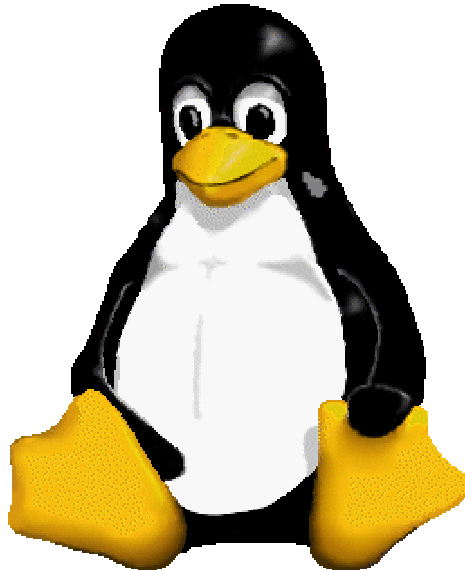


Der Linux Kernel

Tilmann Bitterberg

<http://tibit.org/linux.html>

HfT-Stuttgart, den 15.06.2001



Intro

Intro in den Kernel

Struktur

Algorithmen und...

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 1 of 46

Go Back

Full Screen

Close

Quit

1. Intro



Intro

Intro in den Kernel

Struktur

Algorithmen und . . .

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 2 of 46

Go Back

Full Screen

Close

Quit

1.1. Was ist Linux

- Ein Unix ähnliches Betriebssystem, das auf einer Vielzahl von Hardware Architekturen läuft.
- kostenlos aus dem Internet herunterzuladen <http://www.kernel.org>.
- besteht aus einem Kernel (*Linux*) und einem Betriebssystem (*GNU*)



Intro

Intro in den Kernel

Struktur

Algorithmen und...

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 3 of 46

Go Back

Full Screen

Close

Quit

1.2. Minix

- von Andy Tanenbaum für 8088er PCs (original IBM-PC) entwickeltes Lehrbetriebssystem.
- Microkernel Design.
- sehr restriktive Lizenz, Veränderungen durften nur in *patch* Form weitergegeben werden.
- Nicht kostenlos, ca. \$150 pro Lizenz.



Intro

Intro in den Kernel

Struktur

Algorithmen und ...

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 4 of 46

Go Back

Full Screen

Close

Quit

1.3. Woher kommt Linux

- 1991 kaufte ein finnischer Informatik Student namens Linus Torvalds einen 386er PC.
- Unzufrieden mit DOS, wünschte er sich ein besseres Betriebssystem
- "Herumspielen" mit dem Lehrbetriebssystem Minix.
- Beschluß, ein besseres Minix als Minix zu schreiben.
- Durch die lange Geschichte von Unix, die Eleganz des grundlegenden Designs und die Verfügbarkeit einer Vielzahl an Software, beschloß Linus ein Unix-ähnliches Betriebssystem zu schreiben.



Intro

Intro in den Kernel

Struktur

Algorithmen und ...

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 5 of 46

Go Back

Full Screen

Close

Quit

Am 5. Oktober 1991 schrieb Linus in der Newsgroup comp.os.minix:

''Do you pine for the nice days of Minix-1.1, when men were men and wrote their own device drivers? Are you without a nice project and just dying to cut your teeth on a OS you can try to modify for your needs? Are you finding it frustrating when everything works on Minix? No more all-nighters to get a nifty program working? Then this post might be just for you.

As I mentioned a month ago, I'm working on a free version of a Minix-lookalike for AT-386 computers. It has finally reached the stage where it's even usable (though may not be depending on what you want), and I am willing to put out the sources for wider distribution. It is just version 0.02...but I've successfully run bash, gcc, gnu-make, gnu-sed, compress, etc. under it.''



Intro

Intro in den Kernel

Struktur

Algorithmen und...

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 6 of 46

Go Back

Full Screen

Close

Quit

1.4. GNU

- GNU steht für GNU is not Unix
- Sammlung von freien Unix Programmen von der FSF.
- frei im Sinne von Redefreiheit, nicht Freibier.
- 1984 vom MIT Programmierer Richard Stallman entwickelt um eine freie, unix-ähnliche Arbeitsumgebung zu schaffen.
- Ursprünglich *hurd* als Kernel konzipiert, Linux war schneller.
- Bekannte Projekte aus dieser Sammlung: emacs, gcc, bash, usw.



