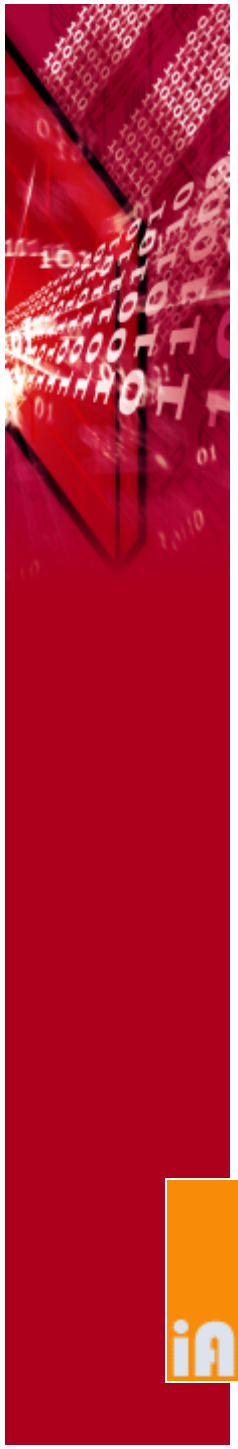
Was ist neu oder besser im Kernel 2.6?

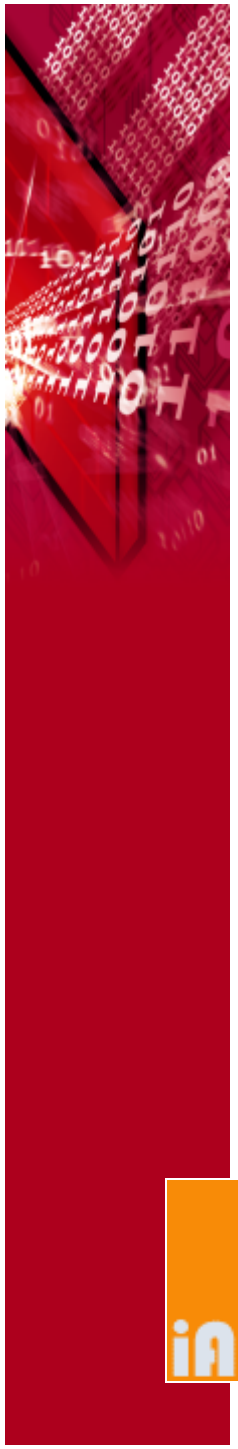
Wilhelm Dolle
Director Information Technology
interActive Systems GmbH

Chemnitzer Linux-Tag 2004
06. März 2004, Chemnitz



Agenda

- Aufgaben, Architektur und Historie des Linux-Kernels
- Übersicht der Änderungen im Kernel 2.6
- Installation
- Linux Device Model (LDM)
- Neues virtuelles Dateisystem sysfs
- Advanced Configuration and Power Interface (ACPI)
- Scheduling
- Sicherheit



Der Linux-Kernel

- **stellt Anwendungen eine einheitliche Schnittstelle (API) unabhängig von der Hardware-Architektur zur Verfügung (hardwareabstrahierende Schicht)**
- **übernimmt das System beim Boot-Prozess wenn der BIOS-Boot Loader erfolgreich war und die System-Initialisierungen des BIOS abgeschlossen sind**
- **ist zum Beispiel zuständig für**
  - Prozessverwaltung, Multitasking, Lastverteilung
  - Erzwingen der Sicherheit
  - Eingabe/Ausgabe-Operationen auf verschiedenen Geräten wie Festplatten oder Netzwerkkarten
- **fast komplett in C geschrieben**
  Ausnahme: architekturabhängigen Teile des Codes in Assembler (werden zum Beispiel beim Bootvorgang benötigt)

- **ist noch kein komplettes Betriebssystem, da ohne weitere Programme relativ nutzlos**

Kurze Historie des Linux-Kernels

Linux-0.01	17. September 1991	62 KB (als bzip2)
Linux-0.02	05. Oktober 1991	Erster „offizieller“ Kernel
Linux-0.12	05. Januar 1992	106 KB; GPL
Linux-1.0	13. März 1994	1.0 MB
Linux-1.2	07. März 1995	1.8 MB
Linux-2.0	09. Juni 1996	M: David Weinehall; 4.5 MB
Linux-2.2	26. Januar 1999	M: Alan Cox; 10 MB
Linux-2.4	04. Januar 2001	M: Marcelo Tosatti; 19 MB
Linux-2.5	22. November 2001	23 MB
Linux-2.6	18. Dezember 2003	M: Andrew Morton (Linus Torvalds); 32 MB; 5 Mio. LoC

