 - ★ Vergleich beider Versionen schwierig, da sehr versionsabhängig

- Infrastruktur zur Verwaltung von Geräten und Treibern
 - ★ Speichert Ressourcenbelegung der Treiber (I/O-Ports, IRQs, DMA-Kanäle, ...)
 - ★ Vereinheitlicht Zugriff auf generische Funktionen der Geräte (z.B. Änderung des Energiemodus)
 - ★ Zusammenfassung von Funktionen einer Geräteklasse
- Viel mehr als die UNIX-typische `read/write/sysctl`-Schnittstelle
 - ★ Vereinheitlichung sowohl intern als auch beim Zugriff von aussen

Kernel Object Abstraction

- Objektorientierte Schnittstelle, über die die Low-Level-Treiber (z.B. Busse) Geräte und Untergeräte (z.B. Partitionen einer Platte) abbilden und verwalten
 - High-Level-Treiber (z.B. Power-Management) erhalten so zentralen Zugriff auf Hardware-Informationen, um den Zustand aller Geräte zu ändern, die dies unterstützen
 - Alle Geräte werden als Hot-Plug behandelt
 - ★ Nach der Erkennung wird ein *Kernel Object* erzeugt
 - ★ Keine Beschränkung mehr auf Hot-Plug-PCI, USB, Firewire, PC Card
 - ★ Beim Systemstart wird nach den vorhandenen Geräten gesucht
- ↪ Deutliche Vereinfachung der Behandlung von Hot-Plug-Geräten

The System Filesystem

- Dateisystem (sysfs), das Gerätebaum nach aussen repräsentiert (/sys)
- Analog:
 - ★ proc für Prozesse
 - ★ devfs für Gerätetreiber
 - ★ devpts für UNIX98 Pseudo-Terminals

- Systembusse
 - ★ ISA PnP: grundlegend überarbeitet
 - ★ PCI: Verbesserung Hot-Plug-Fähigkeit, Power-Management
 - ★ AGP: mehrere Geräte möglich
- Externe Bussysteme
 - ★ USB 2.0, USB On-the-Go (wahrscheinlich)
- Wireless
 - ★ Integration aller Technologien in ein Subsystem (ein API; nur ein Satz User-Space-Tools)
 - ★ Bluetooth und IrDA wesentlich überarbeitet

- Blockgeräte
 - ★ IDE: komplett neu für bessere Skalierbarkeit, Beseitigung von Einschränkungen
 - * CD/RW ohne Emulation direkt nutzbar
 - * Auslesen von (Timing-)Parametern aus unbekanntem Controllern
 - ★ SCSI: Unterstützung für Geräte mit mehr als 2 LUNs
- Hardware Sensor Drivers (Im_sensors)
 - ★ Überwachung der Hardware und Umgebungsbedingungen (Temperatur, Spannung, Energieversorgung, . . .)

- Unterstützung für Windows' Logical Disk Manager („Dynamic Disks“)
- Unterstützung für erweiterte Attribute, Meta-Daten und POSIX-ACLs
- Synchrone Verzeichnisse: Möglichkeit, atomare Operationen zu erzwingen
- EXT3:
 - ★ Commit-Interval anpassbar (sehr hilfreich für Laptop-Nutzer)
 - ★ Default-Optionen können im Dateisystem gespeichert werden
 - ★ Suchindex für Verzeichnisse möglich
- NTFS mit Lese-/Schreibzugriff
- FAT12: Workarounds für Bugs in verbreiteter Hardware (z.B. MP3-Player)
- XFS: Platten-kompatibel mit IRIX-Systemen

Human Interface Devices

- Alle Nutzerschnittstellen (Tastatur, Maus, Display, . . .)
- Voll modularisiert und auch entfernbar
- Framebuffer-Unterstützung
 - ★ Skalierung, Rotation möglich (wichtig für PDAs)
 - ★ Hardware-Beschleunigung auf vielen Systemen, Abfrage Display-Informationen
- Unterstützung für Touch-Screens, Tieman Voyager Braille TTY
- Eingabe:
 - ★ Mäuse immer als `/dev/input/mouse0`, unabhängig von der Hardware
 - ★ Erweiterte (Multimedia-)Tasten, Gamepads, Force-Feedback-Geräte

Multimedia

- ALSA (*Advanced Linux Sound Architecture*) im Kernel
 - ★ Zahlreiche neue Hardware unterstützt
 - ★ Voll-Duplex-Unterstützung, Mischen von Quellen, ...
- Video4Linux-, Radio-Unterstützung grundlegend überarbeitet
 - ★ Neue API; inkompatibel mit alter Version
 - ★ Verbesserung für viele Webcams, digitale Videorekorder
 - ★ Unterstützung für DVB-Hardware (z.B. in Settop-Boxen)
- Asynchrone I/O-Operationen
 - ★ Programme können weiterarbeiten, während Daten gelesen/geschrieben werden
 - ★ Fertigstellung wird signalisiert
 - ★ Software muss speziell dafür geschrieben sein

- IPsec für IPv4 und IPv6
- Verbesserung Multicast-Unterstützung
 - ★ Verschiedene neue SSM (*Source Specific Multicast*) Protokolle wie z.B. MLDv2 (*Multicast Listener Discovery*) und IGMPv3 (*Internet Group Messaging Protocol*)
- IPv6 in Token Ring-Netzwerken
- Stabile VLAN (*Virtual Bridged Local Area Network*; IEEE802.1Q) Unterstützung
- Verbessertes NAT/Masquarading für Protokolle mit mehreren Verbindungen (H.323, PPTP, . . .)
- Zusammenlegung, Vereinheitlichung diverser Treiberfunktionen