Intro

Intro in den Kernel

Struktur

Algorithmen und ...

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 7 of 46

Go Back

Full Screen

Close

Quit

1.5. GNU/Linux

- Benutzt GNU Software für die Arbeitsumgebung
- Benutzt Linux als Kernel
- Beides steht unter der *GNU Public License (GPL)*
- Die GPL erlaubt unbeschränkte Modifikation und Verteilung der Software.



Intro

Intro in den Kernel

Struktur

Algorithmen und ...

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 8 of 46

Go Back

Full Screen

Close

Quit

1.6. Linux-Distributionen

- Bieten eine Sammlung von Programmen, die auf einander abgestimmt sind.
- Entwickeln Installationsanweisungen, -Skripte und Konfigurationstools.
- Man spart sich die Downloadkosten.
- Verlangen Geld für die Zusammenstellung und Support
- Bekannte Distributionen: RedHat, SuSE, GNU/Debian, Caldera



Intro

Intro in den Kernel

Struktur

Algorithmen und . . .

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 9 of 46

Go Back

Full Screen

Close

Quit

1.7. Wer braucht Linux?

Menschen, die

- ein modernes, zuverlässiges Betriebssystem haben wollen.
- wissen wollen, was hinter den Kulissen vorgeht.
- sich nicht mit teurer, fehlerbehafteter Software zufrieden geben.
- kostengünstige Webserver betreiben wollen.
- optimale Performance aus ihrem Rechner herausholen wollen.
- einfach Software entwickeln wollen.



Intro

Intro in den Kernel

Struktur

Algorithmen und . . .

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 10 of 46

Go Back

Full Screen

Close

Quit

1.8. Portabilität

- Linux 2.4 läuft auf 15 verschiedenen Hardware Plattformen
- durch Portierung auf viele Plattformen wurde eine saubere Schnittstelle geschaffen.
- Neue Architekturen hinzu zufügen ist relativ einfach.
- Gerätetreiber sind meist portabel geschrieben.



Intro

Intro in den Kernel

Struktur

Algorithmen und ...

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 11 of 46

Go Back

Full Screen

Close

Quit



Intro

Intro in den Kernel

Struktur

Algorithmen und . . .

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

[Home Page](#)

[Title Page](#)



Page 12 of 46

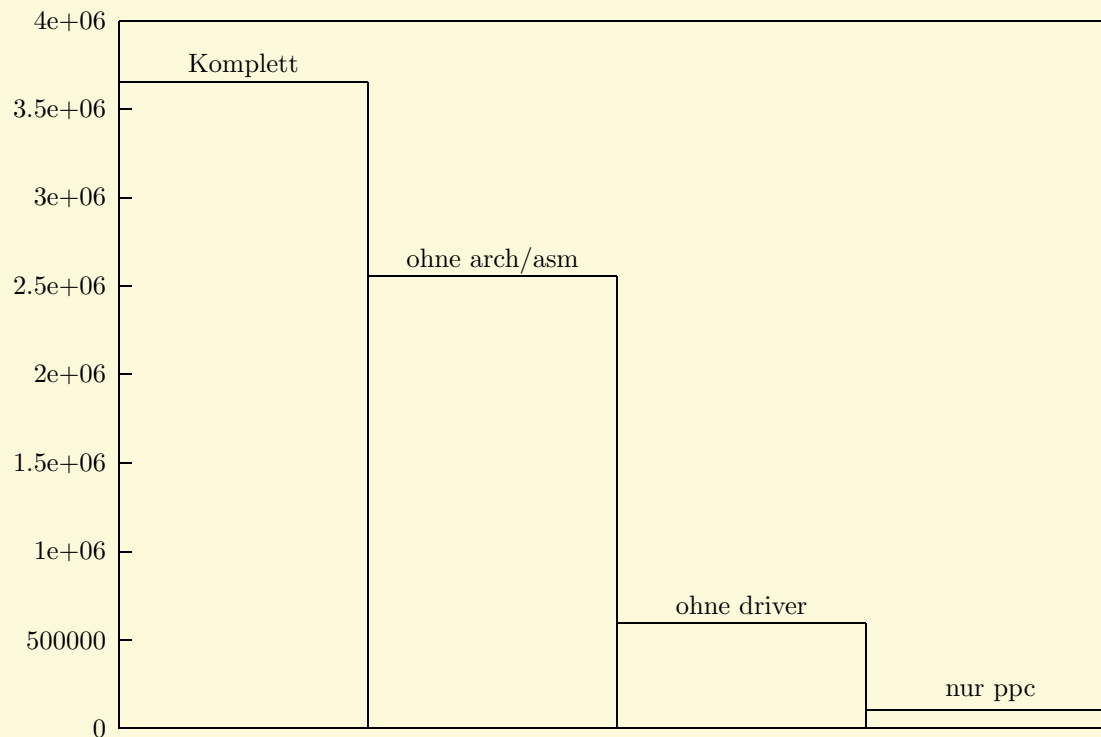
[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