Übersicht der Änderungen I

- **Verbesserungen für den High-End-Bereich**
  - neuer Scheduler
  - besserer Umgang mit großen Massenspeichern
  - schnellerer IP-Stack
  - besseres Virtual-Memory-Subsystem
  - verbesserter Threading-Support (Native POSIX Thread Library, NPTL)

Übersicht der Änderungen II

- **Sicherheit**
  - direkte Kernel-Unterstützung von IPSec
  - Linux Security Modules (LSM-Infrastruktur)
- **Unterstützung neuer Hardware-Plattformen**
  - m68k-Prozessoren (z.B. PDAs)
  - 64-Bit-Architekturen: AMD64, IA64, Sparc64
  - NUMA-Architekturen (Non Uniform Memory Access)

Übersicht der Änderungen III

- **Verbesserungen für Anwender**
  - neue Sound-Architektur: ALSA
  - Unterstützung für ACPI
  - Unterstützung für USB 2.0
  - Unterstützung für CAPI 2.0 (macht isdn4linux obsolet)
  - neues IDE-Subsystem
  - POSIX-ACLs (Access Control Lists)
  - Heruntertakten für Intel Speedstep und AMD PowerNow!
  - Linux Device Model (LDM)
  - Unterstützung für Hyperthreading
  - BIOS-Erweiterungen
    - Enhanced Disk Device Polling (EDD) – entnimmt dem BIOS zur Laufzeit Infos über bootfähige Geräte
    - Simple Boot Flag (SBF) - erlaubt, nach erfolgreichem Booten künftig einen umfangreichen Power-On-Self-Test zu unterbinden

Installation (modutils)

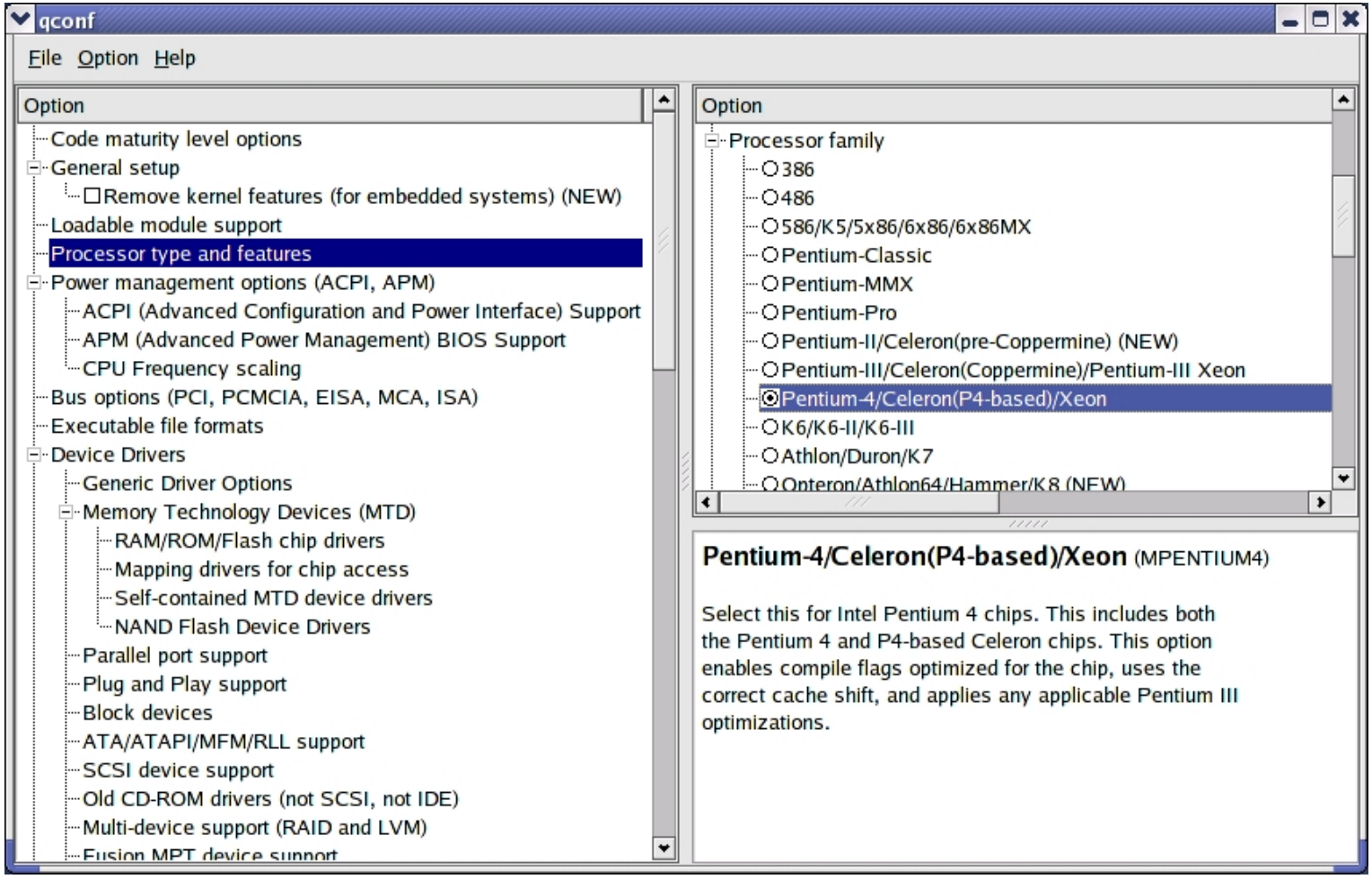
- Kernel 2.6 benötigt aktuelle modutils, da sich der Mechanismus zum Laden und Entfernen von Modulen geändert hat

