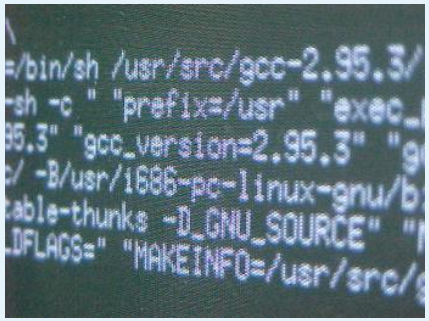
Linux Kernel 2.6

- und nun?

Michael Prokop / Grazer LinuxTage 2004

michael@linuxtage.at / www.michael-prokop.at / www.grml.org



Inhalt

Einleitung
- Inhalt
- Motivation für diesen Vortrag

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- Motivation für diesen Vortrag
- Geschichte des Kernels
- ...
- Q&A

Motivation für diesen Vortrag

Einleitung
- Inhalt
- Motivation für diesen Vortrag

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- Motivation? Viele Anfragen wie, was und warum
- seit Anfang Jänner entstanden
- Erwartungen:
  - Einstieg für Kernelnewbies
  - Abriss der Geschichte
  - Thomas und Markus: Kernelinternas

Geschichte - Der Anfang

Einleitung

Geschichte
- Geschichte - Der Anfang
- Geschichte - 1991-1992
- Geschichte - 1993-1994
- Geschichte - Versionsnummern
- Geschichte - 1994-1996
- Geschichte - 1997-1999
- Geschichte - 1999-2000
- Geschichte - 2.4.0-Release 1/5
- Geschichte - 2.4.0-Release 2/5
- Geschichte - 2.4.0-Release 3/5
- Geschichte - 2.4.0-Release 4/5
- Geschichte - 2.4.0-Release 5/5
- Geschichte - 2001-2003
- Geschichte - 2.6-Release
- Geschichte - Aktuelles
- Geschichte - Quellen

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- 1.1.1991: Linus Benedict Torvalds (aufbauend auf Minix von Andrew Tanenbaum): unixartiges Betriebssystem für AT-386-Computer
- Linus' Posting 'Hello everybody out there using minix'-Posting in comp.os.minix <1991Aug25.205708.9541@klaava.Helsinki.FI>
- Anfang 1992: legendärer Thread - "Linux is obsolete" http://www.dina.dk/ abraham/Linus_vs_Tanenbaum.html

Geschichte - 1991-1992

Einleitung

Geschichte
- Geschichte - Der Anfang
- Geschichte - 1991-1992
- Geschichte - 1993-1994
- Geschichte - Versionsnummern
- Geschichte - 1994-1996
- Geschichte - 1997-1999
- Geschichte - 1999-2000
- Geschichte - 2.4.0-Release 1/5
- Geschichte - 2.4.0-Release 2/5
- Geschichte - 2.4.0-Release 3/5
- Geschichte - 2.4.0-Release 4/5
- Geschichte - 2.4.0-Release 5/5
- Geschichte - 2001-2003
- Geschichte - 2.6-Release
- Geschichte - Aktuelles
- Geschichte - Quellen

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- 17.09.1991 (17:29:55): Kernelversion 0.01 [62 KiB]: ”Wirzenius wrote this portably, Torvalds fucked it up :-)” – kernel/vsprintf.c
  Start des Samba-Projektes von Andrew Tridgell (Vernetzung von DEC-/SUN-Computern)
- 16.01.1992 (6:39:10): Kernelversion 0.12; Linus verteilt Code den er unter die GPL stellt per anonymous FTP im Internet; Entstehung von alt.os.linux
- 08.03.1992 (12:04:59): Kernelversion 0.95
- April/Mai 1992: Kernelversion 0.96 - funktionierendes X Window System
  21. April: XFree86 gegründet

Geschichte - 1993-1994

Einleitung

Geschichte
- Geschichte - Der Anfang
- Geschichte - 1991-1992
- Geschichte - 1993-1994
- Geschichte - Versionsnummern
- Geschichte - 1994-1996
- Geschichte - 1997-1999
- Geschichte - 1999-2000
- Geschichte - 2.4.0-Release 1/5
- Geschichte - 2.4.0-Release 2/5
- Geschichte - 2.4.0-Release 3/5
- Geschichte - 2.4.0-Release 4/5
- Geschichte - 2.4.0-Release 5/5
- Geschichte - 2001-2003
- Geschichte - 2.6-Release
- Geschichte - Aktuelles
- Geschichte - Quellen

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- 1993: >100 Programmierer beteiligt; GNU-Umgebung der Free Software Foundation (FSF); viele, viele Subversionen bis zur Version 1.0;
  Dezember: Announcement von ”Netbios for Unix” (später Samba); 30. April: CERN gibt WWW frei; 26. Juli: LaTeX 2e angekündigt; 16. August: Debian Projekt gegründet
- 13.03.1994 (22:38:57): Kernelversion 1.0 [993 KiB]: netzwerkfähig(!)

Geschichte - Versionsnummern

Einleitung

Geschichte
- Geschichte - Der Anfang
- Geschichte - 1991-1992
- Geschichte - 1993-1994
- Geschichte - Versionsnummer
- Geschichte - 1994-1996
- Geschichte - 1997-1999
- Geschichte - 1999-2000
- Geschichte - 2.4.0-Release 1/5
- Geschichte - 2.4.0-Release 2/5
- Geschichte - 2.4.0-Release 3/5
- Geschichte - 2.4.0-Release 4/5
- Geschichte - 2.4.0-Release 5/5
- Geschichte - 2001-2003
- Geschichte - 2.6-Release
- Geschichte - Aktuelles
- Geschichte - Quellen

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- Kernelversionsnummern:
  - 1. Nummer = ”epochale Version” [Grundlegendes]
  - 2. Nummer = ”wichtige Neuerungen” [Schnittstellen bleiben (idR) gleich]
    - gerade Nummer: stabiler Kernel
    - ungerade Nummer: Entwicklerversion
  - 3. Nummer = Patchlevel [ausgemerzte Fehler]

Geschichte - 1994-1996

Einleitung

Geschichte
- Geschichte - Der Anfang
- Geschichte - 1991-1992
- Geschichte - 1993-1994
- Geschichte - Versionsnummern
- Geschichte - 1994-1996
- Geschichte - 1997-1999
- Geschichte - 1999-2000
- Geschichte - 2.4.0-Release 1/5
- Geschichte - 2.4.0-Release 2/5
- Geschichte - 2.4.0-Release 3/5
- Geschichte - 2.4.0-Release 4/5
- Geschichte - 2.4.0-Release 5/5
- Geschichte - 2001-2003
- Geschichte - 2.6-Release
- Geschichte - Aktuelles
- Geschichte - Quellen

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- 1994: 4. Juni: LaTeX 2e freigegeben;
  15.12.: Netscape Navigator 1.0
- 1995: 7. März: Kernelversion 1.2; Support der Plattformen Intel (i386), Digital (DEC) und Sun Sparc; 21. November: 1. Version von The Gimp; 1. Dezember: Apache 1.0
- 09.06.1996 (10:48:07): Kernelversion 2.0 [4.5 MiB] ”Fuck me gently with a chainsaw...” - arch/sparc/kernel/ptrace.c
- 1996: kommerzielle Interessen dank der Möglichkeit, mehrere Prozessoren anzusteuern wachsen; 16. Oktober: Start des KDE-Projektes (http://www.kde.org/announcements/announcement.php)

Geschichte - 1997-1999

Einleitung

Geschichte
- Geschichte - Der Anfang
- Geschichte - 1991-1992
- Geschichte - 1993-1994
- Geschichte - Versionsnummern
- Geschichte - 1994-1996
- Geschichte - 1997-1999
- Geschichte - 1999-2000
- Geschichte - 2.4.0-Release 1/5
- Geschichte - 2.4.0-Release 2/5
- Geschichte - 2.4.0-Release 3/5
- Geschichte - 2.4.0-Release 4/5
- Geschichte - 2.4.0-Release 5/5
- Geschichte - 2001-2003
- Geschichte - 2.6-Release
- Geschichte - Aktuelles
- Geschichte - Quellen

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- 1997: Netscapes Webbrowser, Office-Anwendung von Applixware und Datenbank Adabas von Software AG verfügbar; Start des GNOME-Projektes / 28. Jänner Gtk+ 1.0 veröffentlicht; August/September: KDE 1.0
- 1998: IBM, Compaq, Informix, Oracle,... entdecken Linux als Plattform; Netscape gibt am 31. März Quellcode vom Browser frei -> Mozilla-Projekt; 12. Juli: KDE 1.0; 17. Juli: Ankündigung des Linux Ports von Oracle
- 26.01.1999 (00:06:27): Kernelversion 2.2.0 [11 MiB]: verbesserter SMP-Support, überarbeiteter Netzwerkcode (Details siehe http://www.tux.org/lkml/#s10)

Geschichte - 1999-2000

Einleitung

Geschichte
- Geschichte - Der Anfang
- Geschichte - 1991-1992
- Geschichte - 1993-1994
- Geschichte - Versionsnummern
- Geschichte - 1994-1996
- Geschichte - 1997-1999
- Geschichte - 1999-2000
- Geschichte - 2.4.0-Release 1/5
- Geschichte - 2.4.0-Release 2/5
- Geschichte - 2.4.0-Release 3/5
- Geschichte - 2.4.0-Release 4/5
- Geschichte - 2.4.0-Release 5/5
- Geschichte - 2001-2003
- Geschichte - 2.6-Release
- Geschichte - Aktuelles
- Geschichte - Quellen

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- 1999: Open Sound System; 14. Jänner: Samba in Version 2.0; 17. Februar: Portierung von Domino Notes wird angekündigt; IBM propagiert seine Linux-Strategie; 3. März: GNOME 1.0; Linus sagt Kernel 2.4 für Weihnachten 1999 voraus.
- 2000: XFree86 4.0; IBM kündigt für 2001 Investitionen in Linux in der Höhe von 1 Milliarde Dollar an; Sun: StarOffice: LGPL -> OpenOffice; 15. August: Debian 2.2; 4. September: Trolltech - GPL Version von Qt; 23. Oktober 2000: KDE 2.0; 25. Dezember: The Gimp 1.2
- 11.12.2000 (00:49:45): Kernelversion 2.2.18 (14 MiB). Support für: USB, AGPgart (Texturen im Hauptspeicher), Kernelmodule der XFree86-4.0 DRI

