

Script generated by TTT

Title: Eingebettete_Systeme (16.10.2012)

Date: Tue Oct 16 16:10:43 CEST 2012

Duration: 60:23 min

Pages: 22



Ein Betriebssystem realisiert die Schnittstelle zwischen dem Benutzer und der physischen Rechenanlage. Aus der Sicht des Benutzers entsteht durch ein Betriebssystem eine virtuelle Maschine.

[BS-Hauptaufgaben](#)

[Systemprogrammierung](#)

[Hardwarekomponenten](#)

[Betriebsarten](#)

Generated by Targeteam

Systemprogrammierung

Die Programmierung eines Betriebssystems gehört zu dem Bereich der Systemprogrammierung.

Definition

Die **Systemprogrammierung** befasst sich mit der Darstellung, der Realisierung, den Eigenschaften und der Konstruktion derjenigen Algorithmen für ein Rechensystem, die die Bearbeitung und Ausführung von Benutzerprogrammen unter vorgegebenen Qualitätsgesichtspunkten organisieren, d.h. **steuern** und **kontrollieren**, und zum Teil selbst durchführen.

direkte Nutzung der generischen Systemprogrammierschnittstelle des BS.

meist in Programmiersprache C.

Qualitätskriterien können z.B. sein:

Zuverlässigkeit der durchgeführten Berechnung (Behandlung von Systemcrashes, Netzausfällen, fehlerhafter Nachrichtenübermittlung etc.).

Effizienz und Performanz einerseits systemglobal, d.h. es wird versucht, das System optimal auszulasten, andererseits Auftrags-lokal, z.B. es wird versucht, zu garantieren, dass eine Auftragsbearbeitung eine festgelegte Zeitdauer nicht überschreitet.

Einhaltung von **Realzeitanforderungen**: zeitkritische Aufträge besitzen z.B. eine Deadline bis zu der sie ausgeführt sein müssen.

Durchsetzung von **Sicherheitsanforderungen**: Schutz der Daten und Informationen vor unberechtigten Zugriffen und Einsichtnahme.

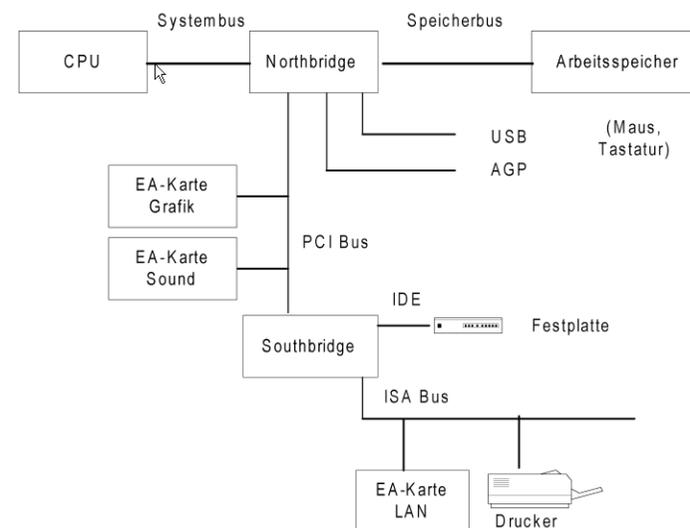
Benutzerfreundlichkeit: bequeme Formulierungsmöglichkeit von Benutzeraufträgen.

Generated by Targeteam

Hardwarekomponenten

Betriebssystem ist eng mit Hardware des Rechensystems verknüpft.

Deshalb an dieser Stelle einen kurzen Überblick über den Aufbau eines Rechensystems.



PCI = Peripheral Component Interconnect



Betriebsarten



Beim Betrieb von Rechenanlagen können bzgl. des Zusammenwirkens von Benutzer und Rechensystem die Betriebsweisen Stapelverarbeitung, Dialogbetrieb, Transaktionsbetrieb und Echtzeitbetrieb unterschieden werden.