- Neues insmod ruft für ältere Kernel insmod.old auf (analog: modprobe, rmmod)
- Neue Konfigurationsdatei: /etc/modprobe.conf
- Evtl. erzeugen mit:

  generate-modprobe.conf /etc/modprobe.conf

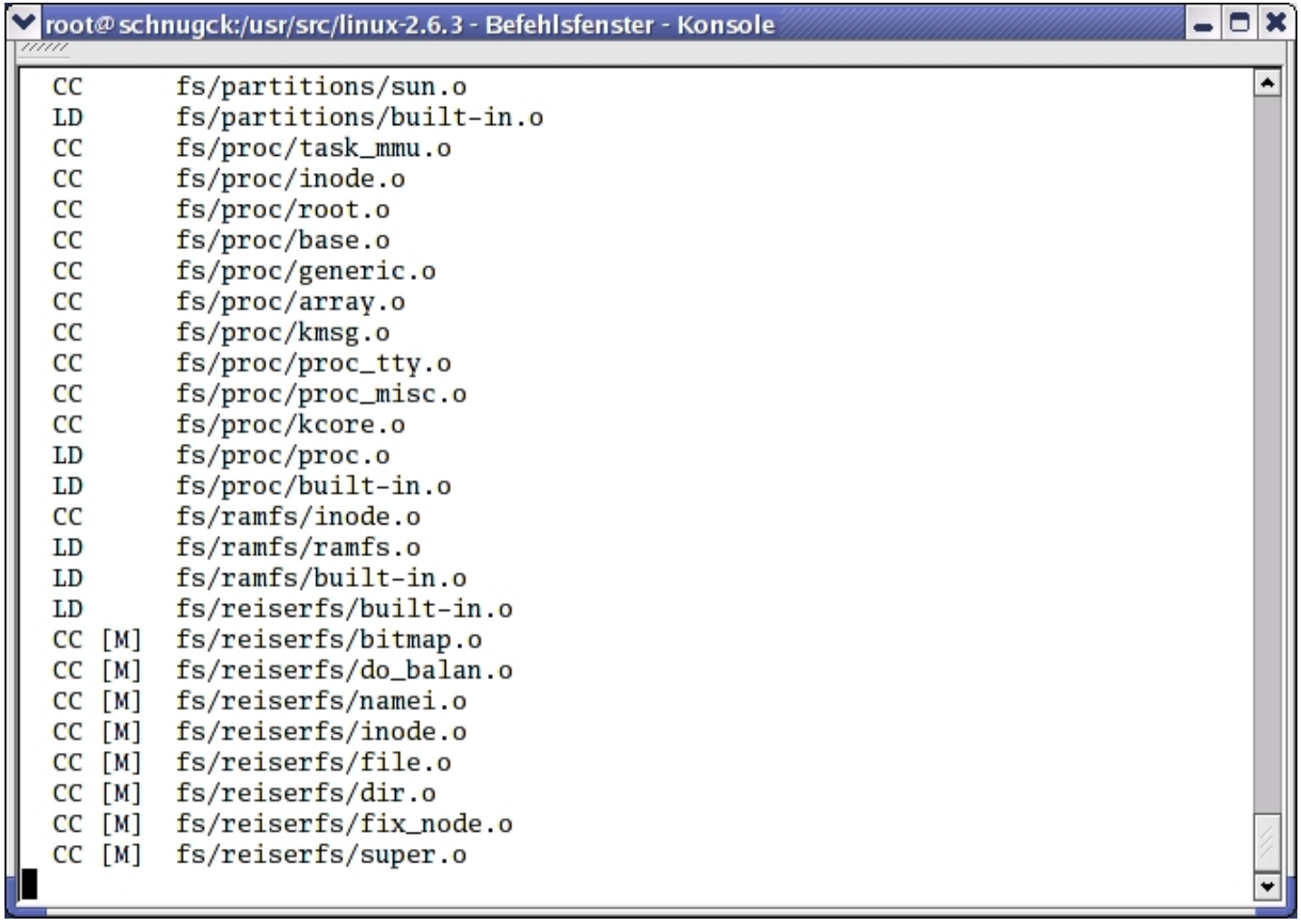
Installation (Kernelkonfiguration)

