

Betriebssysteme

Fabian Stebler

2. Semester (FS 2012)

1 Einleitung

1.1 Definition

Ein Betriebssystem ist die Software die die Verwendung eines Computers ermöglicht. Es verwaltet Betriebsmittel wie den Speicher, die I/O-Gerte, usw.

1.2 Bestandteile

Betriebssysteme bestehen in der Regel aus einem Betriebssystemkern (englisch: Kernel), der die Hardware des Computers verwaltet, sowie grundlegenden Programmen, die dem Start des Betriebssystems und dessen Konfiguration dienen.

Zu den Komponenten zählen:

- Boot-Loader
- Gertetreiber
- Systemdienste
- Programmbibliotheken
- Dienstprogramme
- Anwendungen

1.3 Varianten von Betriebssystemen

- Einbenutzer- und Mehrbenutzersysteme
- Einzelprogramm- und Mehrprogrammsysteme
- Stapelverarbeitungs- und Dialogsysteme

Betriebssysteme finden sich in fast allen Computern: als Echtzeitbetriebssysteme auf Prozessrechnern, auf normalen PCs und als Mehrprozessorsysteme auf Hosts und Grossrechnern.

1.4 Geschichte

Mechanische Rechenmaschinen wurden mit der Zeit mit Lochstreifen versehen und somit konnte von einer Art Betriebssystem gesprochen werden. Später wurden die mechanischen Teile durch die Rhrentechologie und anschliessend durch Transistoren ersetzt (ca.1947).

- 1955 Erfindung Mikroprogrammierung
- 1964 Erstes modellreihenbergreifendes BS
- 1969 Beginn Arbeit an UNIX
- 1972-1974 Umschreiben UNIX in C (portabilität)
- 1980-1990 Popularitätssteigerung bei Heimcomputern
- 1981 Entwicklung erste graphische Oberfläche. Apple kauft sich mit Aktien ein, kreiert MAC und MAC OS. Verliert aber aufgrund der Experimentierfreudigkeit Marktanteile an Windows.
- 1991 Linus Torvalds beginnt mit der Entwicklung des LINUX-Kernels. (Start Open Source Bewegung)
- Microsoft entwickelte MS-DOS weiter und liefert MS-Windows 95 Mitte der 90er Jahre aus. (Tabellekalkulation)
- Im PC-Desktop-Bereich tobte ein eigentlicher Glaubenskrieg zwischen Microsoft und Apple.
- IBM und andere zogen sicher immer mehr in den Midrange/Mainframe-Markt zurück.

1.5 Lessons learned

- Die Grundkonzepte haben sich stark angenähert.
- Kompatibilität wird (oft zögernd) bereitgestellt.
- Entscheidungen für/gegen ein Betriebssystem (bes. im Privatbereich) haben teilweise weltanschauliche Hintergründe.
- Der Quellcode ist kein Geheimnis und kein Marktvorteil mehr.
- Partizipative Entwicklung durch Communities hat ein grosses Markt- und Sparpotential.
- Die Positionen scheinen bezogen, der Markt wächst immer noch stark genug, um den etablierten Anbietern Wachstum zu ermöglichen.
- Die Wertschöpfung hat sich verlagert:
 - Hardware zu Betriebssystem
 - GUI zu Applikationen
 - Daten zu Business Intelligence