Stapelbetrieb

Das Rechensystem verarbeitet Ströme von Auftragspaketen (engl. batch processing). Ein Benutzer deklariert vollständig alle Teile eines Auftragspaketes, bevor es in das System eingegeben wird.

Dialogbetrieb

Im Dialogbetrieb (engl. Timesharing) erteilt der Benutzer dem Betriebssystem einen Auftrag nach dem anderen im Dialog. Innerhalb eines Benutzerauftrags findet eine Interaktion zwischen dem Benutzer und der Systemumgebung statt (z.B. Eingabe weiterer Daten, Ausgabe von Zwischenergebnissen).

Transaktionsbetrieb

Bewältigung einer Vielzahl von kleinen Aufgaben in kürzester Zeit, z.B. Banküberweisungen oder Buchungen.

Echtzeitbetrieb

In der Prozesssteuerung (automatische Fertigungssysteme, Roboter) und im Multimediabereich sind die Reaktionszeiten des Betriebssystems von großer Bedeutung. Dies erfordert spezielle Mechanismen bei der Behandlung von Ereignissen und Unterbrechungen sowie der CPU-Zuteilung an rechenbereite Prozesse / Threads. Unterscheidung zwischen

harte Echtzeitsysteme: Reaktionszeit darf nicht überschritten werden.

weiche Echtzeitsysteme: gewisse Toleranzen bzgl. der Abweichung sind erlaubt.

Generated by Targeteam



Betriebssystem - Überblick



Ein Betriebssystem realisiert die Schnittstelle zwischen dem Benutzer und der physischen Rechenanlage. Aus der Sicht des Benutzers entsteht durch ein Betriebssystem eine virtuelle Maschine.

BS-Hauptaufgaben

Systemprogrammierung

Hardwarekomponenten

Betriebsarten

Generated by Targeteam



Monolithischer Ansatz

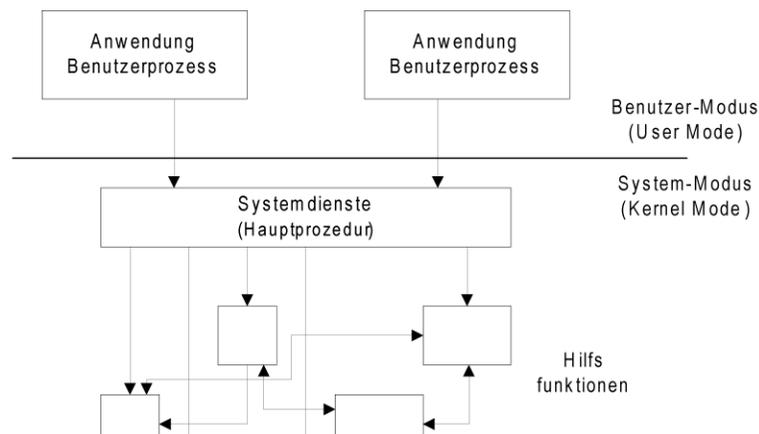


Das Betriebssystem besteht aus einer umfangreichen Menge an Funktionen, die sich bei Bedarf gegenseitig aufrufen können. Die Funktionen werden in einem großen BS-Kern zusammengefasst. Der BS-Kern wird durch Aufruf von Systemdiensten betreten.

