Linux - Der Kernel

Daniela Staritzbichler, Felix Wolfersberger, Bernhard Schauer

18. Jänner 2008

Übersicht

Übersicht

- Nur der eigentliche Kern - Torvalds bezeichnet auch Systeme als Linux
- Multiplatform - Multiuser Betriebssystem
- Client und Serversystem
- Alle Linux Distributionen enthalten den Kern
- Registriertes Markenzeichen

Geschichte

- 1991: Torvalds begann die Entwicklung
- 25.08.1991: Erste Ankündigung in comp.os.minix
- 17.09.1991: Version 0.01
- 1992: GNU/GPL Lizenziert
- 1992: Tanenbaum Posting: “Linux is obsolete“
- Den Rest zeigt die Geschichte - Linux heute

Die Köpfe

- Linus Torvalds
  Hauptentwickler und “Vater“ von Linux
  Beteiligt sich immer noch aktiv
  Beschäftigt bei der Linux Foundation

- Andrew Morton
  Maintainer des aktuellen Kerns 2.6
  eigener -mm Tree des Kernels
  Beschäftigt bei Google

- Greg Kroah-Hartman
  Maintainer der stabilen Kernel 2.6.x.y
  Maintainer für einige Subsysteme
  Beschäftigt bei den Novell SuSE Labs

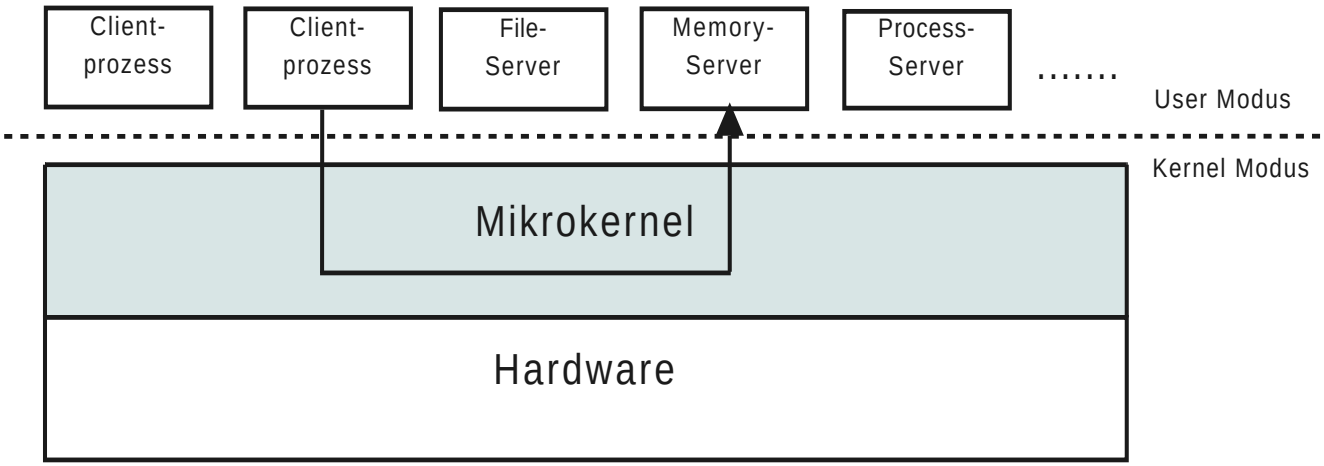
Der Entwicklungsprozeß

- Liberales Entwicklungskonzept
- Jeder kann über die Mailingliste beitragen
- Qualität durch Kontrolle u.a. von Torvalds
- Privatpersonen, Firmen und Konzerne tragen aktiv bei

Übersicht

Mikrokernel: eine allgemeine Einführung

- Funktionen zur Speicher- und Prozessverwaltung und Grundfunktionen zur Synchronisation und Kommunikation.
- Der Kernel übernimmt die Abwicklung der Kommunikation zwischen Anwendungsprozessen und Serverprozessen.