- make xconfig

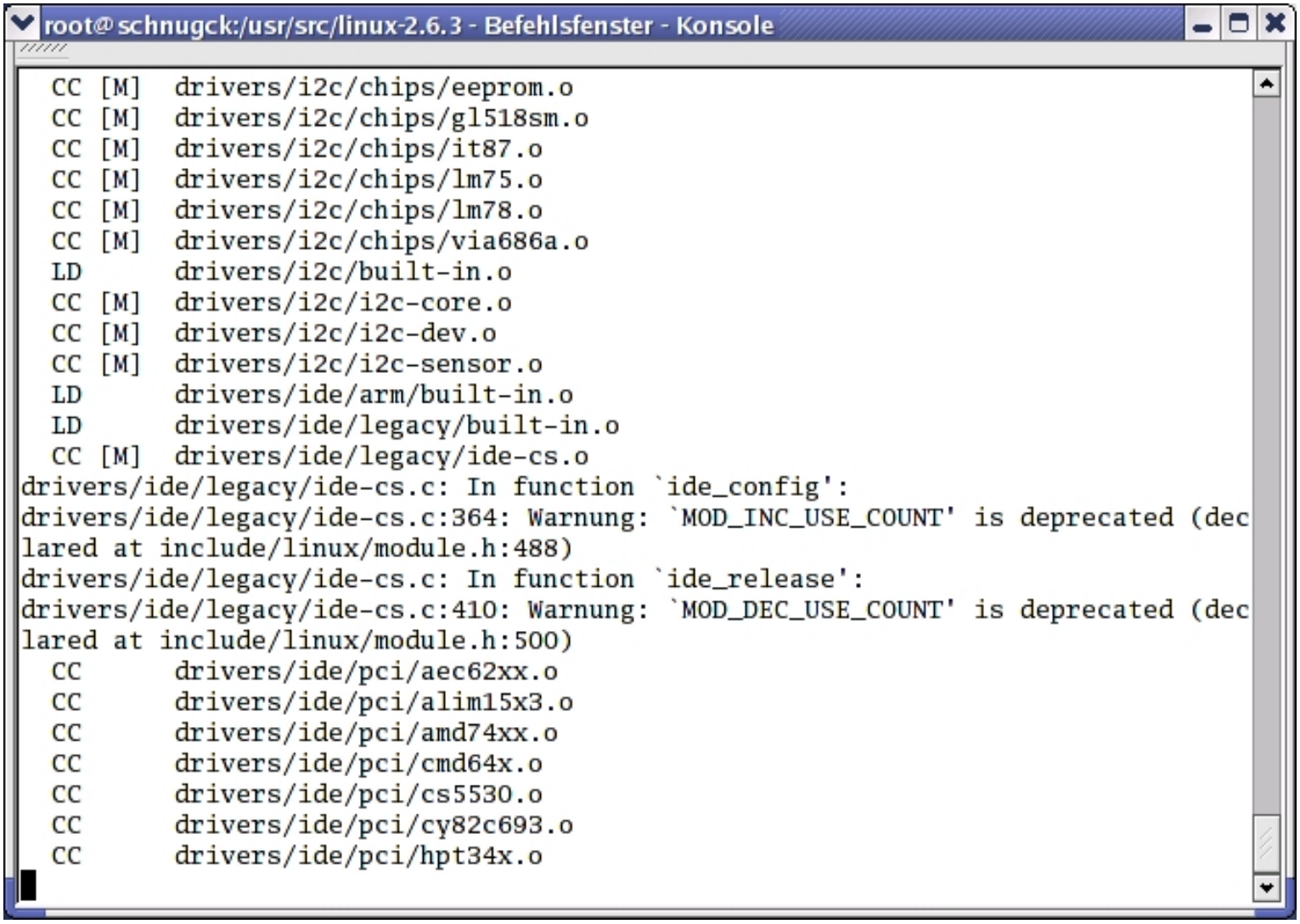


Installation (Übersetzen)

- make all



Installation (Übersetzen)



Installation (Übersetzen)