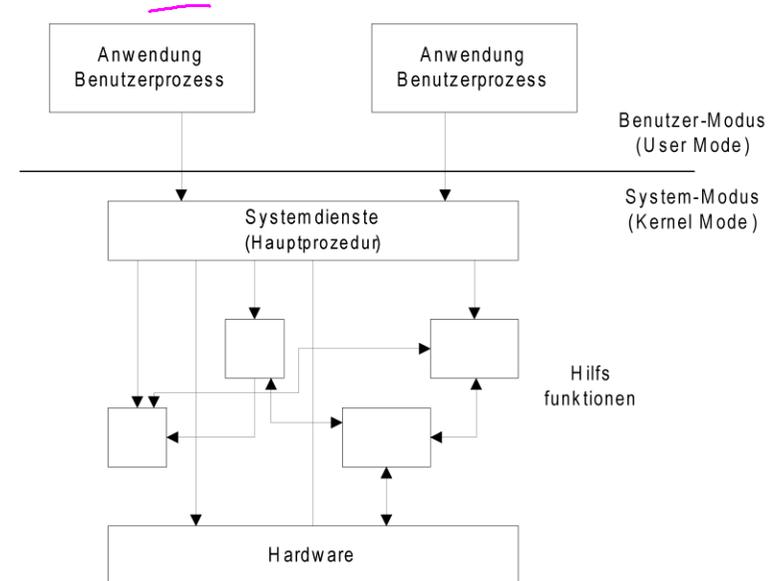
BS-Kern arbeitet im Systemmodus.

Er hat hohe Ablaufpriorität.

Er ist permanent im Arbeitsspeicher.



Monolithischer Ansatz



komplexe, monolithische Betriebssysteme sind sehr schwierig zu warten und zu erweitern.

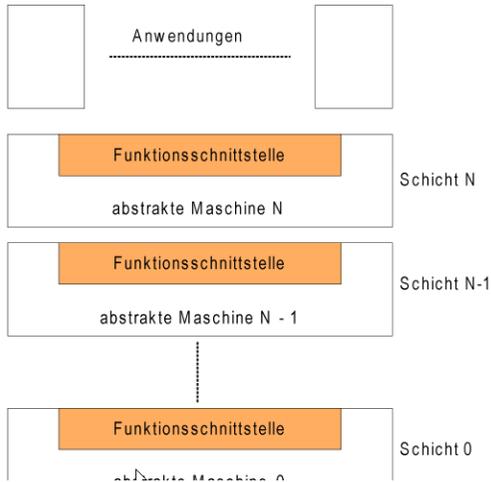
Geschichtete Systeme

Geschichtete Systeme

Einen Ausweg aus der Problematik monolithischer Systeme bieten geschichtete Systeme; das Betriebssystem besteht aus einer Hierarchie abstrakter Maschinen.

Jede Schicht hat wohldefinierte Schnittstellen und eine wohldefinierte Aufgabe

⇒ Reduktion der Systemkomplexität.



In der Praxis findet man einige verschiedene BS-Architekturkonzepte, wobei der monolithische Ansatz und zunehmend auch der Mikrokern-Ansatz am weitesten verbreitet sind.

Monolithischer Ansatz

Mikrokern-Ansatz

Beispiel: BS-Architekturen

Unix Betriebssystem

Windows NT Betriebssystem

Linux Betriebssystem

Systemaufrufe

Virtuelle Maschine

Generated by Targeteam

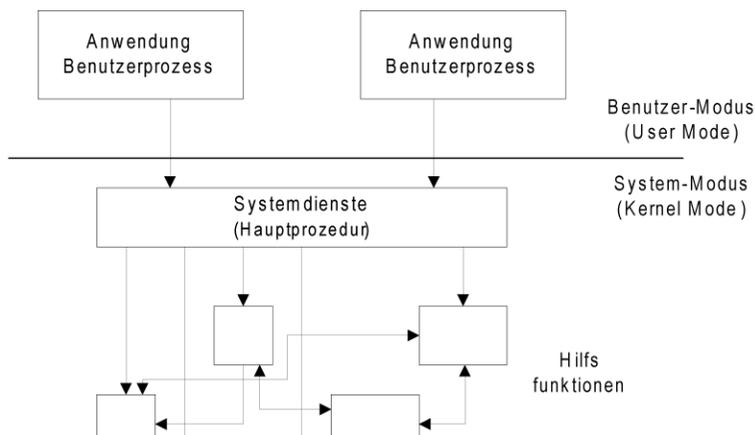
Monolithischer Ansatz

Das Betriebssystem besteht aus einer umfangreichen Menge an Funktionen, die sich bei Bedarf gegenseitig aufrufen können. Die Funktionen werden in einem großen BS-Kern zusammengefasst. Der BS-Kern wird durch Aufruf von Systemdiensten betreten.