Monolithischer Kernel: eine allgemeine Einführung

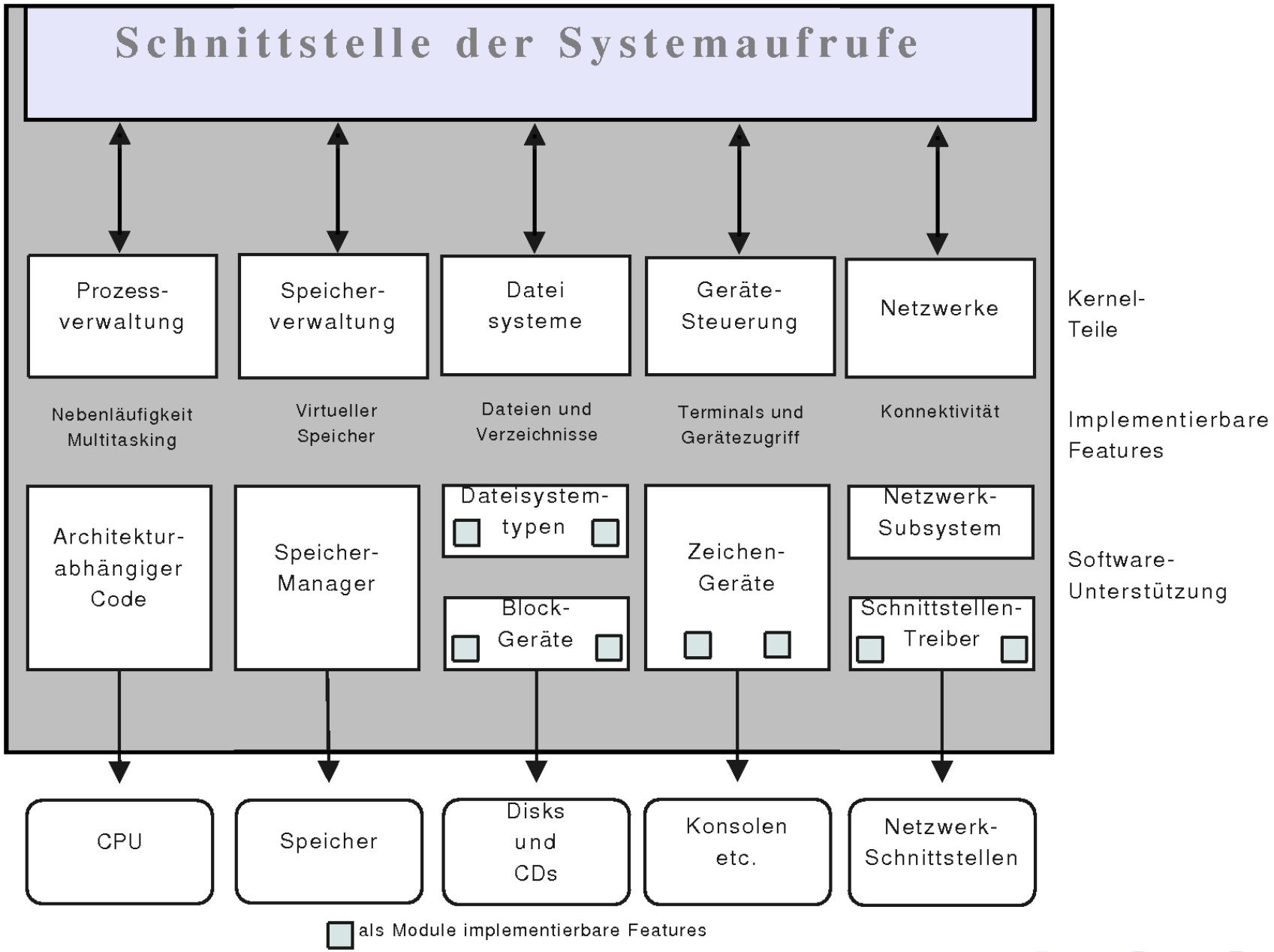
- Der Betriebssystemkern umfasst alle Dienste des Betriebssystems, die möglichst immer geladen sein sollen.
- Dazu gehören unter anderem:

  Prozessverwaltung
  Speicherverwaltung
  Dateiverwaltung
  Geräteverwaltung (Treibersoftware)
  Netzwerkverwaltung
- Module können dynamisch nachgeladen werden (in begrenztem Umfang)
- Typisches Betriebssystem ist Linux.

Der Linux Kernel

- Linux ist der eigentliche Systemkern
- Der Kern koordiniert und verteilt Ressourcen
- Durch kompilieren eines eigenen Kernels ist es möglich nur die benötigten Gerätetreiber, Dateisysteme und Kernel Funktionen einzubinden.
- Kernel ist optimal an die Hardware angepasst.