Linecount



2. Intro in den Kernel



Intro

Intro in den Kernel

Struktur

Algorithmen und . . .

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 13 of 46

Go Back

Full Screen

Close

Quit

2.1. Download

- Zum Kernel lesen braucht man nur den Source.
- Gehe auf <http://www.kernel.org> oder direkt unter (ca. 24MB)
<ftp.kernel.org/pub/linux/kernel/v2.4/linux-2.4.5.tar.gz>
- entpacke das TAR Archiv. `tar xvzf linux-2.4.5.tar.gz`
- alles steht im Unterverzeichnis `linux/`



Intro

Intro in den Kernel

Struktur

Algorithmen und...

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 14 of 46

Go Back

Full Screen

Close

Quit

2.2. Wer programmiert Linux

- Tausende von Entwicklern, die über das Internet kommunizieren.
- Austausch über die *linux-kernel* Mailing Liste und IRC.
- Wöchentliche Zusammenfassung von *linux-kernel* unter <http://kt.zork.net>
- Linus Torvalds hat das letzte Wort was im Kernel ist.
- Alan Cox ist Linus rechte Hand.
- Jedes Subsystems/jeder Treiber hat seinen eigenen Maintainer
- Maintainer sammeln Code von anderen Leuten und schicken das an Linus
- Linus vertraut Maintainern.
- Alle Maintainer stehen in `linux/CREDITS`
- Linux Who-is-Who (mit Bildern) unter http://www.linux-mag.com/who/lmwho_1.html



Intro

Intro in den Kernel

Struktur

Algorithmen und ...

Spezielle Dateisysteme

Module

Beispiel `hft-demo.c`

Home Page

Title Page



Page 15 of 46

Go Back

Full Screen

Close

Quit



Linus Torvalds



Intro

Intro in den Kernel

Struktur

Algorithmen und . . .

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



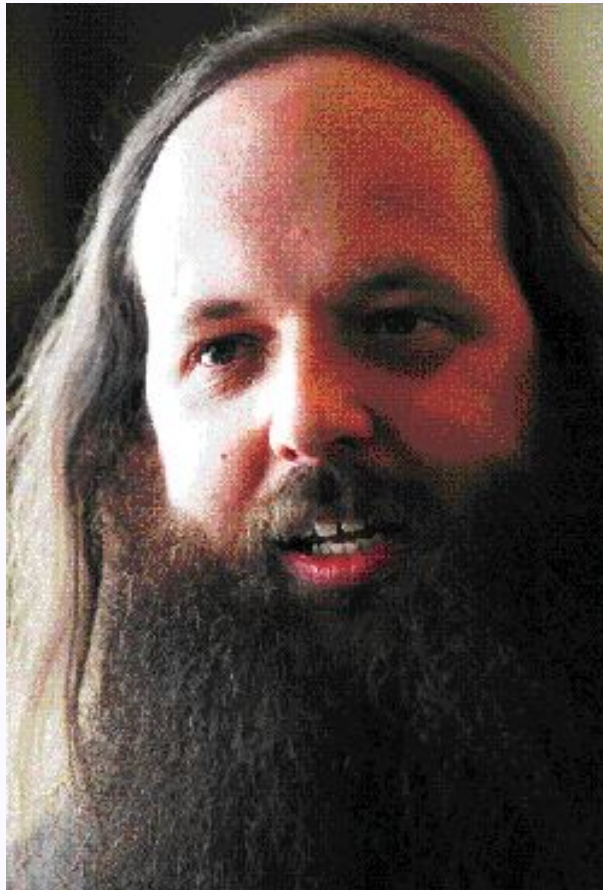
Page 16 of 46

Go Back

Full Screen

Close

Quit



Alan Cox



Intro

Intro in den Kernel

Struktur

Algorithmen und . . .