Geschichte - 2.4.0-Release 1/5

Einleitung

Geschichte
- Geschichte - Der Anfang
- Geschichte - 1991-1992
- Geschichte - 1993-1994
- Geschichte - Versionsnummern
- Geschichte - 1994-1996
- Geschichte - 1997-1999
- Geschichte - 1999-2000
- Geschichte - 2.4.0-Release 1/
- Geschichte - 2.4.0-Release 2/5
- Geschichte - 2.4.0-Release 3/5
- Geschichte - 2.4.0-Release 4/5
- Geschichte - 2.4.0-Release 5/5
- Geschichte - 2001-2003
- Geschichte - 2.6-Release
- Geschichte - Aktuelles
- Geschichte - Quellen

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- 31.12.2000 (19:44:02): 2.4.0-Pre-release
- 04.01.2001 (23:25:55): Kernelversion 2.4.0 (19 MiB):
  - zwölf Plattformen: alpha, arm, i386, ia64, m68k, mips, mips64, ppc, s390, sh, sparc, sparc64 (Probleme bei PowerPC, Mips64 in Entwicklung)
  - Pentium-4-Familie sowie MMX- und MMX2-Befehlssätze
  - IA64 (Intel Itanium), Mips64 (64 Bit Mips) und SH (Hitachi-SH3/SH4-Prozessoren [Dreamcast-Spielekonsole]
  - bis zu 64 Prozessoren

Geschichte - 2.4.0-Release 2/5

Einleitung

Geschichte
- Geschichte - Der Anfang
- Geschichte - 1991-1992
- Geschichte - 1993-1994
- Geschichte - Versionsnummern
- Geschichte - 1994-1996
- Geschichte - 1997-1999
- Geschichte - 1999-2000
- Geschichte - 2.4.0-Release 1/5
- Geschichte - 2.4.0-Release 2/
- Geschichte - 2.4.0-Release 3/5
- Geschichte - 2.4.0-Release 4/5
- Geschichte - 2.4.0-Release 5/5
- Geschichte - 2001-2003
- Geschichte - 2.6-Release
- Geschichte - Aktuelles
- Geschichte - Quellen

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- Fortsetzung 2.4.0...
  - modernerem IO-APIC (bisher prinzipiell auf dem vom IBM AT übernommenen Konzept mit kaskadierten 8259 Interrupt-Controllern) auch auf Ein-Prozessor-Systemen und mehrere IO-APICs pro System
  - bis zu 64 GByte RAM auf x86-Systemen
  - Dateigrößenbeschränkung auf 2 GByte ist gefallen
  - Logical Volume Manager (ähnlich jenem von HP/UX)
  - Datentypen für Benutzer- und Gruppen-IDs nicht mehr wie bisher auf 16 Bit beschränkt sondern jetzt 32 Bit (> 4 Milliarden Benutzer auf Linux-System ;-))

Geschichte - 2.4.0-Release 3/5

Einleitung

Geschichte
- Geschichte - Der Anfang
- Geschichte - 1991-1992
- Geschichte - 1993-1994
- Geschichte - Versionsnummern
- Geschichte - 1994-1996
- Geschichte - 1997-1999
- Geschichte - 1999-2000
- Geschichte - 2.4.0-Release 1/5
- Geschichte - 2.4.0-Release 2/5
- Geschichte - 2.4.0-Release 3/
- Geschichte - 2.4.0-Release 4/5
- Geschichte - 2.4.0-Release 5/5
- Geschichte - 2001-2003
- Geschichte - 2.6-Release
- Geschichte - Aktuelles
- Geschichte - Quellen

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- Fortsetzung 2.4.0...
  - Raw I/O Devices (schneller ohne Puffer)
  - NFS-3
  - Support von Gigabit-Ethernet-Karten
  - Verbesserung von IPv6
  - Neugestaltung des PPP-Codes
  - Neustrukturierung der PPP-Unterstützung (z.B. PPPoE-Protokoll)
  - Journaling File Systems: in Betatestphase sind IBMs JFS (AIX, OS/2 & IRIX) und XFS (Extended File System). Weiterentwicklung von Ext2 = Ext3 und ReiserFS von Hans Reiser, allerdings kein Einzug in 2.4.0

Geschichte - 2.4.0-Release 4/5

Einleitung

Geschichte
- Geschichte - Der Anfang
- Geschichte - 1991-1992
- Geschichte - 1993-1994
- Geschichte - Versionsnummern
- Geschichte - 1994-1996
- Geschichte - 1997-1999
- Geschichte - 1999-2000
- Geschichte - 2.4.0-Release 1/5
- Geschichte - 2.4.0-Release 2/5
- Geschichte - 2.4.0-Release 3/5
- Geschichte - 2.4.0-Release 4/
- Geschichte - 2.4.0-Release 5/5
- Geschichte - 2001-2003
- Geschichte - 2.6-Release
- Geschichte - Aktuelles
- Geschichte - Quellen

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- Fortsetzung 2.4.0...
  - iptables (2.0-Kernel = ipfwadm, 2.2er = ipchains): Gesamtstruktur heißt Netfilter -> filter für Paketfilterung, nat für NAT, Masquerading und Umlenkungen und mangle fürs weitere Bearbeiten von Paketen.
  - Input core: Sammlung von Treibern, die ein allgemeines Framework zur Unterstützung von Eingabegeräten bietet
  - IEEE 1394 (von Apple Firewire und bei Sony iLink genannt): Host-Controller von Texas Instruments und solche nach dem offenen Standard OHCI werden unterstützt.

Geschichte - 2.4.0-Release 5/5

Einleitung

Geschichte
- Geschichte - Der Anfang
- Geschichte - 1991-1992
- Geschichte - 1993-1994
- Geschichte - Versionsnummern
- Geschichte - 1994-1996
- Geschichte - 1997-1999
- Geschichte - 1999-2000
- Geschichte - 2.4.0-Release 1/5
- Geschichte - 2.4.0-Release 2/5
- Geschichte - 2.4.0-Release 3/5
- Geschichte - 2.4.0-Release 4/5
- Geschichte - 2.4.0-Release 5/
- Geschichte - 2001-2003
- Geschichte - 2.6-Release
- Geschichte - Aktuelles
- Geschichte - Quellen

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- Fortsetzung 2.4.0...
  - devfs versucht Problem der knappen Major- und Minor-Device-Nummern (mknod) zu beheben
  - DRI: direkte Hardwarebeschleunigung von 3D-Grafik

(siehe auch Wonderful World of Linux 2.4)

Geschichte - 2001-2003

Einleitung

Geschichte
- Geschichte - Der Anfang
- Geschichte - 1991-1992
- Geschichte - 1993-1994
- Geschichte - Versionsnummern
- Geschichte - 1994-1996
- Geschichte - 1997-1999
- Geschichte - 1999-2000
- Geschichte - 2.4.0-Release 1/5
- Geschichte - 2.4.0-Release 2/5
- Geschichte - 2.4.0-Release 3/5
- Geschichte - 2.4.0-Release 4/5
- Geschichte - 2.4.0-Release 5/5
- Geschichte - 2001-2003
- Geschichte - 2.6-Release
- Geschichte - Aktuelles
- Geschichte - Quellen

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- 2001: Linux läuft auf IBM iSeries (AS/400); Samba 2.2; 10. März: FSF Europe gegründet
- 2002: 3. April: KDE 3.0; 6. April: Apache 2.0; 1. Mai: OpenOffice 1.0; 5. Juni: Mozilla 1.0; 26. Juni: Gnome 2.0; 19. Juli: Debian 3.0 (Woody)
- 2003: Linus Torvalds wechselt von Transmeta in das Open Source Development Lab (OSDL); Linux findet zusehends Verbreitung auf Embedded Systemen; 28.05.: Münchener Stadtrat (auf Grund einer Studie) für Umstellung von 14.000 Computern von Windows auf Linux entschieden. XFree86: Version 4.3; KDE Desktop 3.1; OpenOffice: Version 1.1; Samba: Version 3.0; 10.09.: Gnome Desktop 2.4

Geschichte - 2.6-Release

Einleitung

Geschichte
- Geschichte - Der Anfang
- Geschichte - 1991-1992
- Geschichte - 1993-1994
- Geschichte - Versionsnummern
- Geschichte - 1994-1996
- Geschichte - 1997-1999
- Geschichte - 1999-2000
- Geschichte - 2.4.0-Release 1/5
- Geschichte - 2.4.0-Release 2/5
- Geschichte - 2.4.0-Release 3/5
- Geschichte - 2.4.0-Release 4/5
- Geschichte - 2.4.0-Release 5/5
- Geschichte - 2001-2003
- Geschichte - 2.6-Release
- Geschichte - Aktuelles
- Geschichte - Quellen

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- Entwicklerserie 2.5 geschlossen, in die Serie 2.6.0-test übergeführt
- 2003-12-18 03:04 UTC: Kernelversion 2.6 (32 MiB) freigegeben, Maintainer bis zum Erscheinen des 2.7er-Zweiges ist Linus, erst dann Andrew Morton

Geschichte - Aktuelles

Einleitung

Geschichte
- Geschichte - Der Anfang
- Geschichte - 1991-1992
- Geschichte - 1993-1994
- Geschichte - Versionsnummern
- Geschichte - 1994-1996
- Geschichte - 1997-1999
- Geschichte - 1999-2000
- Geschichte - 2.4.0-Release 1/5
- Geschichte - 2.4.0-Release 2/5
- Geschichte - 2.4.0-Release 3/5
- Geschichte - 2.4.0-Release 4/5
- Geschichte - 2.4.0-Release 5/5
- Geschichte - 2001-2003
- Geschichte - 2.6-Release
- Geschichte - Aktuelles
- Geschichte - Quellen

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- 03.02.2004: KDE 3.2
- 03.02.2004: Kernel 2.6.2 (”Feisty Dunnart”)
- 11.03.2004: Kernel 2.6.4 [2004-03-11 03:16 UTC]
- 24.03.2004: The Gimp 2.0
- 01.04.2004: Gnome 2.6
- 19.04.2004: KDE 3.2.2
- xx.05.2004: Kernel 2.6.6