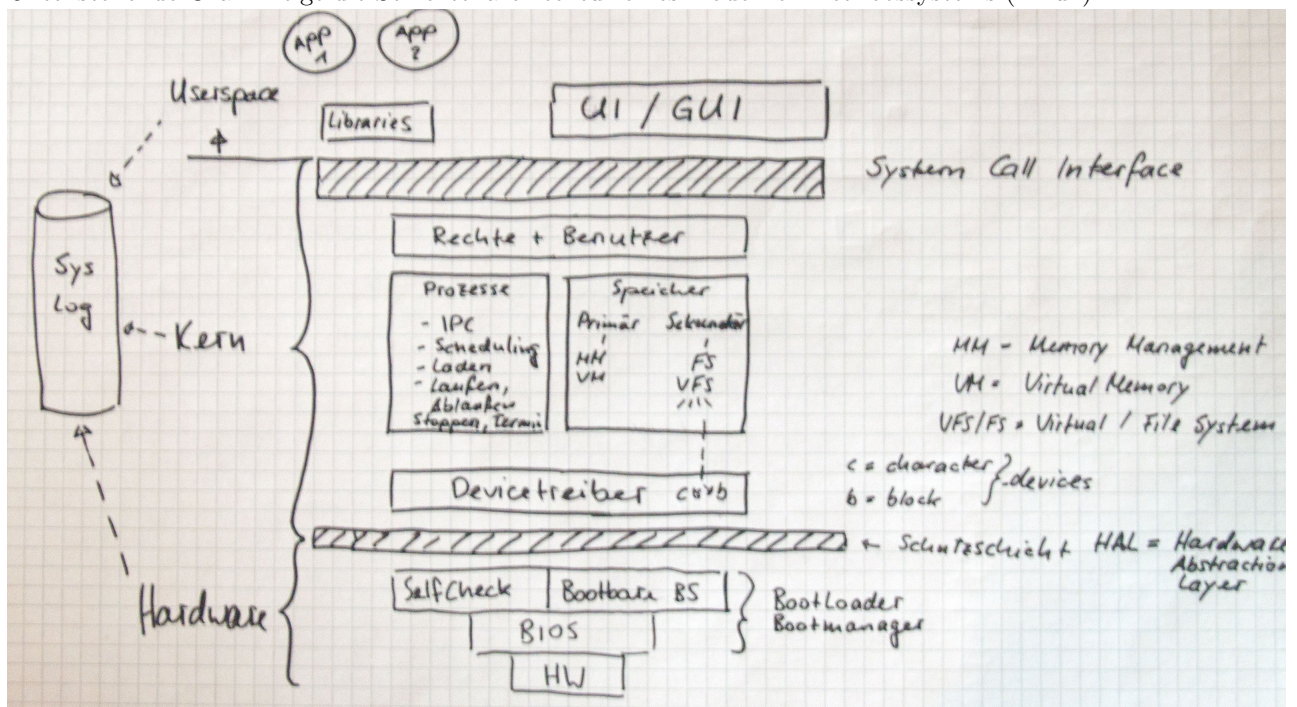
2 Blockstruktur eines Betriebssystems

2.1 Aufgaben des Betriebssystems

- Start des Systems
- Laden und Unterbrechen von Programmen (Laufzeitumgebung)
- Methoden für die Interprozesskommunikation
- Verwaltung der Prozessorzeit
- Verwaltung des primären und sekundären Speicherplatzes für das Betriebssystem und seine Anwendungen
- Verwaltung der angeschlossenen Geräte, Netzwerke etc.
- Schutz des Systemkerns und seiner Ressourcen vor nicht intendierter Benutzung
- Benutzerführung, Rollen und Rechte
- Einheitliche Schnittstelle für die System- und Anwendungsprogrammierung Ereignisprotokollierung

2.2 Graphische Darstellung

Untenstehende Grafik zeigt die Schichtenarchitektur eines modernen Betriebssystems (Linux).



2.3 Aufgabenteilung der Blöcke

2.3.1 Dateisystem

- Struktur des Dateisystems (Baum, Graph, flach, ...)
- Strukturelemente (Directories)
- Zugriffsrechte auf Directories und Dateien
- Anlage, Suche, Manipulation, Löschen von Dateien

- Verwaltung von Datenblöcken auf Speichermedien
- Kombination von Dateisystemen (mounting)
- Benutzerschnittstelle und Navigation
- Backup / Restore

2.3.2 Prozesssteuersystem

- Prozesse kreieren
- Prozesse starten
- Prozesse scheitern, Warteschlangen, Ressourcenverbrauch
- Prozesse stoppen / unterbrechen
- Prozesse terminieren (freiwillig / wegen Fehler)
- Prozesskommunikation (Prozess-Prozess und Kern-Prozess / Prozess-Kern)
- Zuordnung von Hauptspeicher und anderen geteilten Ressourcen
- Ein-/Auslagerung von Prozessen
- Prozesse und ihre Zustände anzeigen

2.3.3 System Call Interface

- Einzige Schnittstelle zwischen Kern und Benutzer
- Normierung der Syntax und Semantik (POSIX 1003)
- Parametrisierung und Übergabe
- Übergabe der Kontrolle \rightarrow Betriebsmodi

2.3.4 Programmierung

- Wahl der Programmiersprache / Systempräferenz
- System-/Applikationsnahe Bibliotheksfunktionen
- Programmierumgebung (Editor, Compiler, Assembler, Linker, Loader, Debugger)
- Bundling in einer Applikation (z.B. Eclipse)