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 17 of 46

Go Back

Full Screen

Close

Quit

2.3. Versionsnummern

- Zusammensetzung `VERSION.PATCHLEVEL.SUBLEVEL`
- aktuell: 2.4.5
- wenn `PATCHLEVEL` eine gerade Zahl, dann *stabile* Version
- wenn `PATCHLEVEL` eine ungerade Zahl, dann *Entwickler* Version
- aus jeder stabilen Version geht nach einiger Zeit eine Entwickler Version hervor; die nächste wird 2.5.x heißen.
- Alte Versionen haben durchaus ihre Berechtigung.
- Version 2.0 wurde bereits Juni 1996 freigegeben.



Intro

Intro in den Kernel

Struktur

Algorithmen und ...

Spezielle Dateisysteme

Module

Beispiel `hft-demo.c`

Home Page

Title Page



Page 18 of 46

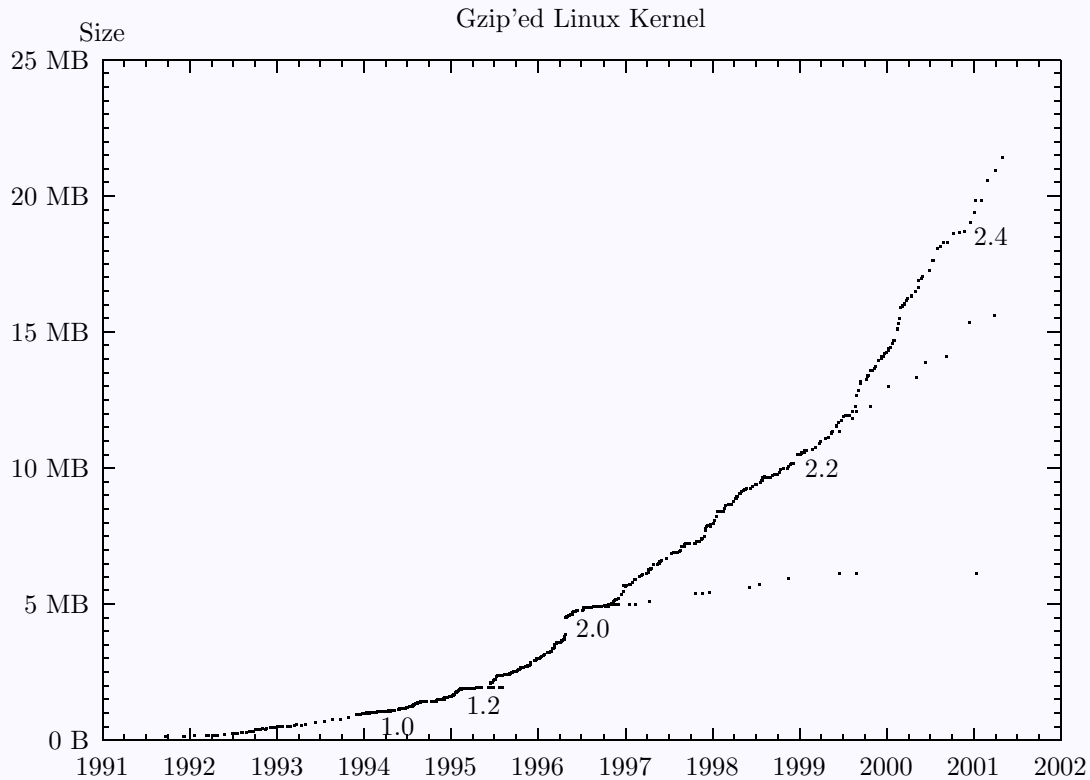
Go Back

Full Screen

Close

Quit

2.4. Zeitschiene



Intro

Intro in den Kernel

Struktur

Algorithmen und ...