- make modules_install (make modules entfällt)

root@schnugck:/usr/src/linux-2.6.3 - Befehlsfenster - Konsole

```
  INSTALL drivers/usb/net/usbnet.ko
  INSTALL drivers/usb/serial/usbserial.ko
  INSTALL drivers/usb/storage/usb-storage.ko
  INSTALL drivers/usb/misc/uss720.ko
  INSTALL fs/vfat/vfat.ko
  INSTALL drivers/video/vga16fb.ko
  INSTALL drivers/video/vgastate.ko
  INSTALL drivers/i2c/chips/via686a.ko
  INSTALL drivers/char/agp/via-agp.ko
  INSTALL drivers/net/via-rhine.ko
  INSTALL drivers/usb/serial/visor.ko
  INSTALL drivers/i2c/chips/w83781d.ko
  INSTALL drivers/usb/input/wacom.ko
  INSTALL net/wanrouter/wanrouter.ko
  INSTALL drivers/net/wireless/wavelan_cs.ko
  INSTALL drivers/net/wireless/wavelan.ko
  INSTALL drivers/scsi/wd7000.ko
  INSTALL drivers/net/wd.ko
  INSTALL drivers/block/xd.ko
  INSTALL net/xfrm/xfrm_user.ko
  INSTALL drivers/net/pcmcia/xirc2ps_cs.ko
  INSTALL drivers/net/yellowfin.ko
  INSTALL drivers/net/wan/z85230.ko
  INSTALL drivers/atm/zatm.ko
  INSTALL lib/zlib_deflate/zlib_deflate.ko
if [ -r System.map ]; then /sbin/depmod -ae -F System.map  2.6.3; fi
[root@schnugck linux-2.6.3]#
```

Linux Device Model (LDM)

- Neues Treibermodell im Advanced Configuration and Power Interface Kernel 2.6
- Systemkomponenten (CPU, Festplatte, Soundkarte, ...) werden mit ihren Abhängigkeiten in einer hierarchischen Struktur abgebildet
- Saubere Suspend- und Resume-Sequenzen werden so möglich
- alle Treiber müssen suspend() und resume() Funktion anbieten
- Erweiterte Hotplug-Fähigkeit des Kernels – unter anderem können automatisch die benötigten Gerätedateien angelegt werden

Neues virtuelles Dateisystem sysfs

- sysfs ist die neue Schnittstelle des Treibermodells zum User-Space
- Module exportieren ihre Infos nicht mehr nach /proc sondern nach /sys
- Informationen zu angeschlossenen Geräten (z.B. Geräteklasse, Power-Status, verfügbare Funktionen) werden in einer Baumstruktur dargestellt
- Alle gerätebezogenen Daten sollen von /proc nach /sys verlegt werden (Infos über Prozesse bleiben dort)
- Virtuelles devfs-Dateisystem wird obsolet werden

Advanced Configuration and Power Interface

- 1996 ACPI Version 1 veröffentlicht
- Soll Plug&Play und Advanced Power Management (APM) ablösen
- Konfiguration der Hardware ist neben Power Management eine sehr wichtige Aufgabe von ACPI
- Sehr komplexe Spezifikationen – weder BIOS- noch Betriebssystemhersteller implementieren alle Funktionen korrekt

ACPI unter Linux 2.6 (Sleep States)

- **/sys/power/state (standby, mem, disk)**
- **Standby**
  - System eingeschaltet, CPU führt keine Befehle aus
- **Suspend-to-RAM**
  - System bis auf Hauptspeicher abgeschaltet (z.B. CPU, Festplatte, USB)
  - Aufwachen erfolgt in wenigen Sekunden
  - Funktioniert eher theoretisch (neu in 2.6, „Experimental“)
- **Suspend-to-Disk**
  - System abgeschaltet, Speicherinhalt, Register, Peripheriestatus auf Festplatte (bei Linux Swappartition)
  - Unter Kernel 2.4 nur mit Kernelpatches möglich

ACPI unter Linux 2.6 (dyn. Taktfrequenz)

- **Intel Speedstep, AMD Power Now!**
- **Zur CPU passenden „cpufreq“-Treiber in der Kernel-Konfiguration auswählen**

- **/sys/devices/system/cpu/cpu0/cpufreq**
  - cpuinfo_max_freq (maximaler Prozessortakt)
  - cpuinfo_min_freq (minimaler Prozessortakt)
  - scaling_max_freq (obere Grenze)
  - scaling_min_freq (untere Grenze)

  - scaling_governor
    (erlaubt: powersave, userspace, performance)

Linux Scalability Effort (LSE)

- **Im Fokus: Performanceverbesserungen auf**
  - „großen Systemen“ – viele Prozessoren, viele Festplatten mit großer Kapazität
  - „stark belasteten Systemen“ – viele Prozesse, viele Benutzer, hohe Auslastung des Hauptspeichers und der Kommunikationsbandbreite