Der Linux Kernel: Aufbau



Der Linux Kernel: Aufbau

- Prozessverwaltung - Was ist ein Prozess?
  - Linux trennt streng zwischen Programm und Umgebung
  - Linux ordnet Prozesse hierarchisch anhand eindeutiger Nummern
- Speicherverwaltung: Bei Linux gibt es die Swap-Partition.
- Gerätesteuerung: Gerätedateien stellen die Schnittstelle zwischen Gerätetreibern und Hardware.
  - Zeichenorientierte Treiber: Zugriff erfolgt zeichenweise.
  - Blockorientierte Treiber: Zugriff wird in größeren Datenblöcken erlaubt.
  - Pseudo-Geräte-Treiber: Softwaretreiber, die keine Hardware kontrollieren.

Der Linux Kernel: Aufbau

- Netzwerke: Die Fähigkeit das Internetprotokoll TCP/IP zu benutzen ist im Kern implementiert.
- Dateisysteme: Linux versteht die Sprache vieler Dateisysteme.
  - ext2: Linux
  - iso 9660: CD-ROM
  - msdos: DOS
  - ntfs: WindowsNT
  - Vfat: Windows95
  - .....
- Der entsprechende Zugriff auf die verschiedenen Dateisysteme ist durch das virtuelle Dateisystem (VFS) möglich.

Das virtuelle Filesystem

- Eine einheitliche Schnittstelle für die Vielzahl der von Linux unterstützten Dateisysteme
- Definition eines universalen Filesystems, auf das sich die unterschiedlichen Dateisysteme abbilden lassen.
- Das Modell definiert (Auswahl):
  - verschiedene Dateiarten
  - Liste der Datenblöcke
  - Dateigröße, Eigentümer, Gruppe und Zugriffsrechte.
  - Zeitpunkt der letzten Modifikation, der Dateierstellung, des letzten Dateizugriffs, usw.
- Speicherung im sogenannten Inode.
- Jedes Verzeichnis ist auch eine Datei

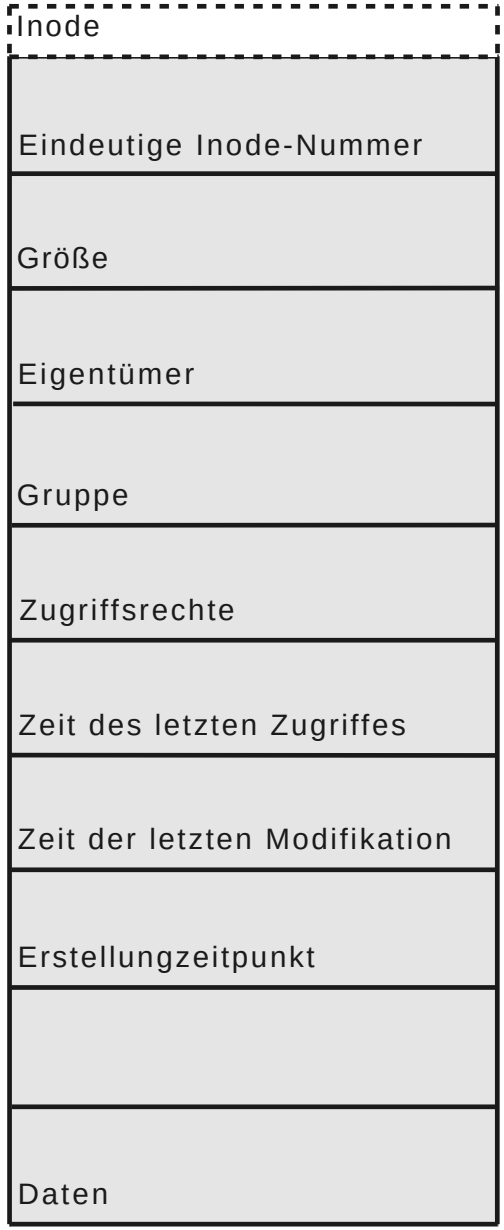


Abbildung: Aufbau eines Inode

Das virtuelle Filesystem

- Wie die Daten auf die einzelnen Blöcke eines Dateisystems verteilt werden oder wo die Inode-Informationen liegen, ist Sache des jeweiligen Filesystems.
- Um auf ein Dateisystem zugreifen zu können, muss der Inode des Root-Verzeichnisses bekannt sein.
- Im Modell ordnet das Root-Verzeichnis die Subverzeichnisse und die abgelegten Dateien den jeweils zugehörigen Inodes zu.
- Diese einzelne Zuordnung wird als Dentry (Directory Entry) bezeichnet.
- Da nur die Kenntnis des Inodes einen Zugriff auf die Daten der Datei erlaubt, ist das Auffinden eines Inodes die wichtigste Operation des virtuellen Filesystems.
- Beispiel: /etc/xml/catalog