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 19 of 46

Go Back

Full Screen

Close

Quit

3. Struktur



Intro

Intro in den Kernel

Struktur

Algorithmen und . . .

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 20 of 46

Go Back

Full Screen

Close

Quit

3.1. Subsysteme

- Prozess Verwaltung
- Speicher Verwaltung
- Dateisysteme
- Geräte Kontrolle
- Vernetzung.



Intro

Intro in den Kernel

Struktur

Algorithmen und . . .

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 21 of 46

Go Back

Full Screen

Close

Quit



Intro
Intro in den Kernel

Struktur

Algorithmen und ...
Spezielle Dateisysteme
Module
Beispiel hft-demo.c

Home Page

Title Page



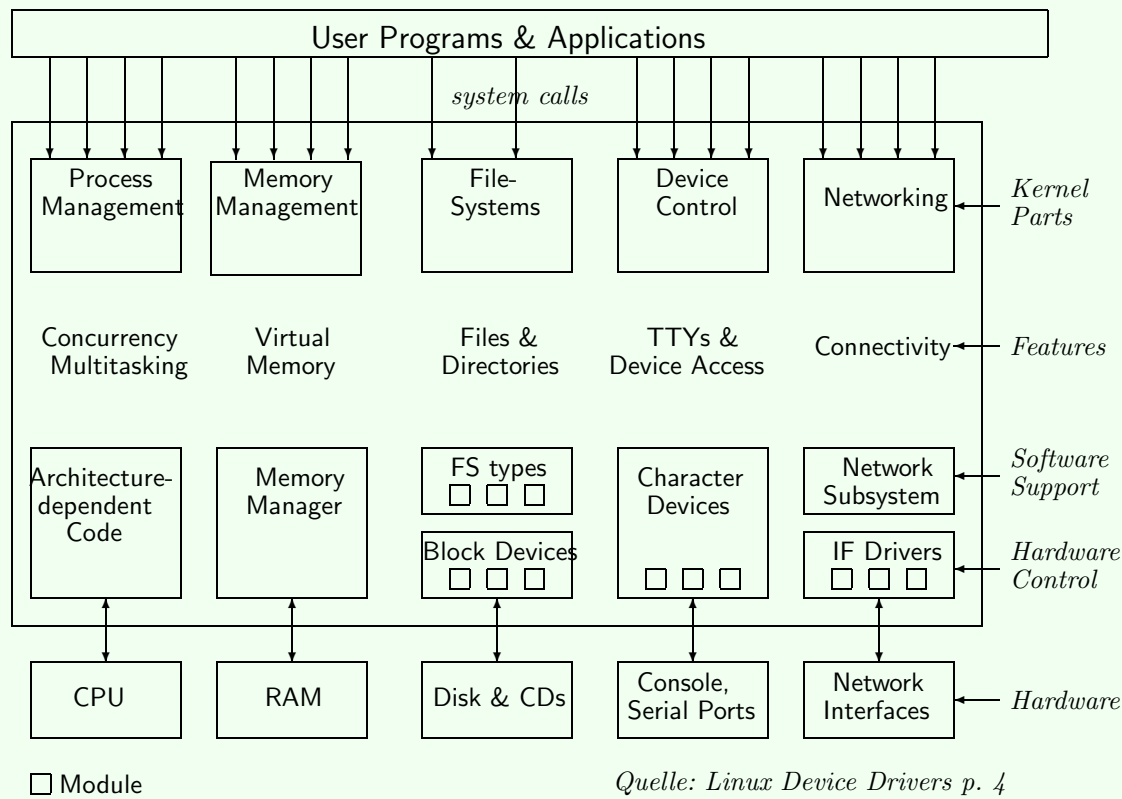
Page 22 of 46

Go Back

Full Screen

Close

Quit



Quelle: Linux Device Drivers p. 4

3.2. Physikalische Orientierung

- Aufteilung in logische Unterverzeichnisse; Beispiele:
- TV Karten
`linux/drivers/media/video/`
- IP-Stack
`linux/net/ipv4/`
- Netzwerk Karten
`linux/drivers/net/`
- Bootbarer Kernel
`linux/arch/i386/boot/bzImage`



Intro

Intro in den Kernel

Struktur

Algorithmen und ...