Geschichte - Quellen

Einleitung

Geschichte
- Geschichte - Der Anfang
- Geschichte - 1991-1992
- Geschichte - 1993-1994
- Geschichte - Versionsnummern
- Geschichte - 1994-1996
- Geschichte - 1997-1999
- Geschichte - 1999-2000
- Geschichte - 2.4.0-Release 1/5
- Geschichte - 2.4.0-Release 2/5
- Geschichte - 2.4.0-Release 3/5
- Geschichte - 2.4.0-Release 4/5
- Geschichte - 2.4.0-Release 5/5
- Geschichte - 2001-2003
- Geschichte - 2.6-Release
- Geschichte - Aktuelles
- Geschichte - Quellen

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- www.linux-magazin.de
- www.selflinux.org (Revision: 1.1.2.11)
- www.memalpha.cx/Linux/Kernel/ (nur via archive.org)
- History-Webpages der Projekte
- Literaturempfehlung: Die Software Rebellen von Glyn Moody - ISBN://3478387302

Neu gegenüber 2.4 - 1/4

Einleitung

Geschichte

Änderungen
- Neu gegenüber 2.4 - 1/4
- Neu gegenüber 2.4 - 2/4
- Neu gegenüber 2.4 - 3/4
- Neu gegenüber 2.4 - 4/4
- Änderungen gegenüber 2.4 - 1/3
- Änderungen gegenüber 2.4 - 2/3
- Änderungen gegenüber 2.4 - 3/3

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- (bessere) Unterstützung (teils neuer) 64-Bit-Architekturen (AMD64, IA64, Sparc64)
- USB 2.0-Support, USB-Gadgets
- IPMI-Treiber für Serviceprozessoren moderner Serverboards
- Unterstützung von Plug-and-Play-BIOS
- ALSA als neue Sound-Architektur
- LM-Sensors fester Bestandteils des Kernels (Nutzung von sysfs statt /proc)

Neu gegenüber 2.4 - 2/4

Einleitung

Geschichte

Änderungen
- Neu gegenüber 2.4 - 1/4
- Neu gegenüber 2.4 - 2/4
- Neu gegenüber 2.4 - 3/4
- Neu gegenüber 2.4 - 4/4
- Änderungen gegenüber 2.4 - 1/3
- Änderungen gegenüber 2.4 - 2/3
- Änderungen gegenüber 2.4 - 3/3

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- Software-basiertes Suspend-to-Disk (für Systeme ohne Hibernation im Sinne der APM-Spezifikation)[via ”resume=<swap-device>”]
- sysfs: virtuelles Dateisystem das eine Userspace-Repräsentation des LDM zur Verfügung stellt. Hinzufügen via
  ”sysfs /sys sysfs defaults 0 0” in /etc/fstab und anschließendes
  ”mkdir /sys && mount /sys”

```
$ ls /sys
block  bus  cdev  class  devices  firmware  power
```

Neu gegenüber 2.4 - 3/4

Einleitung

Geschichte

Änderungen
- Neu gegenüber 2.4 - 1/4
- Neu gegenüber 2.4 - 2/4
- Neu gegenüber 2.4 - 3/4
- Neu gegenüber 2.4 - 4/4
- Änderungen gegenüber 2.4 - 1/3
- Änderungen gegenüber 2.4 - 2/3
- Änderungen gegenüber 2.4 - 3/3

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- BIOS-Erweiterungen:
  - Enhanced Disk Device: während der Laufzeit dem BIOS Informationen über bootfähige Geräte entnehmen [/sys/firmware/edd]
  - Simple Boot Flag (nach erfolgreichem Betriebssystemstart SBF setzen → beim nächsten Boot langwierigen Power-On-Self-Test unterbinden)
- LDM (Linux Device Model): neues Treibermodell:
  - Abbildung von Systemkomponenten in hierarch. Struktur (bessere Handhabung von Abhängigkeiten + Susped-/Resume-Sequenzen)
  - erweiterte Hotplug-Fähigkeiten (hotplug-executable als Kerneläquivalent zu Userspace-Tool modprobe)

Neu gegenüber 2.4 - 4/4

Einleitung

Geschichte

Änderungen
- Neu gegenüber 2.4 - 1/4
- Neu gegenüber 2.4 - 2/4
- Neu gegenüber 2.4 - 3/4
- Neu gegenüber 2.4 - 4/4
- Änderungen gegenüber 2.4 - 1/3
- Änderungen gegenüber 2.4 - 2/3
- Änderungen gegenüber 2.4 - 3/3

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- CPU Frequency Scaling: Prozessorthrottling u.a. von Mobil-Prozessoren (Intel Speedstep, AMD PowerNow!) im laufenden Betrieb z.B. via cpudyn oder cupfreqd
- neue Locking-Primitive $\rightarrow$ futex (fast user-space mutex)
- Integration des Preemption-Patches zur Verringerung der Latenzzeit
- Unterstützung der Native POSIX Thread Library (NPTL)
- direkte Kernel-Unterstützung von IPSec zur Nutzung von VPNs
- Integration der ext-ACL-Patches

Änderungen gegenüber 2.4 - 1/3

Einleitung

Geschichte

Änderungen
- Neu gegenüber 2.4 - 1/4
- Neu gegenüber 2.4 - 2/4
- Neu gegenüber 2.4 - 3/4
- Neu gegenüber 2.4 - 4/4
- Änderungen gegenüber 2.4 - 1
- Änderungen gegenüber 2.4 - 2/3
- Änderungen gegenüber 2.4 - 3/3

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

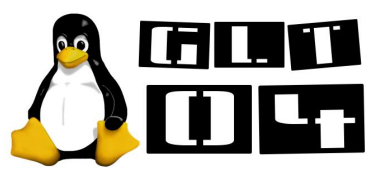
Scheduling

Sicherheit

Systemaufrufe

The End...

- Unterstützung einer Vielzahl von Mikrocontrollern (neuere m68k-Prozessoren)
- ACPI-Code
- Video4Linux -> neue API namens v4l2 (siehe auch include/linux/videodev2.h
- Neuimplementierung des IDE-Layers:
  - ide-scsi mangels Maintainer entfällt
  - CD-/DVD-Brennen via ATAPI-Interface unter Benutzung von DMA
  - Nutzung von DMA bei Rippen von Audio-CDs



Änderungen gegenüber 2.4 - 2/3

Einleitung

Geschichte

Änderungen
- Neu gegenüber 2.4 - 1/4
- Neu gegenüber 2.4 - 2/4
- Neu gegenüber 2.4 - 3/4
- Neu gegenüber 2.4 - 4/4
- Änderungen gegenüber 2.4 - 1/3
- Änderungen gegenüber 2.4 - 2
- Änderungen gegenüber 2.4 - 3/3

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

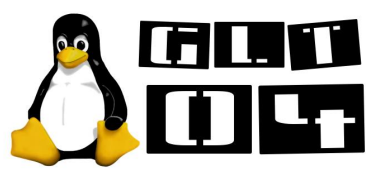
Scheduling

Sicherheit

Systemaufrufe

The End...

- isdn4linux ist obsolet - durch CAPI-2.0 ersetzt
- 32 Bit für einzelne Geräte (12 für Major- und 20 für Minor-Nummer)
- 32-Bit UID-Support
- neuer Modullader $\longrightarrow$ module-init-tools (=rückwärtskompatibel)
- LSE (Linux Scalability Effort)-Projekt: O(1)-Scheduler, bessere bessere Skalierung auf SMP-Systemen, NUMA-Support, drastische Reduzierung globaler Spin Locks,...



Änderungen gegenüber 2.4 - 3/3

Einleitung

Geschichte

Änderungen
- Neu gegenüber 2.4 - 1/4
- Neu gegenüber 2.4 - 2/4
- Neu gegenüber 2.4 - 3/4
- Neu gegenüber 2.4 - 4/4
- Änderungen gegenüber 2.4 - 1/3
- Änderungen gegenüber 2.4 - 2/3
- Änderungen gegenüber 2.4 - 3

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- erneuertes Build-System:
  - schnellerer Build-Vorgang im Vergleich zu 2.4
  - make xconfig (Qt) und make gconfig (gtk)
  - übersichtlichere Strukturierung bei make ?config
  - neue Debug-Targets: allyesconfig, allnoconfig und allmodconfig
  - weniger Output beim Build-Vorgang (via ”make V=1” oder ”set KBUILD_VERBOSE=1” umgehbar)
  - ”make dep” nicht mehr notwendig

Skalierung

Einleitung

Geschichte

Änderungen

Skalierung

● Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- ”Linux 2.6 scales O(1) in all benchmarks. Words fail me on how impressive this is. If you are using Linux 2.4 right now, switch to Linux 2.6 now!” – http://bulk.fefe.de/scalability/
- ”The v2.6 kernel ushers in a new era of support for big iron with big workloads, opening the door for Linux to handle the most demanding tasks that are currently handled by Solaris, AIX, or HP/UX.” Linux v2.6 scales the enterprise

USB (Universal Serial Bus) - 1/3

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..
- USB (Universal Serial Bus) - 1
- USB (Universal Serial Bus) - 2/3
- USB (Universal Serial Bus) - 3/3
- ALSA

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- USB ist asymmetrisch und hotplugging-fähig
  - ”High Speed” 480 Mbit/sec (60 MByte/sec)
  - ”Full Speed” 12 Mbit/sec (1.5 MByte/sec)
  - ”Low Speed” 1.5 Mbit/sec
- USB 1.1: low-speed und full speed; via ohci-hcd (in 2.4 noch als usb-ohci) [Open Host Controller Interface] oder dem älteren uhci [Universal Host Controller Interface] (Intel- und VIA-Systeme)
- USB 2.0: high speed via ehci (EHCI-Standard)
- USB 2.0-Hubs: usbcore
- Disks, CD-RW, Drives: usb-storage

USB (Universal Serial Bus) - 2/3

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

- USB (Universal Serial Bus) - 1/3
- USB (Universal Serial Bus) - 2
- USB (Universal Serial Bus) - 3/3
- ALSA

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- Entstehung von USB-Support: für einen DAB-Empfänger (Digital-Audio-Broadcast) musste eine stabile Systemumgebung geschaffen werden. Existierende USB-API: ”schlecht”, unterstützte kaum (1 Webcam ;-)) Multimedia-Geräte → Rewrite
- USB 2.0-Support? Aufwand: Treiber für neuen Host-Controller, Änderungen im Verwaltungssystem (USB 2.0 = abwärtskompatibel zu 1.1)
- Änderungen 2.4 → 2.6? Support d. vereinfachten usbcore-APIs, Interrupt-Transfers können größer ausfallen + trotzdem gequeued werden, Unterstützung einiger non-PCI-Implementierungen von OHCI, weniger Code durch hdc-Framework