2.3.5 Benutzerschnittstelle

- Textuelle Basis-Schnittstelle mit Kommando-Interpreter (Shell) Konsole
- Programmierbarkeit (Scripting, Pipelining, I/O-Redirection) der Benutzerschnittstelle
- Graphische Benutzerschnittstelle (GUI) mit
- Abstraktion der unterliegenden Komplexität und Syntax für Nicht-Systemspezialisten
- Austauschbarkeit der Shell und der Systembefehle (Applikationen)

2.3.6 I/O Management

Ein Betriebssystem muss auch die Hardware kontrollieren:

- Die Fähigkeiten der Hardware voll ausschöpfen.
- Die verschiedenen inhomogenen Komponenten zu einer Einheit formen.
- Die Hardware schützen vor unerlaubtem Zugriff.

Anm.: Peripheriegeräte sind meist unterschiedlich, sollten aber leicht in das System integrierbar sein.

2.3.7 File System als generelle Schnittstelle

Die Idee von Unix war, dass File System für möglichst viele Subsysteme als Schnittstelle zu verwenden.

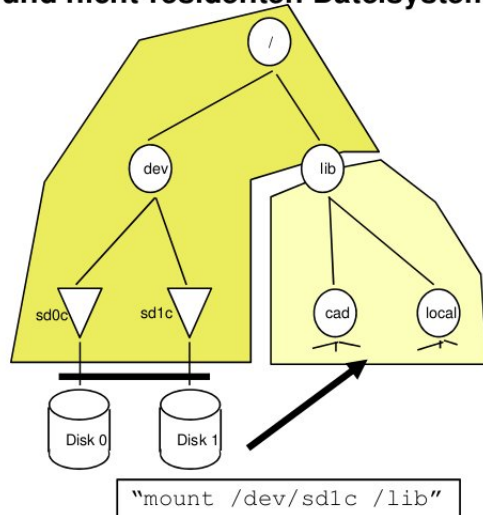
- Dateien, Directories
- Prozesssynchronisation (Lock Files, ...)
- Prozesskommunikation (Pipes, Sockets)
- Peripheriegeräte (Device Special Files)
- Kommunikationsprotokolle (TCP/IP, ...)
- Prozesse (/proc Dateisystem)

Anm.: Es bedingt eine zusätzliche Abstraktionsschicht.

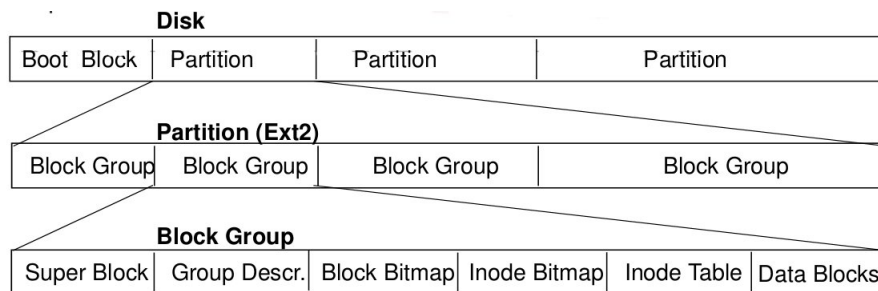
3 Filesystem

Die anschliessende Graphik zeigt ein virtuelles Dateisystem:

Virtuelles Dateisystem mit Wurzel- und nicht-residenten Dateisystemen



3.1 Partition auf der Disk



Superblockinhalt:

- Anzahl inodes und Datenblcke
- Adresse des 1. Datenblocks
- Anzahl freie Blcke und inodes
- Grsse eines Datenblocks
- Blcke / inodes pro Gruppe
- Anzahl Bytes pro inode

3.2 Blockallokationen

3.2.1 Zusammenhngende Belegung

Vorteile:

- Einfachste aller Methoden
- Sehr schneller direkter Zugriff auf die Daten

- Für die Lokalisierung der Dateiblocke brauchen wir nur Anfangsblock und Größe der Datei zu wissen.
- Lese-Operationen können sehr effizient implementiert werden.
- Gute Fehlereingrenzung

Nachteile:

- Dynamische Dateigrößen sind ein Problem
- Im Laufe der Zeit wird die Platte fragmentiert.
- Platz zu finden für neue Dateien ist ein Problem
- Verwaltung von freien Speicherplätzen notwendig
- Regelmäßige Kompaktifizierung notwendig
- Platte hin- und zurück kopieren

3.2.2 Verlinkte Blöcke

Jede Datei wird als verkettete Liste von Plattenblöcken gespeichert. Nur die Plattenadresse des ersten und letzten Blocks wird in dem Verzeichniseintrag gespeichert.