BS-Kern arbeitet im Systemmodus.

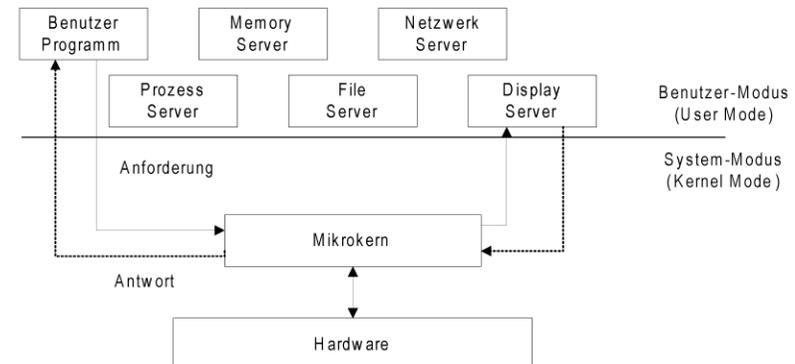
Er hat hohe Ablaufpriorität.

Er ist permanent im Arbeitsspeicher.



Im Mikrokern sind nur mehr Basismechanismen, z.B. Prozesskommunikation (Austausch von Nachrichten), CPU-Zuteilung. Möglichst viele Subsysteme sind als Systemprozesse außerhalb des Mikrokerns realisiert. Sie laufen im Benutzermodus ab, z.B. Dateisystem, Verwaltungsstrategien, Speicherverwaltung.

Einfaches Austauschen von Subsysteme ⇒ ermöglicht die einfache Anpassung von Systemanforderungen.



Generated by Targeteam

In der Praxis findet man einige verschiedene BS-Architekturkonzepte, wobei der monolithische Ansatz und zunehmend auch der Mikrokern-Ansatz am weitesten verbreitet sind.

[Monolithischer Ansatz](#)

[Mikrokern-Ansatz](#)

Beispiel: BS-Architekturen

[Unix Betriebssystem](#)

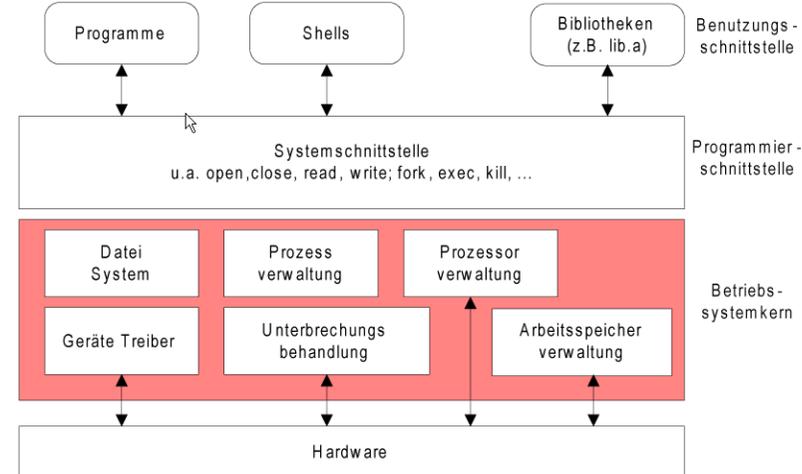
[Windows NT Betriebssystem](#)

[Linux Betriebssystem](#)

[Systemaufrufe](#)

[Virtuelle Maschine](#)