USB (Universal Serial Bus) - 3/3

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

- USB (Universal Serial Bus) - 1/3
- USB (Universal Serial Bus) - 2/3
- USB (Universal Serial Bus) - 3
- ALSA

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- Backports des USB-Codes auch in 2.2 (ab 2.2.18) eingeflossen, im offiziellen Zweig seit 2.4 zu finden
- Hauptprobleme bei USB-Entwicklung? Asynchronität und SMP
- Kernel-Config: Device Drivers -> USB-Support
- Quellen:
  - Interview mit den USB-Entwicklern: Die Geburtsstunde des Linux-USB
  - Linux-USB Project
  - Linux USB Guide
  - Linux-USB Device Overview
  - $ linux-2.6.0/Documentation/usb

ALSA

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

- USB (Universal Serial Bus) - 1/3
- USB (Universal Serial Bus) - 2/3
- USB (Universal Serial Bus) - 3/3
- ALSA

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- Advanced Linux Sound Architecture - hat Open Sound System (OSS) abgelöst
- Modulare Design, SMP- und Thread-Safe-Design, Full Duplex, digitaler I/O
- Benutzerschnittstelle (alsa-lib)
- OSS-Emulation (CONFIG_SND_OSSEMUL) zwecks Abwärtskompatibilität
- Projekt-Homepage: www.alsa-project.org
- Guter Einstieg: www.gentoo.de/doc/de/alsa-guide.xml und http://alsa.opensrc.org/

Software-Requirements

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg
- Software-Requirements
- Kernelkonfi guration
- Kompilieren
- make menuconfi g
- make xconfi g
- make gconfi g
- Probleme?
- Eine Aussicht, meine Wünsche

Scheduling

Sicherheit

Systemaufrufe

The End...


```
module-init-tools   0.9.13          # depmod -V
Gnu C               2.95.3          # gcc --version
Gnu make            3.78            # make --version
binutils            2.12            # ld -v
util-linux          2.10o           # fdformat --version
procps              2.0.9           # ps --version
```

Andernfalls: unresolved symbols, QM_MODULES...

Kernelkonfiguration

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg
- Software-Requirements
- Kernelkonfi guration
- Kompilieren
- make menuconfi g
- make xconfi g
- make gconfi g
- Probleme?
- Eine Aussicht, meine Wünsche

Scheduling

Sicherheit

Systemaufrufe

The End...


```
CONFIG_VGA_CONSOLE=y        % verantwortlich wenn man bei
CONFIG_VT=y                 % Booten nichts sieht ;)
CONFIG_VT_CONSOLE=y
CONFIG_INPUT=y              % Keyboard-/Mausprobleme?
CONFIG_INPUT_KEYBOARD=y
CONFIG_KEYBOARD_ATKBD=y
CONFIG_INPUT_MOUSE=y
CONFIG_MOUSE_PS2=y
```

Kompilieren

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg
- Software-Requirements
- Kernelkonfi guration
- Kompilieren
- make menuconfi g
- make xconfi g
- make gconfi g
- Probleme?
- Eine Aussicht, meine Wünsche

Scheduling

Sicherheit

Systemaufrufe

The End...

- make help
- make oldconfig # manchmal problematisch
- make all # Debian: make-kpkg ...
- make modules_install
- cp arch/$ARCHITEKTUR/boot/bzImage /boot/vmlinuz-2.6.x
- cp System.map /boot/System.map-2.6.x
- lilo aufrufen nicht vergessen (Grub: menu.lst ;-))

make menuconfig

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

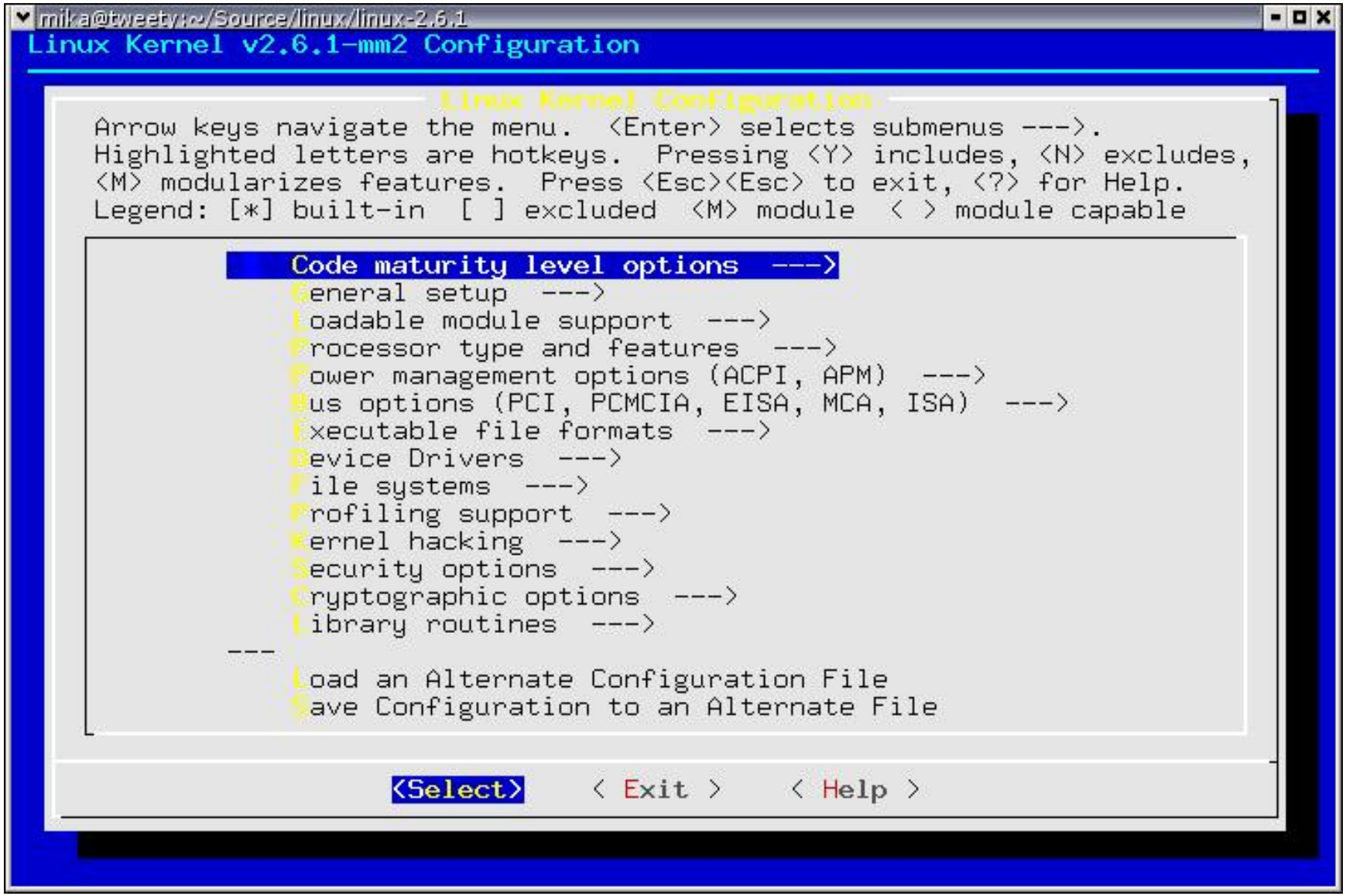
Der Umstieg
- Software-Requirements
- Kernelkonfi guration
- Kompilieren
- make menuconfi g
- make xconfi g
- make gconfi g
- Probleme?
- Eine Aussicht, meine Wünsche

Scheduling

Sicherheit

Systemaufrufe

The End...



make xconfig

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

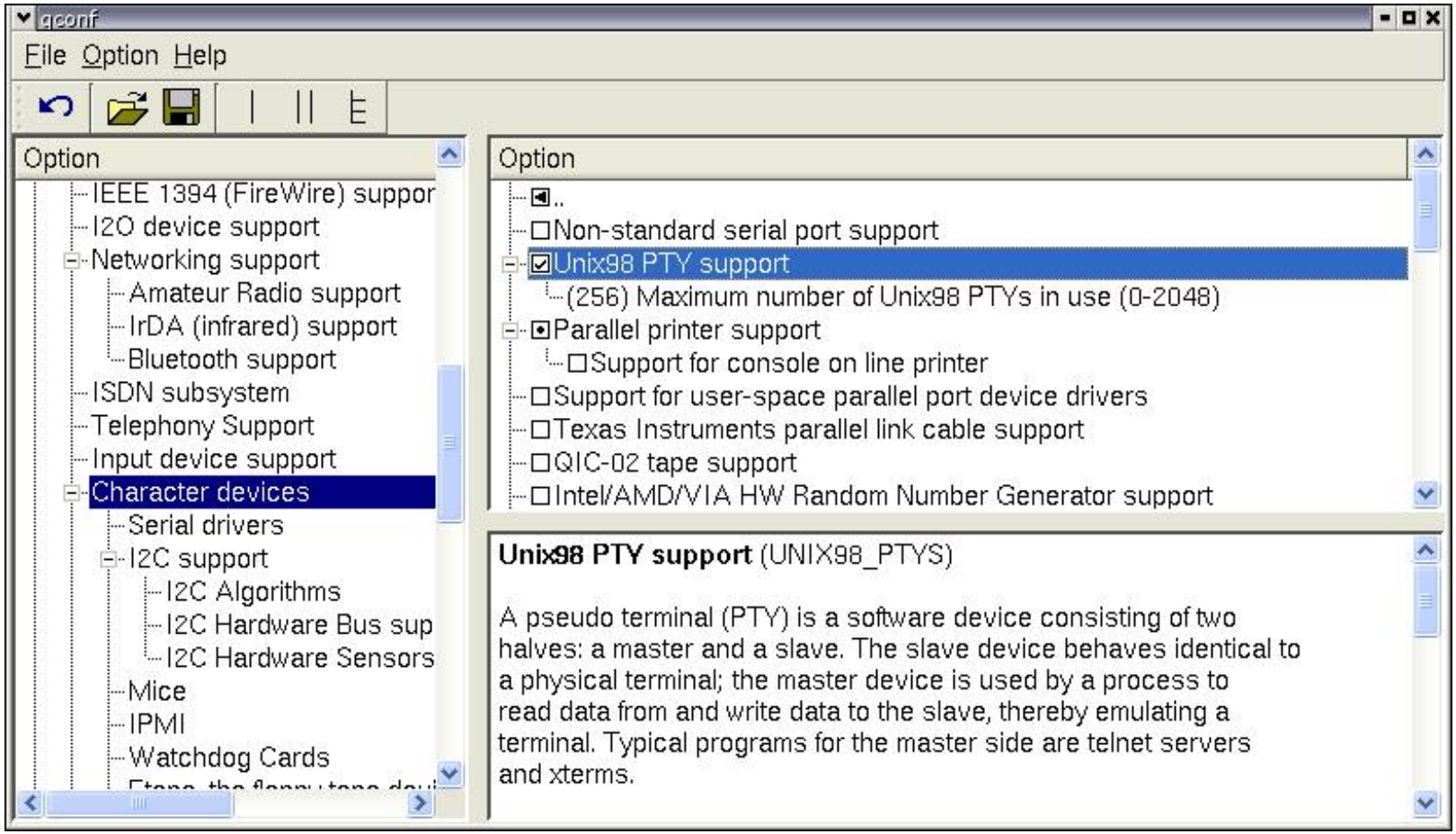
Der Umstieg
- Software-Requirements
- Kernelkonfi guration
- Kompilieren
- make menuconfi g
- make xconfi g
- make gconfi g
- Probleme?
- Eine Aussicht, meine Wünsche