Vorteile:

- Keine externe Fragmentierung
- Sequenzieller Zugriff ist kein Problem

Nachteile:

- Schlechter wahlfreier Zugriff auf Dateiinhalte
- Jeder Verweis verursacht einen neuen Plattenzugriff
- Overhead für das Speichern der Verkettung (Lösung: clusters aus mehreren Blöcken)
- Erhöhter Aufwand bei Dateizugriffen
- Schlechte Fehlereingrenzung

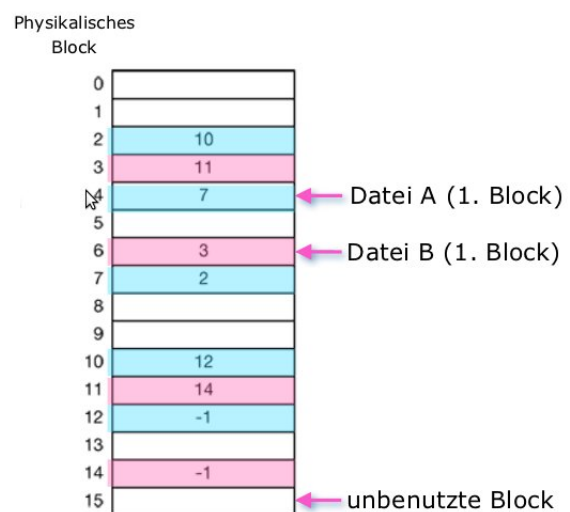
3.2.3 Filemap (Landkarte)

Vorteile:

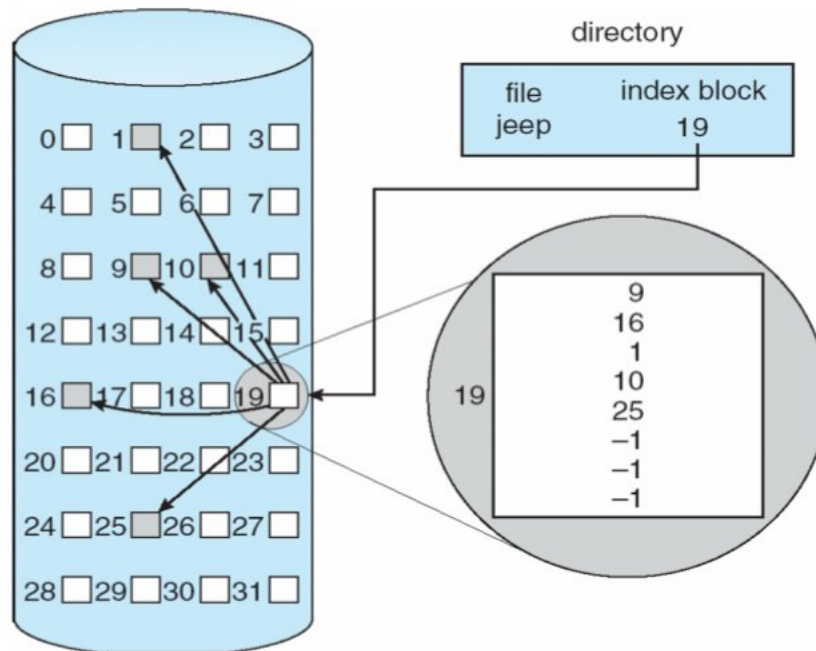
- Dateien können sehr leicht und effizient vergliedert werden

Nachteile:

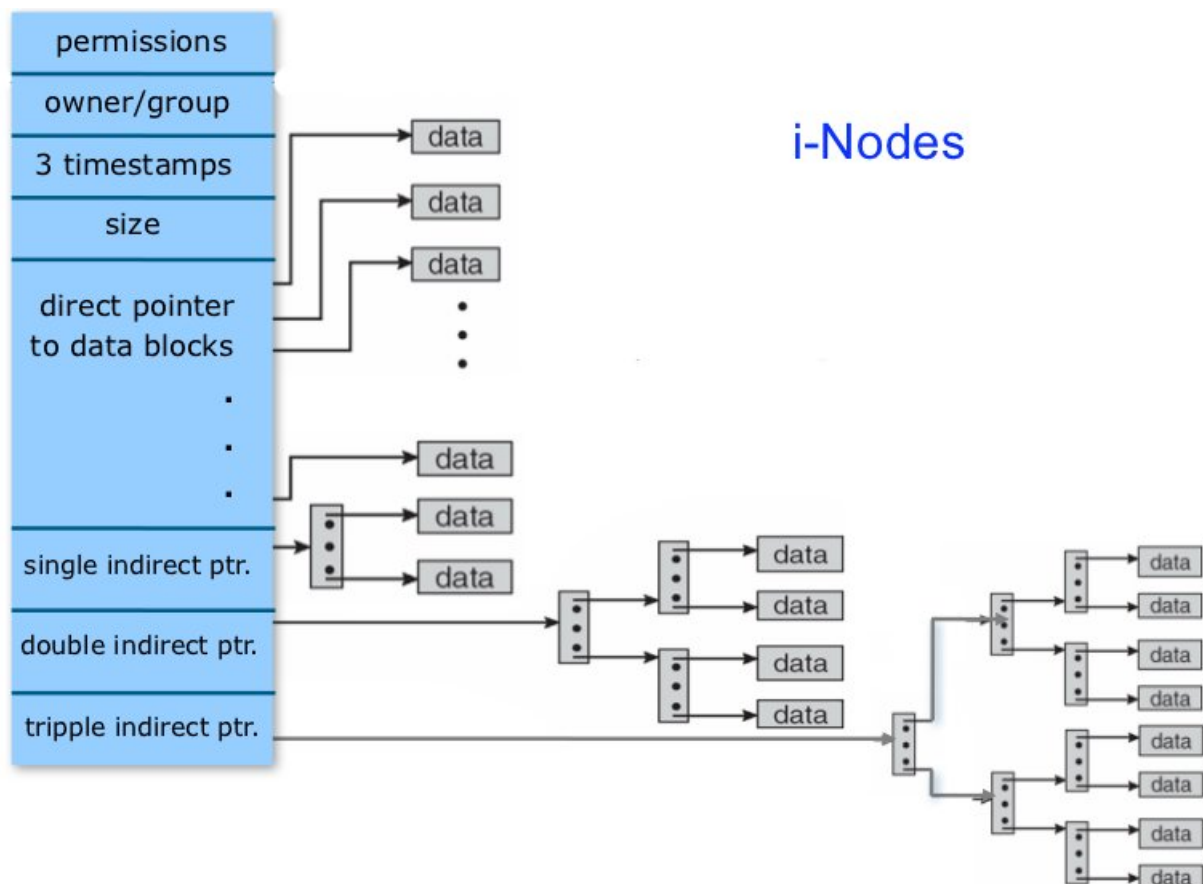
- Interne Fragmentierung
- schlecht für random accesses
- fehleranfällig



3.2.4 Index Allokation



3.3 Struktur von Inodes



3.3.1 Informationen Inode

Auf der Disk	Zustzlich im Memory
<ul style="list-style-type: none">• Inode Nummer• Anzahl hard links• Typ (-, l, d, b, c, p)• Rechte (-, r, w, x, s)• Besitzer• Gruppe• Grsse• Letzte access time• Letzte content change time• Letzte inode modification time• Datenblock Information	<ul style="list-style-type: none">• Link auf die hash list• Link auf die inode list• Benutzerzhler• Gertennummer• Device special file Indikator• Grsse eines Blocks• Anzahl Blcke• Lock auf den inode• Mount point Indikator• Warteschlange wartender Prozesse• Locks auf die Datei• Hauptspeicher-Region (fr Memory-mapped file I/O)• Belegte Seiten im Hauptspeicher

3.4 Designvorgaben fr Filesystem

- Anzahl Disks und Disk Controller
- Verteilung auf Partitionen
- Blockgrsse pro Partition
 - Grosse Blcke: schneller Zugriff auf Dateien (Datei-Durchschnittsgrsse < 4 kB)
 - Kleine Blcke: weniger interne Fragmentierung
- Anzahl Dateien / Inodes pro Partition
 - Wenige Inodes: mehr Platz fr Dateiblocke
 - Viele Inodes: mehr Dateien pro Partition mgl.