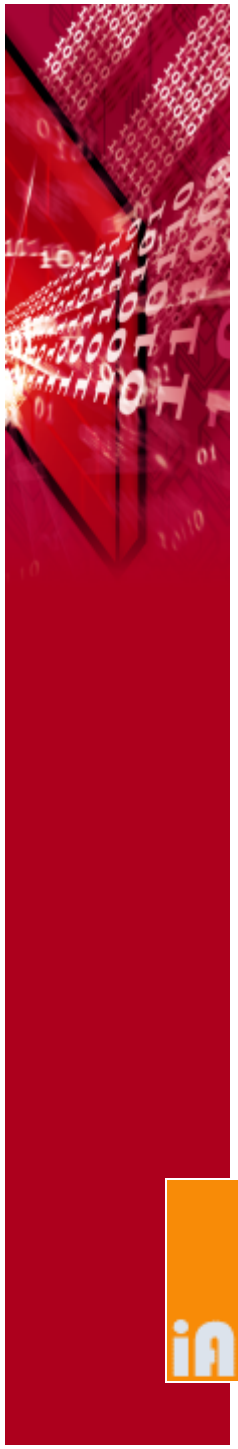
- **Ergebnisse für 2.6er Kernel**
  - Neuer O(1) Scheduler
  - Neuer Block Device Layer
  - Asynchrones I/O
  - NUMA-Support
  - Reduzierung globaler Spin-Locks, Verkleinerung der kritischen Regionen

Scheduling allgemein

- **Was ist Scheduling?**
  - Zeitliche Ordnung für Ereignisse
- **Wofür brauchen wir Scheduling?**
  - Optimierte Ausnutzung von Ressourcen
  - Optimierte Reaktionszeiten
  - Maximierter Durchsatz, minimierte Ausführungszeit
- **Arten des Scheduling**
  - Befehlsscheduling
  - I/O Scheduling
  - CPU Scheduling (Welche ***ablauffähigen*** Prozesse dürfen die CPU benutzen? - Berechnet Prioritäten und Zeitscheiben)

Scheduling-Algorithmen

- **First come, first serve (FCFS)**
  - z.B. bei Linux-Echtzeit: SCHED_FIFO (Prozess läuft bis er fertig ist oder von sich aus die Kontrolle abgibt)
- **Shortest job first**
  - Unterbrechend (PSJF)
  - Nicht unterbrechend (SJF)
- **Priority scheduling**
  - Unterbrechend
  - Nicht unterbrechend
- **Round-robin scheduling (RR)**
  - z.B bei Linux-Echtzeit: SCHED_RR (Prozess läuft mit Echtzeitpriorität, hat aber definierte Zeitscheiben)



Scheduling mit mehr als einer CPU

- Auf welcher CPU soll der Prozess laufen?
- Gleichmäßige Belastung der Prozessoren
- Verwaltungsaufwand wenn ein Prozess von einer auf die andere CPU „wechselt“

Scheduling im Kernel 2.4

- **Liste (ungeordnet) mit allen lauffähigen Prozessen**
- **Wertung in goodness() basierend auf verbleibender Zeit und nice-Level**
- **Realtime-Prozesse**
  - goodness=1000+rt_priority
- **Time-Sharing**
  - quantum>0: goodness=quantum+priority
  - quantum=0: goodness=0
- **Probleme**
  - Neuberechnung wenn alle Zeitscheiben verbraucht sind – Durchlauf der kompletten Liste – Komplexität: O(n)
  - Schlecht skalierbar wegen globaler Prozessliste

Neuer O(1) Scheduler im Kernel 2.6

- **Vorteile**
  - O(1) Algorithmen
  - CPU Affinität
  - Verbesserte Interaktivität und Skalierbarkeit
  - Unterbrechbarer Kernel (preemptiver Kernel)
  - Ermöglicht „weiche“ Echtzeit

Funktionsweise des O(1) Schedulers

- **Scheduler benutzt pro CPU eine „Run-Queue“ mit zwei Feldern**
  - „active array“: Prozesse die noch Prozessorzeit übrig haben
  - „expired array“: Prozesse die ihre Prozessorzeit verbraucht haben
- **Prozesse die Ihre Zeit verbraucht haben werden vom „active array“ ins „expired array“ verschoben**
- **Jedes Feld enthält 140 Einträge (in einer Bitmap) die entweder 0 sind, oder auf Listen mit Prozessen zeigen**
  - 0 bis 100: Echtzeitprozesse
  - 101 bis 140: ***nice***-Werte von -19 bis 20