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 23 of 46

Go Back

Full Screen

Close

Quit

linux/

— arch/	Architekturspezifische Dateien
— Documentation/	Anwender Dokumentation
— drivers/	Gerätetreiber
— fs/	Dateisysteme
— include/	Öffentliche Header Dateien
— init/	Das main.c file des Kernels
— ipc/	Prozess Kommunikation
— kernel/	Core Kernel (eg. Scheduler)
— lib/	Nützliche Hilfsfunktionen
— mm/	Core Kernel Memory Management
— net/	Netzwerk Protokolle (Stacks)
— scripts/	Skripte, die das Leben leichter machen



[Intro](#)

[Intro in den Kernel](#)

Struktur

[Algorithmen und ...](#)

[Spezielle Dateisysteme](#)

[Module](#)

[Beispiel hft-demo.c](#)

[Home Page](#)

[Title Page](#)



Page 24 of 46

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

3.3. linux/arch/*

- Aufspaltung von Linux in Rechner Architektur abhängige und unabhängige Teile.
- in arch/* steht Rechner Architektur abhängiges



Intro

Intro in den Kernel

Struktur

Algorithmen und . . .

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 25 of 46

Go Back

Full Screen

Close

Quit

linux/
└─ arch/

— alpha/	Frueher Digital, jetzt Compaq
— arm/	Emebdedd, x86 aehnlich
— cris/	?
— i386/	Intel Pentium, AMD Athlon, ...
— ia64/	Intel Titanium 64Bit
— m68k/	Motorola M680000 (Amiga, Atari)
— mips/	SGI
— mips64/	SGI 64Bit
— parisc/	HP Risc Prozessoren
— ppc/	IBM/Apple/Motorola PowerPC
— s390/	IBM 31Bit s/390
— s390x/	IBM 64Bit s/390
— sh/	?
— sparc/	SUN Microsystems
— sparc64/	SUN Microsystems 64Bit



Intro

Intro in den Kernel

Struktur

Algorithmen und ...

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

[Home Page](#)

[Title Page](#)



Page 26 of 46

[Go Back](#)

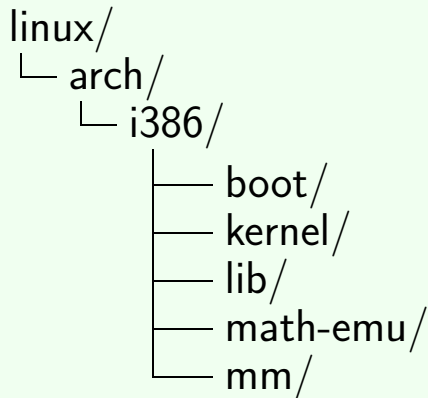
[Full Screen](#)

[Close](#)

[Quit](#)

3.4. Beispiel i386

- Hauptsächlich Low-Level, Hardwarenahes
- Assembler



x86 Bootsektor, bootbarer Kernel
Low-Level Kernel, zB IRQ-Handling
hochoptimierte Hilfsfunktionen
Emulation des math. Coprozessors
Low-Level Speichermanager



[Intro](#)

[Intro in den Kernel](#)

Struktur

[Algorithmen und ...](#)

[Spezielle Dateisysteme](#)

[Module](#)

[Beispiel hft-demo.c](#)

[Home Page](#)

[Title Page](#)



Page 27 of 46

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)



Intro

Intro in den Kernel

Struktur

Algorithmen und . . .

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 28 of 46

Go Back

Full Screen

Close

Quit

3.5. Verzeichnisgrößen

linux (129368)	fs (9272)	nls (3204)
	include (18776)	linux (4268)
	net (6220)	net (15136)
	drivers (64332)	char (7164)
		scsi (11824)
		sound (4444)
		isdn (3544)
		video (3920)
		usb (3044)
		ppc (2916)
	n68k (4340)	
	ia64 (2520)	
	Documentation (5220)	



- Intro
- Intro in den Kernel
- Struktur**
- Algorithmen und ...
- Spezielle Dateisysteme
- Module
- Beispiel hft-demo.c

Home Page

Title Page

Navigation: << >>

Navigation: < >

Page 29 of 46

Go Back

Full Screen

Close

Quit

4. Algorithmen und Datenstrukturen



Intro

Intro in den Kernel

Struktur

Algorithmen und . . .