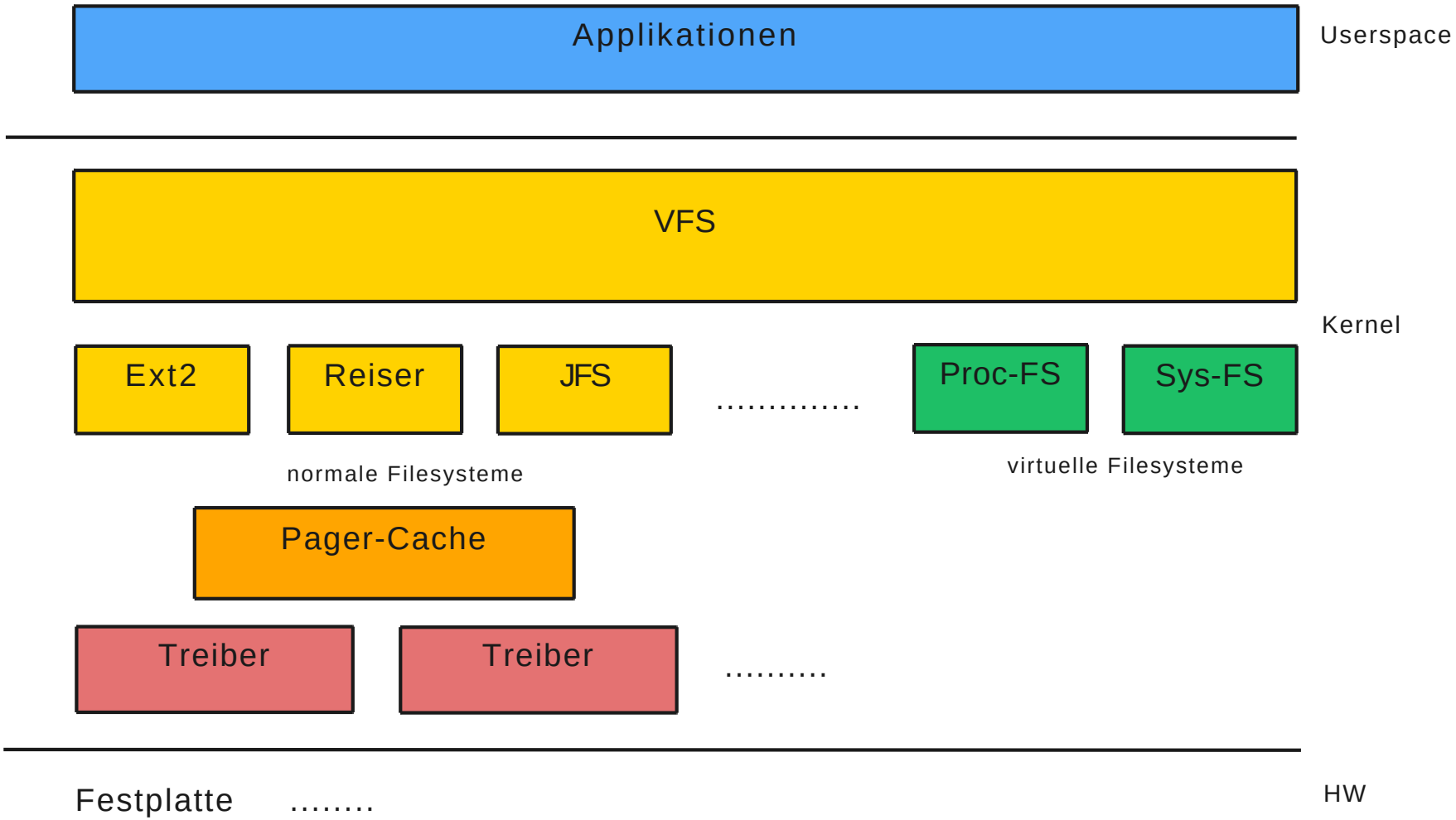


Abbildung: Beim Zugriff auf Daten sind mehrere Komponenten des Kernels beteiligt, die schichtartig aufgebaut sind. Unten im Kernel befinden sich die Gerätetreiber, ganz oben das VFS.