Scheduling

Sicherheit

Systemaufrufe

The End...



make gconfig

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

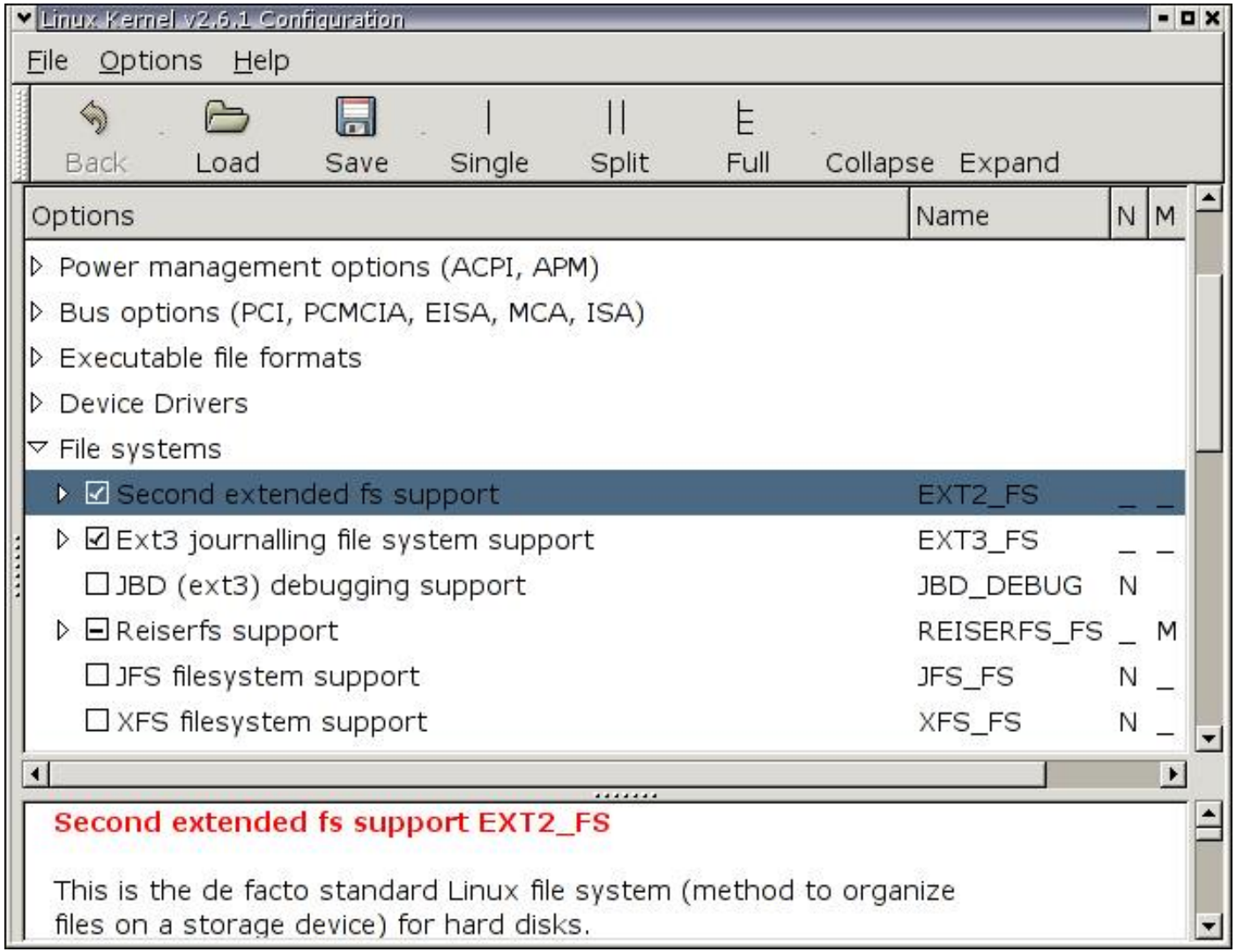
Der Umstieg
- Software-Requirements
- Kernelkonfi guration
- Kompilieren
- make menuconfi g
- make xconfi g
- make gconfi g
- Probleme?
- Eine Aussicht, meine Wünsche

Scheduling

Sicherheit

Systemaufrufe

The End...



Probleme?

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg
- Software-Requirements
- Kernelkonfi guration
- Kompilieren
- make menuconfi g
- make xconfi g
- make gconfi g
- Probleme?
- Eine Aussicht, meine Wünsche

Scheduling

Sicherheit

Systemaufrufe

The End...

- Kernelconfig in Ordnung?
- Kerneloptionen: ”noapic”, ”acpi=off”
- dt. Version von ”The post-halloween document”
- Linux: 2.6 Input Drivers FAQ

Eine Aussicht, meine Wünsche

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg
- Software-Requirements
- Kernelkonfi guration
- Kompilieren
- make menuconfi g
- make xconfi g
- make gconfi g
- Probleme?
- Eine Aussicht, meine Wünsch

Scheduling

Sicherheit

Systemaufrufe

The End...

- Audio-Bereich: Low-Latency-Treiber (<15ms Verarbeitungszeit), Entwicklungsarbeit nur in ALSA stecken (Dokumentation!)
- Security-Bereich: einheitlichere Infrastruktur, Non-Executable-Stack im Vanilla-Kernel, vservers/jails

Aufgaben

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling
- Aufgaben
- Prozessprioritäten
- Timeslices
- Prozess Descriptor
- task_struct
- Runqueues 1/2
- Runqueues 2/2
- prio_array
- Runqueue
- Scheduler 1/3
- Scheduler 2/3
- Scheduler 3/3
- Priorität/Timeslice
- Kernel Preemption
- Performance

Sicherheit

Systemaufrufe

The End...

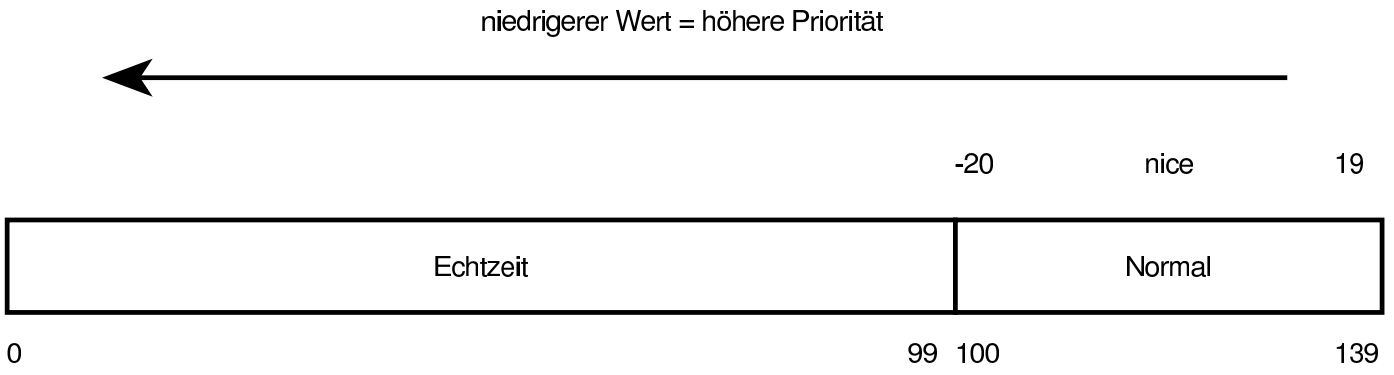
- Rechenzeit fair zwischen den Prozessen aufteilen
- Prozesse sind mit Priorität versehen
- seit Kernel 2.6 O(1) Scheduler

Prozessprioritäten

Einleitung
Geschichte
Änderungen
Skalierung
Beispiele: USB, ALSA,..
Der Umstieg
Scheduling
- Aufgaben
- Prozessprioritäten
- Timeslices
- Prozess Descriptor
- task_struct
- Runqueues 1/2
- Runqueues 2/2
- prio_array
- Runqueue
- Scheduler 1/3
- Scheduler 2/3
- Scheduler 3/3
- Priorität/Timeslice
- Kernel Preemption
- Performance

Sicherheit
Systemaufrufe
The End...

- statische Priorität (nice): -20 bis +19 (= niedrigste Priorität)
- zusätzliche Prioritäten für Echtzeitprozesse
- Prioritäten im Kernel von 0 bis 139
  - 0 bis 99: Echtzeitprozesse
  - 100 bis 139: nice Werte von -20 bis +19



Timeslices

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling
- Aufgaben
- Prozessprioritäten
- Timeslices
- Prozess Descriptor
- task_struct
- Runqueues 1/2
- Runqueues 2/2
- prio_array
- Runqueue
- Scheduler 1/3
- Scheduler 2/3
- Scheduler 3/3
- Priorität/Timeslice
- Kernel Preemption
- Performance

Sicherheit

Systemaufrufe

The End...