Berechnung der Prioritäten

- Statische Prozesspriorität (static_prio) wird vom nice-Wert (-19 bis 20) bestimmt und zunächst in den Bereich von 101 bis 140 gemappt (z.B. Priorität 120 für nice-Wert 0)
- Dynamische (effektive) Prozesspriorität (prio) wird anhand der Interaktivität berechnet
- Grad der Interaktivität ist abhängig davon, ob der Prozess eher I/O-lastig (hohe Interaktivität) oder CPU-lastig (niedrige Interaktivität) ist
- Kernel protokolliert dazu für jeden Prozess die Zeit die der Prozess schläft (und z.B. auf I/O wartet) oder die CPU benutzt und berechnet daraus die „Sleep Average“ (sleep_avg) die zwischen 0 und 10 (MAX_SLEEP_AVG) liegt.
- Prozesse mit hoher Interaktivität (hohe sleep_avg) können maximal 5 Punkte Bonus, solche mit niedriger Interaktivität maximal 5 Punkte Malus bekommen

Berechnung der Zeitscheiben

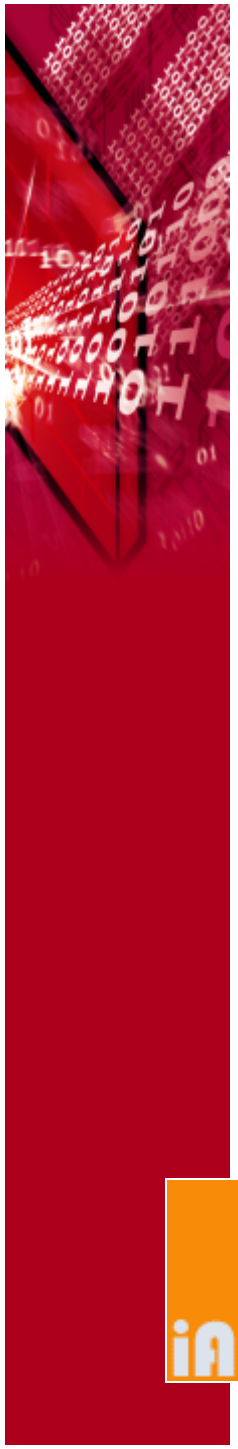
- Länge der Zeitscheiben liegt, abhängig von der Priorität, zwischen 10 und 200 Millisekunden – Prozesse mit „nice“-Level 0 erhalten die Standard-Zeitscheibe 100 ms

- Ist die Zeitscheibe eines Prozesses aufgebraucht, wird sie direkt neu berechnet und der Prozess ins „expired array“ verschoben

Vorteile des neuen Schedulers

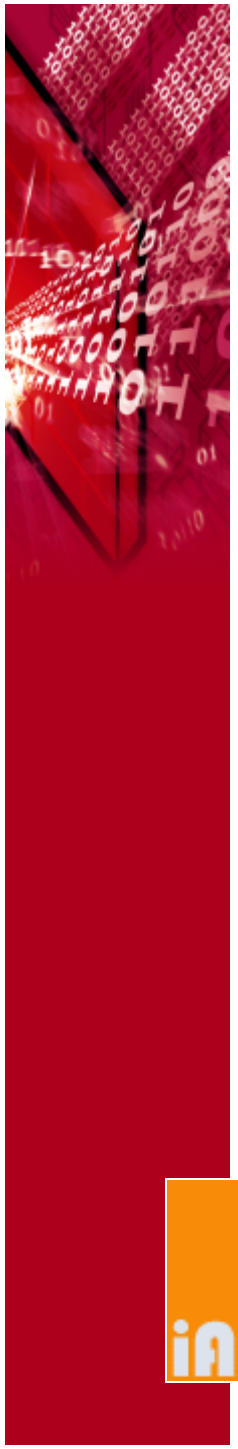
- Bei der Suche nach dem Prozess mit der höchsten Priorität wird eine Bitmap von konstanter Größe statt einer Liste mit dynamischer Größe durchsucht

- Zeiger der Run-Queue wird nur vom „active array“ auf das „expired array“ getauscht, statt die Zeitscheiben aller Prozesse neu zu berechnen



Sicherheit: Linux Security Modules (LSM)