- NFSv4
 - ★ Client und Server, aber nicht alle Features der SUN-Implementierung
 - ★ Verschlüsselung, Transport über TCP (auch als Root-FS)
 - ★ Zero-Copy-Implementierung bei passender Hardware
- Verbesserung der SMB/CIFS-Unterstützung (auch SMB-UNIX-Erweiterung)
- Neuer NCP (*Netware Core Protocol*) Treiber, für bessere Netware-Unterstützung
- AFS (*Andrew Filesystem*) read-only nutzbar
- InterMezzo (AFS Nachfolger) vollständig
 - ★ Unterstützt Offline-Nutzung zwischengespeicherter Dateien
 - ★ Geeignet für Hochverfügbarkeitssysteme, PDA-/Laptop-Synchronisation mit Netzwerkserver

- LSM (*Linux Security Modul*)
 - ★ Zugriffsbeschränkungen auf den Kernel modularisiert
 - ★ Modell des UNIX-Superusers ersetzbar
 - ★ Alle Kernel-Komponenten benutzen *Capabilities*
- Binary-Only-Treiber können keine Kernel-Funktionen mehr überladen und unterliegen weiteren Einschränkungen
- Nutzung von Hardware-Zufallszahlengeneratoren, wenn vorhanden
- Crypto-API
 - ★ Sammlung kryptografischer Funktionen, die von verschiedenen Subsystemen/Treibern benötigt werden
 - ★ Vermeidung von Code-Duplikaten, weniger Fehlerquellen
 - ★ Inspiriert vom Crypto-Patch der 2.4er Serie
 - ★ Nutzung in CryptoFS, HostAP, Virtual Memory Encryption, CIPE (crypto IP encapsulation), Crypto-Swap, . . .

- Integration des UML (*User Mode Linux*)
 - ★ Kernel kann auch als einzelner Nutzerprozess laufen (auch mehrfach)
 - ★ Virtuelle Hardware wird auf reale Hardware abgebildet
- Verbesserungen für Laptop-Nutzer
 - ★ Verbesserungen APM-, ACPI-, Wireless-Unterstützung
 - ★ Suspend-to-Disk vollständig in Software möglich
 - ★ Speed Stepping und verwandte Technologien werden benutzt zum Stromsparen
- xconfig (Tcl/Tk) wurde durch QT-basiertes grafisches Konfigurationstool ersetzt
- Viele Funktionen verschiedener Erweiterungen wurden als Backport schon im 2.4er Kernel getestet

- Export der Tabelle der System Calls
 - ★ System Calls nicht mehr beliebig überladbar
 - ★ Neue API erlaubt GPL-kompatiblen Treibern, System Calls dynamisch anzulegen oder zu ändern (SMP fähig)
- Task Queues
 - ★ Speichern verzögerte Systemaktionen (z.B. Top-Half-Routinen)
 - ★ Laufen ausserhalb eines Prozess-Kontextes
 - ★ Ersetzt durch *Work Queues*
 - * Sammlung von Threads auf einer CPU
 - * Laufen im Prozess-Kontext \rightsquigarrow Prozesse können schlafen gelegt werden
 - * Nur GPL-kompatible Treiber können eigene Queues anlegen
- khttp / TUX (Redhat Content Accelerator)
 - ★ Webserver (für statische oder dynamisch vorgenerierte Seiten) im Kernel
 - ★ Durch viele Verbesserungen der Kernel-Userspace-Kommunikation überflüssig
 - ★ Ersatz z.B. durch x15-Server

Wer profitiert wovon?

- Mehrprozessorsysteme
 - ★ O(1)-Scheduler, NUMA-Unterstützung, 64bit-Adressierung, feingranulares Locking
- Heimanwender
 - ★ ALSA, feingranulares Locking, preemptiver Kernel, IDE-Subsystem, ISA PnP, USB 2.0
- Notebook-Nutzer
 - ★ Speed Stepping, Software-Suspend-to-Disk, APM-/ACPI-Verbesserungen
- Eingebettete Systeme
 - ★ uCLinux-Integration, Headless-Konfiguration, Swap optional

- *The Wonderful World of Linux 2.6:* <http://www.kniggit.net/wwol26.html>
- *Linux Kernel 2.6: New Features – I / II*
<http://www.axian.com/docs/linuxkernel.pdf>
<http://www.axian.com/docs/linuxkernel2.pdf>
- *Vanishing Features of the 2.6 Kernel:*
<http://linux.oreillynet.com/pub/a/linux/2002/12/12/vanishing.html>
- *Reiser4 Benchmarks:*
<http://www.namesys.com/benchmarks/v4marks.html>
- *Linux: Benchmarking Filesystems In 2.6.0-test2:*
<http://kerneltrap.org/node/view/715>
- *Interviews: Interview with a Kernel Hacker:*
<http://www.tinyminds.org/modules.php?op=modload&name=News&file=article&sid=845>

- *Kernel Traffic*: <http://kt.zork.net/>
- *x15-Webserver*: <http://www.chromium.com/x15.html>
- *Next Generation POSIX Threads*:
<http://www-124.ibm.com/developerworks/oss/pthreads/>
- *InterMezzo Filesystem Homepage*: <http://www.inter-mezzo.org/>