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 30 of 46

Go Back

Full Screen

Close

Quit

4.1. Allgemeines

- Alles möglichst simpel gehalten (dort wo es die Performance zulässt)
- Viele Algorithmen "einfach" aus Lehrbüchern umgesetzt
- Einfache Listenstrukturen (einfach/doppelt verkettet, Queues)
- Fast alle Datenstrukturen dienen dazu, Asynchronität im Kernel bereitzustellen.



Intro

Intro in den Kernel

Struktur

Algorithmen und...

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 31 of 46

Go Back

Full Screen

Close

Quit

4.2. Scheduler

- Findet man in `linux/kernel/sched.c`
- Implementiert Round-Robin mit Prioritäten (goodness)
- Zeitscheibe ist ca. 20ms lang.



Intro

Intro in den Kernel

Struktur

Algorithmen und...

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 32 of 46

Go Back

Full Screen

Close

Quit

4.3. Virtueller Speicher

- LRU für swapping
- Balancing Zones
- Memory overcommitter (tut meistens)



Intro

Intro in den Kernel

Struktur

Algorithmen und . . .

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 33 of 46

Go Back

Full Screen

Close

Quit

4.4. Wait Queues

- Wenn ein Prozess etwas machen/haben will, was gerade nicht möglich ist, wird er in eine Warteschlange gesteckt.
- Will man selber Warteschlangen benutzen, kann man entweder eine schon vorhandene aus dem Kernel nehmen oder selber eine deklarieren mit `DECLARE_WAIT_QUEUE_HEAD()`, `DECLARE_WAIT_QUEUE()`
- Bekannte Warteschlangen ist z.B die `kswapd_wait`, in der der kernel swap daemon schläft, wenn es nicht zum swappen gibt.



Intro

Intro in den Kernel

Struktur

Algorithmen und ...

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 34 of 46

Go Back

Full Screen

Close

Quit

4.5. Kernel Timers

- erweiterte Warteschlange.
- Benutzt man um eine Funktion zu einem späteren Zeitpunkt ausführen zu lassen.
- Das Datenelement wird automatisch aus der `timer_list` entfernt, wenn der timer abgelaufen ist.



Intro

Intro in den Kernel

Struktur

Algorithmen und...

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 35 of 46

Go Back

Full Screen

Close

Quit

4.6. Bottom Halves

- Innerhalb eines Interrupt handlers sollte nur relativ kurzer Code stehen.
- Dieser Code wird mit abgeschalteten Interrupts ausgeführt
- Wenn Code zu lang, gehen Datenpakete verloren.
- Bottom Halves bestehen aus zwei Teilen
- Kurzes Vorderteil, das während des Interrupts läuft
- Langes Hinterteil, das später ausgeführt wird.



Intro

Intro in den Kernel

Struktur

Algorithmen und...

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 36 of 46

Go Back

Full Screen

Close

Quit

4.7. Task Queues

- Erweiterung der Bottom Halves
- Mehr Flexibilität, dynamischer
- Benutzung mit `DECLARE_TASK_QUEUE()` und `queue_task()`



Intro

Intro in den Kernel

Struktur

Algorithmen und . . .

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 37 of 46

Go Back

Full Screen

Close

Quit

5. Spezielle Dateisysteme



Intro

Intro in den Kernel

Struktur

Algorithmen und ...

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 38 of 46

Go Back

Full Screen

Close

Quit

5.1. proc

- beinhaltet Informationen über den Kernel und das laufende System.
- keine echten Dateien, werden vom Kernel beim lesen generiert.
- bietet einstellbare Kernel-Parameter zur Laufzeit.
- einfach programmierbar.



Intro

Intro in den Kernel

Struktur

Algorithmen und ...