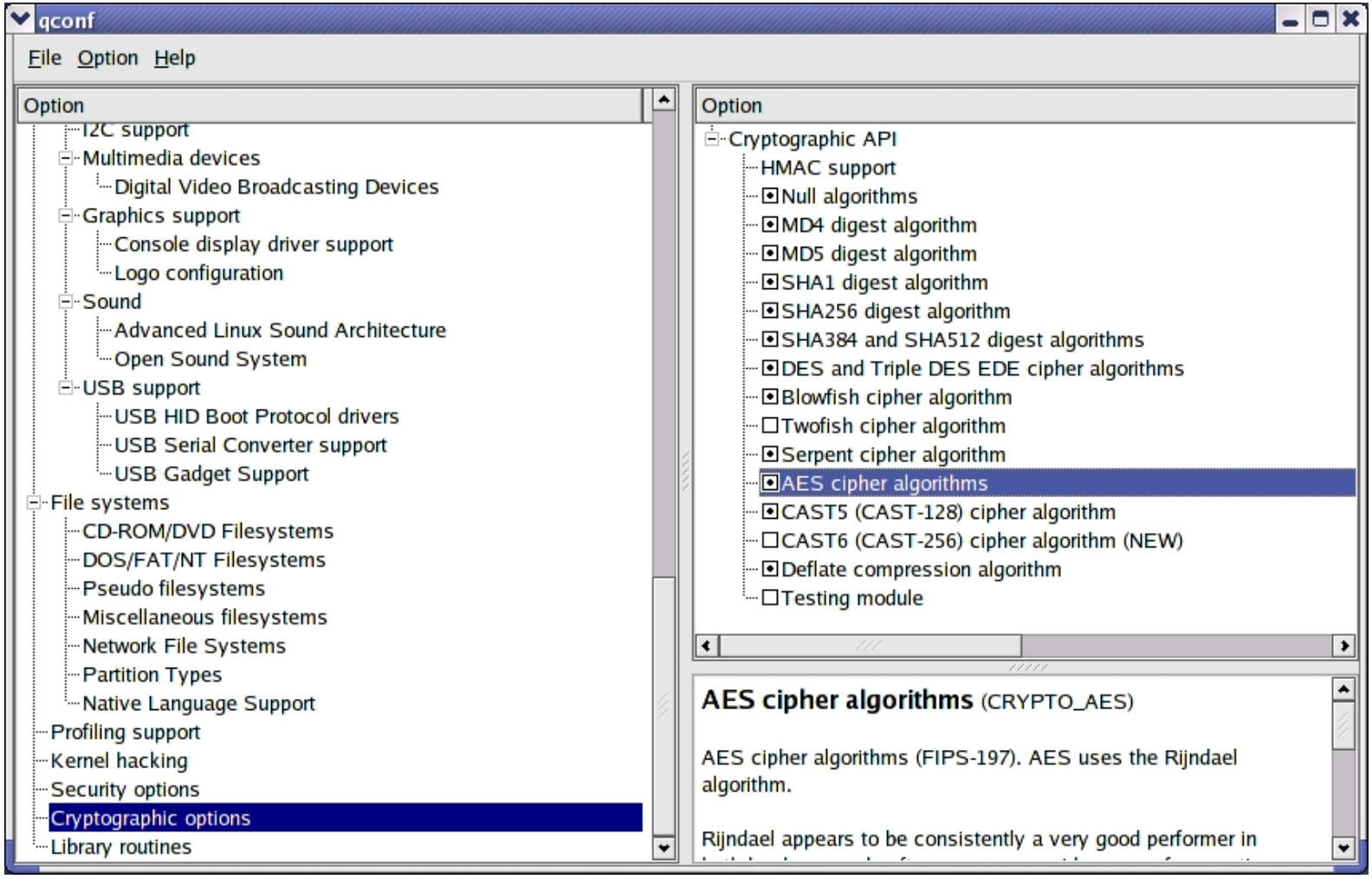
- **März 2001: Security Enhanced Linux (SE-Linux) der NAI-Labs und der NSA**
- **Seit 2.5 als Patch, seit 2.6 als Teil der LSM offizieller Bestandteil des Kernels (muss statisch eingebunden werden)**
- **Subjekte greifen über Object-Manager und Security-Server auf Objekte zu (Zugriffsmatrix)**
- **Infrastruktur zur Umsetzung von Sicherheitskonzepten**
  - Mandatory Access Control (MAC)
  - Capabilities
  - Role-based Access Control (RBAC)
  - Domain and Type Enforcement (DTE)
- **Klassische Alternativen zur „Kernelhärtung“**
  - LIDS
  - OpenWall



Sicherheit: IPSec

- Seit Kernel 2.2 VPNs mit FreeS/WAN Kernelpatches möglich
- FreeS/WAN wurde nicht in den Kernel aufgenommen (FreeS/WAN wird demnächst eingestellt)
- Mit Kernelversion 2.6 native IPSec Unterstützung (orientiert sich an der BSD-Implementierung)
- Baut auf Crypto-API auf

Sicherheit: Crypto-API



Fazit

- „großer Wurf“
- Wegfall vieler Einschränkungen
- Sehr gute Skalierbarkeit
- Design (z.B. Gerätemodell, Block Device Layer) deutlich verbessert
- Viele sinnvolle Features integriert (Wegfall einer Menge von Patches)

Fragen?

Vielen Dank für die Aufmerksamkeit!

Wilhelm Dolle
Director Information Technology

iAS interActive Systems
Dieffenbachstrasse 33c
D-10967 Berlin

phone +49-(0)30-69004-100
fax +49-(0)30-69004-101
mail wilhelm.dolle@interActive-Systems.de
web http://www.interActive-Systems.de