Das virtuelle Filesystem

- Aus Performancegründen sucht der Kernel nicht jedemal aufs Neue mit jeder Pfadkomponente den zugehörigen Inode, sondern speichert die Suchanfrage zusammen mit dem Ergebnis in einem Dentry-Objekt. Dei Dentry-Objekte werden in einem Dcache (Dentry Cache) gespeichert.

Übersicht

Systemcalls - Was ist das?

- Interface zwischen Kernel und Userspace
- Alle Funktionen wie (kleiner Auszug):
  - Öffnen, Lesen, Schreiben von Deskriptoren
  - Setzen von GID und UID, chroot
  - Setzen von Prozessprioritäten
  - uvm.
- IA32: Realisiert über Softwareinterrupts bzw. sysenter/sysexit
- Alpha: Befehl call_pal
- PPC: Prozessorbefehl sc (syscall)
- x86_64: Prozessorbefehl syscall

Systemcalls - Was ist das?

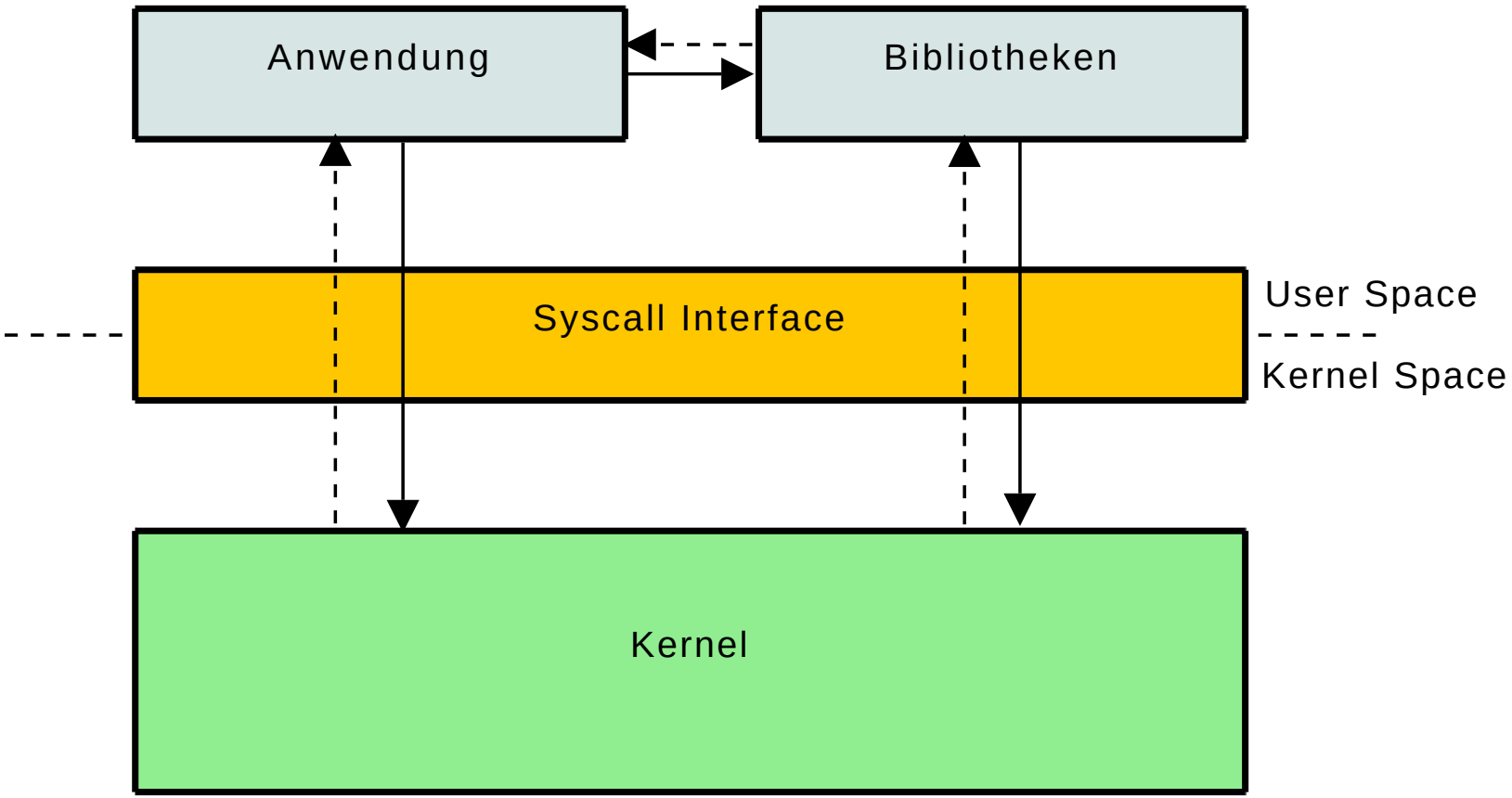


Abbildung: Verwendung der SysCalls - alles ist möglich

Was sind sie?