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 39 of 46

Go Back

Full Screen

Close

Quit

5.2. devfs

- statt statischen Geräte Einträgen in `/dev` wird bei devfs wirklich nur das angezeigt, was tatsächlich im System ist.
- logisch und physikalisch gegliedert.
- sorgt für Ordnung im `/dev` Verzeichnis.
- Früher die 1. Partition auf der 1. Festplatte im System hieß `/dev/hda1`, jetzt `/dev/discs/disc0/part1` oder `/dev/ide/host0/bus0/target0/lun0/part1`
- noch nicht alle Treiber wurden umgeschrieben.



Intro

Intro in den Kernel

Struktur

Algorithmen und ...

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 40 of 46

Go Back

Full Screen

Close

Quit

5.3. tmpfs

- Schnittstelle zum RAM
- Kompatibel zum MIT shm
- wenn tmpfs gemounted ist, kann als ganz normales Filesystem darauf zugegriffen werden.



Intro

Intro in den Kernel

Struktur

Algorithmen und ...

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 41 of 46

Go Back

Full Screen

Close

Quit

6. Module

- Module sind object files (.o), die nicht gelinkt sind
- Können dynamisch zur Laufzeit des Kernels hinzugeladen werden.
- Laufen dann im Kernelspace, haben also unbeschränkten Zugriff auf alles.
- Laden mit insmod, entladen mit rmmod.
- Initialisierung mit `module_init()`, `module_exit()`



Intro

Intro in den Kernel

Struktur

Algorithmen und ...

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 42 of 46

Go Back

Full Screen

Close

Quit

7. Beispiel hft-demo.c

```
/*
 * Demo proc entry
 *
 * Written by Tilmann Bitterberg
 */

#include <linux/module.h>
#include <linux/init.h> /* for module_init */
#include <linux/errno.h>
#include <linux/sched.h> /* for task struct */
#include <linux/proc/fs.h>

#include <asm/uaccess.h>

static ssize_t proc_demo_read(struct file * file, char * buf,
                             size_t count, loff_t *ppos);
static ssize_t proc_demo_write(struct file * file,
                              const char * buffer,
                              size_t count, loff_t *ppos);
struct file_operations proc_demo_operations = {
owner:         THIS_MODULE,
    read:      proc_demo_read,
    write:    proc_demo_write,
```



Intro

Intro in den Kernel

Struktur

Algorithmen und...

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 43 of 46

Go Back

Full Screen

Close

Quit

```

};

static ssize_t proc_demo_read(struct file * file, char * buf,
    size_t count, loff_t *ppos)
{
    int n = 0;
    struct task_struct *p;
    char buffer[512];

    static int number=0;

    number++;

    n = sprintf (buffer ,
        "Ich wurde %d mal gelesen\nPids aktive: \n", number);
    for_each_task(p)
        n += sprintf (buffer+n, "%d ", p->pid);
    n += sprintf (buffer+n, "\n");

    if (count < 0)
        return -EINVAL;

    if (*ppos >= strlen(buffer))
        return 0;

```



Intro

Intro in den Kernel

Struktur

Algorithmen und...

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 44 of 46

Go Back

Full Screen

Close

Quit

```

if (n > strlen(buffer) - *ppos)
n = strlen(buffer) - *ppos;
if (n > count)
n = count;
copy_to_user(buf, buffer + *ppos, n);
*ppos += n;
return n;
}

static ssize_t proc_demo_write(struct file * file,
                               const char * buffer, size_t count, loff_t *ppos)
{
return 0;
}

MODULE_DESCRIPTION("Proc demo module");
MODULE_AUTHOR("Tilman Bitterberg");

static void cleanup_demoproc(void)
{
remove_proc_entry("hft-demo", NULL);
}

static int init_demoproc(void)
{
struct proc_dir_entry *entry;

```



Intro

Intro in den Kernel

Struktur

Algorithmen und...

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page

◀▶

◀▶

Page 45 of 46

Go Back

Full Screen

Close

Quit

```
entry = create_proc_entry("hft-demo", S_IRUGO|S_IWUSR, NULL);
if (entry) entry->proc_fops = &proc_demo_operations;

return 0;
}

module_init(init_demoproc);
module_exit(cleanup_demoproc);
```



Intro

Intro in den Kernel

Struktur

Algorithmen und...

Spezielle Dateisysteme

Module

Beispiel hft-demo.c

Home Page

Title Page



Page 46 of 46

Go Back

Full Screen

Close

Quit