- Timeslice ist Zahlenwert der angibt wie lange ein Task laufen darf bis ihm der Prozessor entzogen wird (der Task „preempted“ wird)
- im Bereich von 10ms bis 200ms
- wird in Abhängigkeit von der Priorität berechnet
- Task muss seine Timelice nicht auf einmal verbrauchen sondern kann vorzeitig die Kontrolle abgeben (z.B. warten auf I/O)

Prozess Descriptor

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling
- Aufgaben
- Prozessprioritäten
- Timeslices
- Prozess Descriptor
- task_struct
- Runqueues 1/2
- Runqueues 2/2
- prio_array
- Runqueue
- Scheduler 1/3
- Scheduler 2/3
- Scheduler 3/3
- Priorität/Timeslice
- Kernel Preemption
- Performance

Sicherheit

Systemaufrufe

The End...

- Prozess Descriptor: `struct task_struct`
  - Informationen über Task, unter anderem auch für Scheduling
  - `prio` dynamische Priorität
  - `static_prio` statische Priorität
  - `sleep_avg` sagt aus wie oft und lange ein Prozess geschlafen hat
  - `policy` Scheduling Policy (Priority, FIFO, RR)
  - `time_slice` verbleibende CPU-Zeit des Tasks

task_struct

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling
- Aufgaben
- Prozessprioritäten
- Timeslices
- Prozess Descriptor
- task_struct
- Runqueues 1/2
- Runqueues 2/2
- prio_array
- Runqueue
- Scheduler 1/3
- Scheduler 2/3
- Scheduler 3/3
- Priorität/Timeslice
- Kernel Preemption
- Performance

Sicherheit

Systemaufrufe

The End...

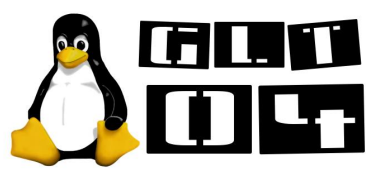

```
struct task_struct {
        /* ... */

        int prio, static_prio;
        struct list_head run_list;
        prio_array_t *array;

        unsigned long sleep_avg;
        long interactive_credit;
        unsigned long long timestamp;
        int activated;

        unsigned long policy;
        cpumask_t cpus_allowed;
        unsigned int time_slice, first_time_slice;

        /* ... */
};
```



Runqueues 1/2

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

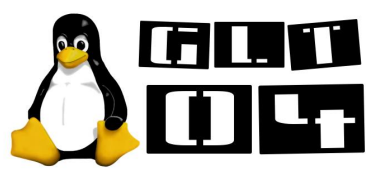
Scheduling
- Aufgaben
- Prozessprioritäten
- Timeslices
- Prozess Descriptor
- task_struct
- Runqueues 1/2
- Runqueues 2/2
- prio_array
- Runqueue
- Scheduler 1/3
- Scheduler 2/3
- Scheduler 3/3
- Priorität/Timeslice
- Kernel Preemption
- Performance

Sicherheit

Systemaufrufe

The End...

- eine Runqueue pro Prozessor
  - `active` Array: alle Prozesse mit Timeslice größer 0
  - `expired` Array: alle Prozesse die ihre Timeslice aufgebraucht haben
  - `nr_running` Anzahl der lauffähigen Prozesse
  - `curr` aktuell laufender Prozess



Runqueues 2/2

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling
- Aufgaben
- Prozessprioritäten
- Timeslices
- Prozess Descriptor
- task_struct
- Runqueues 1/2
- Runqueues 2/2
- prio_array
- Runqueue
- Scheduler 1/3
- Scheduler 2/3
- Scheduler 3/3
- Priorität/Timeslice
- Kernel Preemption
- Performance

Sicherheit

Systemaufrufe

The End...


```
struct runqueue {
        spinlock_t lock;
        unsigned long nr_running, nr_switches,
                      expired_timestamp,
                      nr_uninterruptible;
        task_t *curr, *idle;
        struct mm_struct *prev_mm;
        prio_array_t *active, *expired, arrays[2];
        int prev_cpu_load[NR_CPUS];
        /* ... */
        task_t *migration_thread;
        struct list_head migration_queue;
        atomic_t nr_iowait;
};
```

– runqueue Datenstruktur (*kernel/sched.c*)

prio_array

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling
- Aufgaben
- Prozessprioritäten
- Timeslices
- Prozess Descriptor
- task_struct
- Runqueues 1/2
- Runqueues 2/2
- prio_array
- Runqueue
- Scheduler 1/3
- Scheduler 2/3
- Scheduler 3/3
- Priorität/Timeslice
- Kernel Preemption
- Performance

Sicherheit

Systemaufrufe

The End...

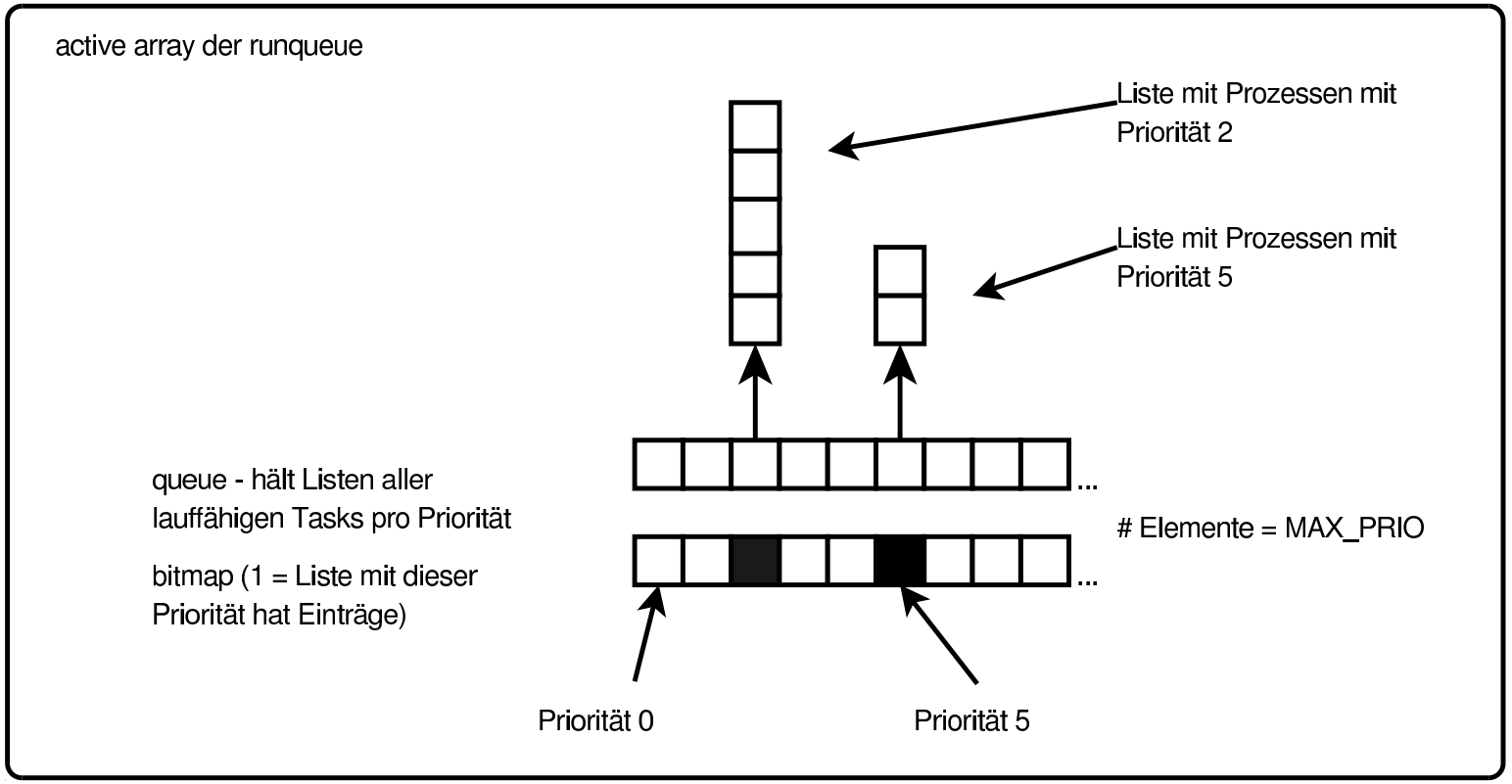

```
struct prio_array {
        int nr_active;
        unsigned long bitmap[BITMAP_SIZE];
        struct list_head queue[MAX_PRIO];
};
```

– `prio_array` Datenstruktur (*kernel/sched.c*)

Runqueue

Einleitung
Geschichte
Änderungen
Skalierung
Beispiele: USB, ALSA,..
Der Umstieg
Scheduling
- Aufgaben
- Prozessprioritäten
- Timeslices
- Prozess Descriptor
- task_struct
- Runqueues 1/2
- Runqueues 2/2
- prio_array
- Runqueue
- Scheduler 1/3
- Scheduler 2/3
- Scheduler 3/3
- Priorität/Timeslice
- Kernel Preemption
- Performance

Sicherheit
Systemaufrufe
The End...



Scheduler 1/3

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling
- Aufgaben
- Prozessprioritäten
- Timeslices
- Prozess Descriptor
- task_struct
- Runqueues 1/2
- Runqueues 2/2
- prio_array
- Runqueue
- Scheduler 1/3
- Scheduler 2/3
- Scheduler 3/3
- Priorität/Timeslice
- Kernel Preemption
- Performance

Sicherheit

Systemaufrufe

The End...

- auswählen welcher Task als nächstes laufen soll
  - erstes gesetzte Bit in bitmap finden
  - ersten Task aus Liste des zugehörigen `queue` Elements auswählen
- Kernel 2.4:
  - Prozesse nicht nach Priorität sortiert gespeichert
  - keine Trennung zwischen `active` und `expired`
  - für Auswahl des nächsten Prozesses über ganze Liste iterieren (O(n))

Scheduler 2/3

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling
- Aufgaben
- Prozessprioritäten
- Timeslices
- Prozess Descriptor
- task_struct
- Runqueues 1/2
- Runqueues 2/2
- prio_array
- Runqueue
- Scheduler 1/3
- Scheduler 2/3
- Scheduler 3/3
- Priorität/Timeslice
- Kernel Preemption
- Performance

Sicherheit

Systemaufrufe

The End...