- Alles ist ein File, auch die Hardware
- Festplatten, Schnittstellen (RS232, USB, ...) werden als File repräsentiert
- Direkter Zugriff möglich
- Gerät wird über Major/Minor Nummern referenziert

```
brw-r-----   1 root     disk   8,   0 24. Jan 11:13 sda
brw-r-----   1 root     disk   8,   1 24. Jan 11:13 sda1
brw-r-----   1 root     disk   8,   2 24. Jan 11:13 sda2
brw-r-----   1 root     disk   8,   3 24. Jan 11:13 sda3
brw-r-----   1 root     disk   8,   4 24. Jan 11:13 sda4
crw-rw----   1 root     lp    99,   0 24. Jan 11:13 parport0
crw--w----   1 root     root   4,   0 24. Jan 11:15 tty0
crw-------   1 root     root   4,   1 24. Jan 11:14 tty1
```

Abbildung: Ausschnitt von ls -al /dev

Major - Minor Nummern

- Major Nummern werden üblicherweise statisch vergeben
- Minor Nummern werden vom Treiber verwaltet
- Jedem Treiber, der über Devicefiles angesprochen werden kann, sind Major/Minor Nummern zugeordnet

```
Character devices:
  1 mem
  4 tty
 99 ppdev

Block devices:
  1 ramdisk
  8 sd
  9 md
```

Abbildung: Ausschnitt von cat /proc/devices

ProcFS - /proc

Informationen über

- das System (Hardware)
- Prozesse
- Speicherauslastung
- einzelne Treiber
- Uptime
- Vorhandene Treiber/Module

SysFS - /sys

- ist relativ neu (erst seit Kernel 2.6)
- in allen großen Distributionen enthalten und gemountet
- sollte ProcFS für nicht prozessbezogene Daten ersetzen
- exportiert die ebenfalls neuen kobjects
- wie ProcFS ein in-Memory Filesystem

Übersicht

```c
#include <linux/kernel.h>
#include <linux/module.h>

MODULE_DESCRIPTION("WAP: Simple Driver");
MODULE_AUTHOR("Staritzbichler, Wolfersberger, Schauer");
MODULE_LICENSE("GPL");

static int wapmod_init_module(void)
{
  printk( KERN_ALERT "Module WAPMod init\n" );
  return 0;
}

static void wapmod_exit_module(void)
{
  printk( KERN_ALERT "Module WAPMod exit\n" );
}

module_init(wapmod_init_module);
module_exit(wapmod_exit_module);
```

Abbildung: Initialisierung eines Kernel Treibers

```c
#define ID_STRING .      "WAP: Ein einfacher Linux Treiber\0"
static struct proc_dir_entry *pprocdir_root  = NULL;
static struct proc_dir_entry *pprocdir_idstr = NULL;

int proc_register(void)
{
  pprocdir_root = proc_mkdir("driver/wapmod", NULL);
  pprocdir_idstr= create_proc_read_entry("id", S_IRUGO, pprocdir_root, proc_read_id, NULL);
  return 0;
}

int proc_unregister(void)
{
  remove_proc_entry("id", pprocdir_root);
  remove_proc_entry("driver/wapmod", NULL);
  return 0;
}

int proc_read_id( char *buf, char **start, off_t offset, int count, int *eof, void *data )
{
  strncpy(buf, ID_STRING, count);
}
```

Abbildung: Erzeugen eines einfachen Files in /proc

Referenzen

- Jonathan Corbet, Alessandro Rubini, and Greg Kroah-Hartman: Linux Device Drivers (das Buch als PDF auf http://lwn.net/Kernel/LDD3/)
- Die Dokumentation des Linux Kernels (Documentation Zweig des Kernels) - www.kernel.org
- Linux Magazin: http://www.linux-magazin.de/content/search?SearchText=%2B%22Kern-Technik%22