- Aufbrauchen der Timeslice (2.4)
  - Timeslices immer dann neu berechnet wenn sie für alle Tasks verbraucht ist
  - Neuberechnung erfolgte als Schleife über alle Tasks – O(n) Zeit
- Aufbrauchen der Timeslice (2.6)
  - wenn Timeslice eines Tasks den Wert 0 erreicht, dann wird er vom `active` ins `expired` Array verschoben und gleichzeitig die Timeslice neu berechnet
  - wenn alle Timeslices 0: einfacher Pointer Tausch (`active` $\leftrightarrow$ `expired`)

Scheduler 3/3

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

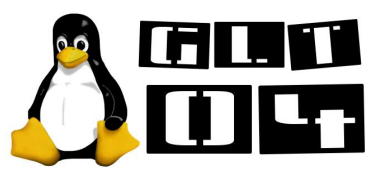
Scheduling
- Aufgaben
- Prozessprioritäten
- Timeslices
- Prozess Descriptor
- task_struct
- Runqueues 1/2
- Runqueues 2/2
- prio_array
- Runqueue
- Scheduler 1/3
- Scheduler 2/3
- Scheduler 3/3
- Priorität/Timeslice
- Kernel Preemption
- Performance

Sicherheit

Systemaufrufe

The End...

- 2 Arten von Tasks:
  - stark I/O lastig (z.B. Texteditor)
  - stark CPU lastig (z.B. Compiler)
- erfordern unterschiedliche Behandlung durch Scheduler
- Anpassung der dynamischen Priorität und der Timeslice



Priorität/Timeslice

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

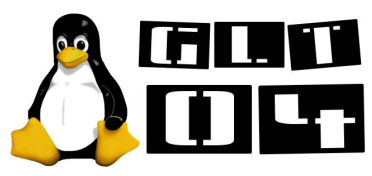
Scheduling
- Aufgaben
- Prozessprioritäten
- Timeslices
- Prozess Descriptor
- task_struct
- Runqueues 1/2
- Runqueues 2/2
- prio_array
- Runqueue
- Scheduler 1/3
- Scheduler 2/3
- Scheduler 3/3
- Priorität/Timeslice
- Kernel Preemption
- Performance

Sicherheit

Systemaufrufe

The End...

- Funktion `effective_prio` berechnet Bonus (-5 bis +5) basierend auf `sleep_avg` → dynam. Priorität
- `sleep_avg` ist Maß für Interaktivität des Tasks
  - wenn Task aufwacht wird die Dauer die er geschlafen hat zu `sleep_avg` dazu addiert
  - bei jedem Timer-Tick wird `sleep_avg` dekrementiert
  - I/O lastige Tasks haben höhere `sleep_avg` als CPU lastige Tasks
- Berechnung der Timeslice basiert auf dynamischer Priorität des Prozesses
- hoch interaktive Tasks können statt in `expired` wieder in `active` eingereiht werden



Kernel Preemption

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling
- Aufgaben
- Prozessprioritäten
- Timeslices
- Prozess Descriptor
- task_struct
- Runqueues 1/2
- Runqueues 2/2
- prio_array
- Runqueue
- Scheduler 1/3
- Scheduler 2/3
- Scheduler 3/3
- Priorität/Timeslice
- Kernel Preemption
- Performance

Sicherheit

Systemaufrufe

The End...

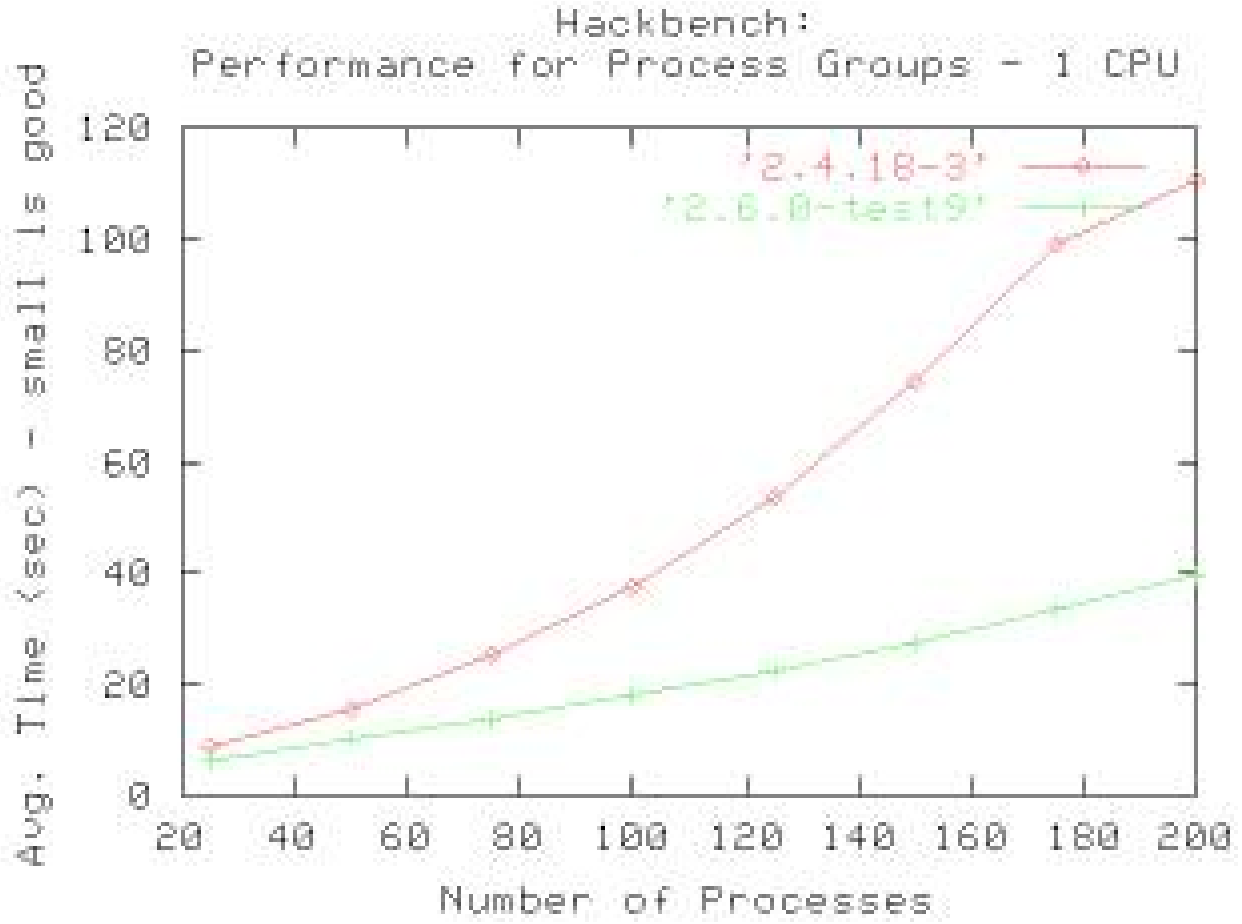
- bisher nicht möglich einem Task die CPU zu entziehen wenn er Kernel-Code ausgeführt hat
- seit 2.6 auch Preemption für Kernel-Code möglich wenn keine Locks gehalten werden
- eigener Counter für gehaltene Locks
- ist bei Rückkehr aus Interrupt Handler der `preempt_count` auf 0 oder wird dieser auf 0 dekrementiert (alle Locks freigegeben) wird der Scheduler aufgerufen
- vor 2.6 nur kooperative Preemption im Kernel durch expliziten Aufruf des Schedulers (Entwickler für sicheren Zustand verantwortlich)

Performance

Einleitung
Geschichte
Änderungen
Skalierung
Beispiele: USB, ALSA,..
Der Umstieg
Scheduling
- Aufgaben
- Prozessprioritäten
- Timeslices
- Prozess Descriptor
- task_struct
- Runqueues 1/2
- Runqueues 2/2
- prio_array
- Runqueue
- Scheduler 1/3
- Scheduler 2/3
- Scheduler 3/3
- Priorität/Timeslice
- Kernel Preemption
- Performance

Sicherheit
Systemaufrufe
The End...

- Vergleich der OSDL Labs zwischen 2.4 und 2.6



Linux Security Modules – LSM

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit
- Linux Security Modules – LSM
- Architektur
- Architektur
- Security Fields
- Hooks
- Datenstruktur
- SC `sys_nice` mit LSM Hook
- Module Stacking
- Performance und Einsatzbereiche

Systemaufrufe

The End...

- Framework für Zugriffskontrollmechanismen durch nachladbare Module
- SELinux von der NSA als Kernel Patch
- Torvalds wollte allgemeines Security-Framework
- Grundstein für die Schaffung von LSM

Architektur

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit
- Linux Security Modules – LSM
- Architektur
- Architektur
- Security Fields
- Hooks
- Datenstruktur
- SC `sys_nice` mit LSM Hook
- Module Stacking
- Performance und Einsatzbereiche

Systemaufrufe

The End...

- Zugriff auf Kernel-Objekte regeln
- darf Subjekt (Prozess) auf Objekt (Datei, Socket, ...) zugreifen
- an den entsprechenden Stellen im Kernel Hooks eingebaut
- Secrutiy Module kann Funktionen für die Hooks registrieren welche dann vom Kernel aufgerufen werden
- Hook-Funktionen liefern ja/nein Entscheidung

Architektur

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

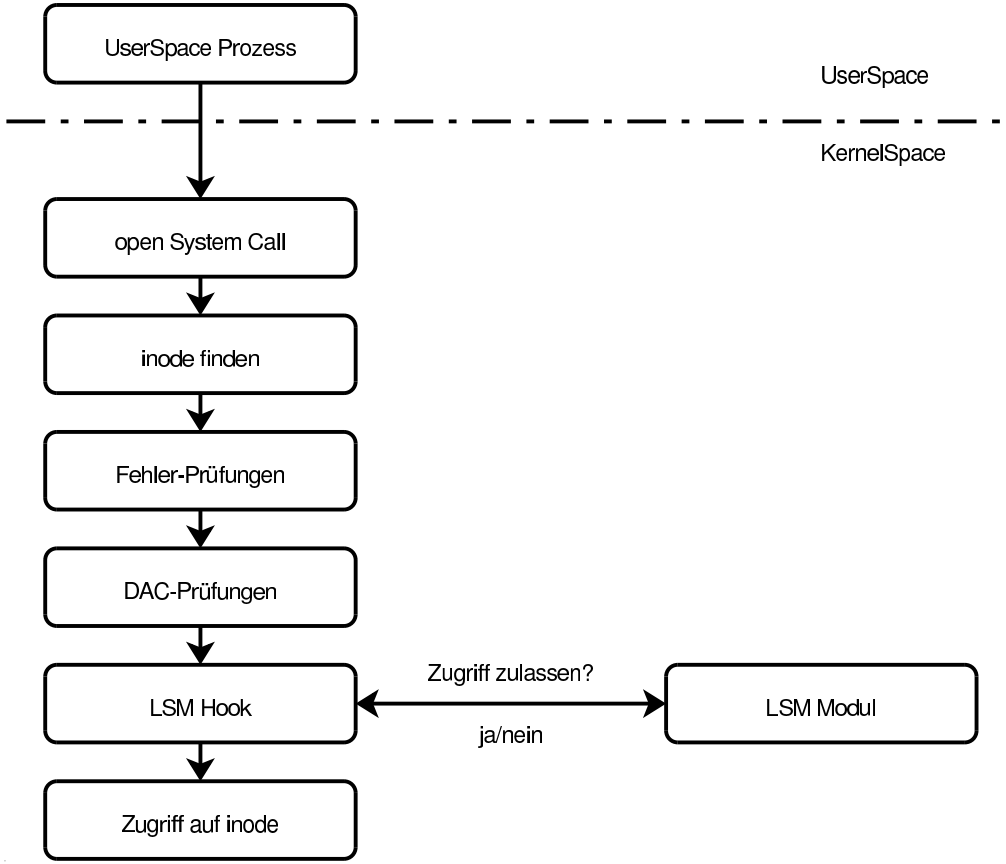
Der Umstieg

Scheduling

Sicherheit
- Linux Security Modules – LSM
- Architektur
- Architektur
- Security Fields
- Hooks
- Datenstruktur
- SC `sys_nice` mit LSM Hook
- Module Stacking
- Performance und Einsatzbereiche

Systemaufrufe

The End...



Security Fields

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit
- Linux Security Modules – LSM
- Architektur
- Architektur
- Security Fields
- Hooks
- Datenstruktur
- SC `sys_nice` mit LSM Hook
- Module Stacking
- Performance und Einsatzbereiche

Systemaufrufe

The End...

- Möglichkeit sicherheitsrelevante Daten an Kernel-Datenstrukturen anzufügen
- einfache `void*` Pointer
- Locking und Speicherverwaltung bleibt den Modulen überlassen

Hooks

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit
- Linux Security Modules – LSM
- Architektur
- Architektur
- Security Fields
- Hooks
- Datenstruktur
- SC `sys_nice` mit LSM Hook
- Module Stacking
- Performance und Einsatzbereiche

Systemaufrufe

The End...

- Hook Funktionen eines Moduls in der globalen Datenstruktur `security_ops` vom Typ `struct security_operations` registriert (Liste von Funktions-Pointern)
- diese Funktionen werden von den Hooks in den Kernel-Subsystemen aufgerufen

Datenstruktur

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit
- Linux Security Modules – LSM
- Architektur
- Architektur
- Security Fields
- Hooks
- Datenstruktur
- SC `sys_nice` mit LSM Hook
- Module Stacking
- Performance und Einsatzbereiche

Systemaufrufe

The End...


```
/* ... */
struct security_operations {
      /* ... */
      int (*task_setnice) (struct task_struct * p,
                                    int nice);
      /* ... */
};

/* global variables */
extern struct security_operations *security_ops;
/* ... */
static inline int security_task_setnice (
                  struct task_struct *p, int nice)
{
      return security_ops->task_setnice (p, nice);
}
```

– Ausschnitt aus *include/linux/security.h*

Datenstruktur `security_operations` mit `task_setnice` Funktions-Pointer und zugehöriger `security_task_setnice` Funktion *include/linux/security.h*

SC sys_nice mit LSM Hook

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit
- Linux Security Modules – LSM
- Architektur
- Architektur
- Security Fields
- Hooks
- Datenstruktur
- SC sys_nice mit LSM Hook
- Module Stacking
- Performance und Einsatzbereiche

Systemaufrufe

The End...


```
/*
* sys_nice - change the priority of the current process.
* @increment: priority increment
*
* sys_setpriority is a more generic, but much slower
* function that does similar things.
*/
asmlinkage long sys_nice(int increment)
{
      int retval;
      /* ... */
      retval = security_task_setnice(current, nice);
      if (retval)
            return retval;

      set_user_nice(current, nice);
      return 0;
      }
```

– System Call sys_nice mit LSM Hook

Module Stacking

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit
- Linux Security Modules – LSM
- Architektur
- Architektur
- Security Fields
- Hooks
- Datenstruktur
- SC `sys_nice` mit LSM Hook
- Module Stacking
- Performance und Einsatzbereiche

Systemaufrufe

The End...

- Möglichkeit um Module zu kombinieren
- LSM Framework selbst sieht immer nur ein (primäres) Modul
- Stacking basiert auf Kooperation der Module
- Module weiter vorne reichen Hook-Aufrufe durch

Performance und Einsatzbereiche

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit
- Linux Security Modules – LSM
- Architektur
- Architektur
- Security Fields
- Hooks
- Datenstruktur
- SC `sys_nice` mit LSM Hook
- Module Stacking
- Performance und Einsatzberei

Systemaufrufe

The End...

- Performance Overhead relativ gering (wenige Prozent)
- bereits mehrere Module verfügbar
- SELinux auf LSM portiert und offizieller Bestandteil von Linux 2.6

Grundlagen

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe
- Grundlagen
- Standards
- Vorhandene Systemaufrufe
- Realisierung von Systemaufrufen
- Ein kleines Beispiel
- Assembler
- Einschränkungen

The End...

- Benutzerprogramme werden im Usermode ausgeführt
  - Kein direkter Zugriff auf Ressourcen
  - Zugriff auf Ressourcen über Service-Routinen des Kernel

→ System-Calls

Standards

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe
- Grundlagen
- Standards
- Vorhandene Systemaufrufe
- Realisierung von Systemaufrufen
- Ein kleines Beispiel
- Assembler
- Einschränkungen

The End...

- Ermöglichen leichtere Portierbarkeit
- Mehrere Standards definiert
  - POSIX (Portable Operating System Interface)
  - System V
  - 4.3 BSD
- Linux bemüht sich, POSIX-Standard zu implementieren
- Andere Standards teilweise implementiert

Vorhandene Systemaufrufe

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe
- Grundlagen
- Standards
- Vorhandene Systemaufrufe
- Realisierung von Systemaufrufen
- Ein kleines Beispiel
- Assembler
- Einschränkungen

The End...

- Über 200 Systemaufrufe (architekturabhängig)
  - Prozessverwaltung
  - Zeitoperationen
  - Signalverarbeitung
  - Scheduling
  - Dateisystem
  - Speicherverwaltung (C-Standardbibliothek)
  - Interprozesskommunikation & Netzwerkfunktionen
  - Systeminformationen und -einstellungen
- Systemaufrufe werden über eindeutige Kennzahl angesprochen (architekturabhängig)

Realisierung von Systemaufrufen

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe
- Grundlagen
- Standards
- Vorhandene Systemaufrufe
- Realisierung von Systemaufru
- Ein kleines Beispiel
- Assembler
- Einschränkungen

The End...

1. Anwendung ruft Funktion aus C-Standardbibliothek auf
2. C-Bibliothek speichert Parameter in Register
3. Auslösen eines Software-Interrupts
4. Interrupt-Serviceroutine (Teil des Kernels) wechselt in Kernelmode
5. Delegiert Systemaufruf an zuständige Funktion (Systemcall-Handler)
6. Return-Wert wird über register an C-Bibliothek zurückgegeben
7. Return-Wert wird an Anwendung zurückgegeben

Ein kleines Beispiel

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe
- Grundlagen
- Standards
- Vorhandene Systemaufrufe
- Realisierung von Systemaufrufen
- Ein kleines Beispiel
- Assembler
- Einschränkungen

The End...


```
int main() {
        exit(0);
}
```

– a simple exit

Assembler

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe
- Grundlagen
- Standards
- Vorhandene Systemaufrufe
- Realisierung von Systemaufrufen
- Ein kleines Beispiel
- Assembler
- Einschränkungen

The End...


```
/* main */
[...]
push   $0x0
call   0x8048488 <exit>

/* exit */
[...]
mov    0x4(%esp,1),%ebx
mov    $0x1,%eax
int    $0x80
[..]
```

Einschränkungen

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe
- Grundlagen
- Standards
- Vorhandene Systemaufrufe
- Realisierung von Systemaufrufen
- Ein kleines Beispiel
- Assembler
- Einschränkungen

The End...

- Nummer des Systemaufrufs wird über Register eax übergeben
- Parameterübergabe mittels Register
  - Nur primitive Datentypen
  - max. 5 Parameter möglich
  - komplexe Datentypen per Referenz
- Kein direkter Zugriff auf Prozessspeicher möglich

Literatur

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- Literatur
- The End...

- Markus Quaritsch, Thomas Winkler: „Linux - Ein Einblick in den Kernel“ http://uni.qwws.net/linpro/seminar
- Robert Love: „Linux Kernel Development“, SAMS Publishing, Indianapolis (2003)
- Wolfgang Mauerer: „Linux Kernelarchitektur - Konzepte, Strukturen und Algorithmen von Kernel 2.6“, Carl Hanser Verlag, München (2003)
- Daniel P. Bovet, Marco Cesati: „Understanding the Linux Kernel, Second Edition“, O’Reilly & Associates, Sebastopol (2002)
- Online: www.linuxtage.at/prokop/
- Weitere Informationen: http://www.michael-prokop.at/computer/linux_kernel.html

The End...

Einleitung

Geschichte

Änderungen

Skalierung

Beispiele: USB, ALSA,..

Der Umstieg

Scheduling

Sicherheit

Systemaufrufe

The End...

- Literatur
- The End...

Danke für die Aufmerksamkeit!
Feedback ist willkommen!

michael@linuxtage.at
quam@qwws.net
tom@qwws.net

Viel Spaß noch auf den Grazer Linuxtagen!