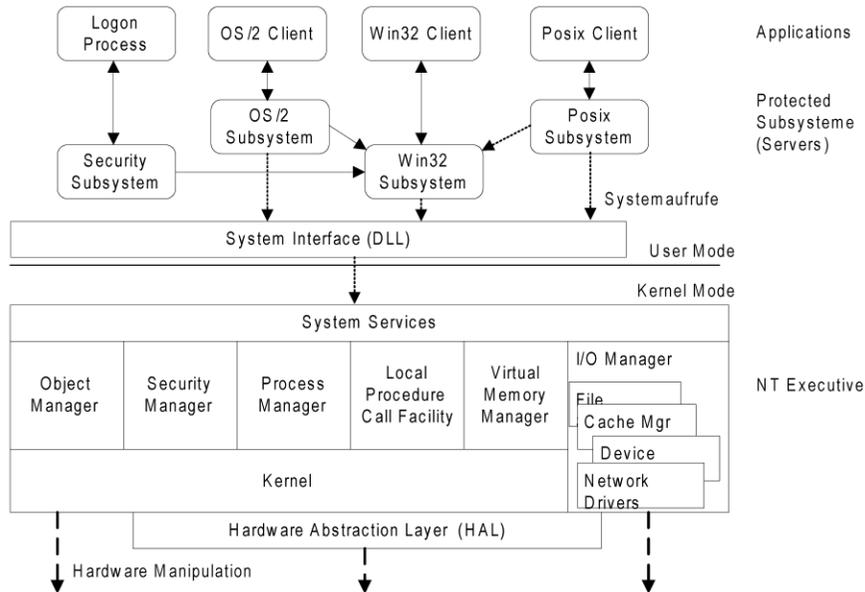
Die nachfolgende Abbildung skizziert die wesentlichen Komponenten des Unix Betriebssystems. Der Unix-BS-Kern enthält die Datei-, Prozess- und Prozessorverwaltung, die Speicherverwaltung und die Geräte-Treiber.



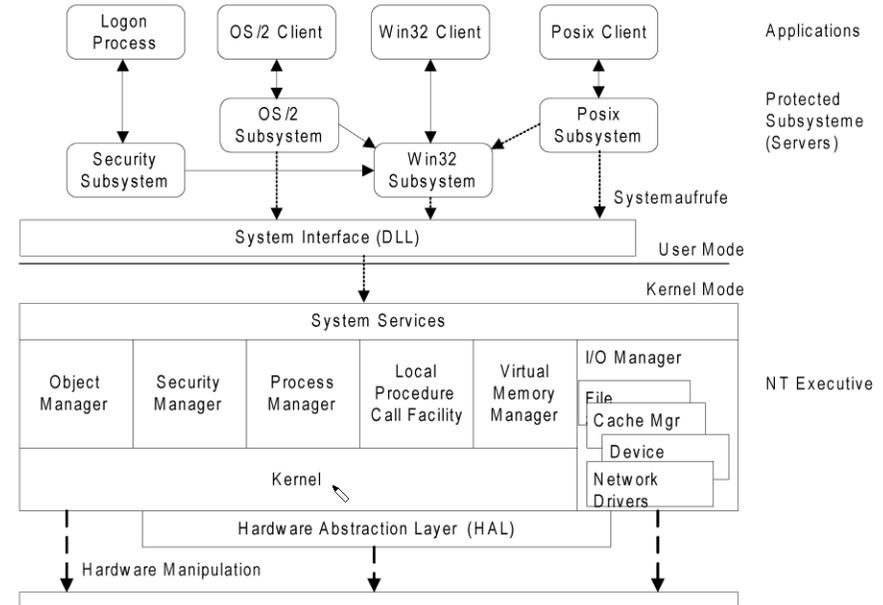
Generated by Targeteam

Generated by Targeteam

Mit Hilfe von **HAL** wird versucht, die meisten Maschinenabhängigkeiten zu verbergen. HAL präsentiert dem restlichen BS abstrakte Hardwaregeräte (z.B. Systembus, Arbeitsspeicher etc).



Mit Hilfe von **HAL** wird versucht, die meisten Maschinenabhängigkeiten zu verbergen. HAL präsentiert dem restlichen BS abstrakte Hardwaregeräte (z.B. Systembus, Arbeitsspeicher etc).



Linux begann Anfang der 1990er Jahre als eine Unix Variante für den IBM PC; erste Version durch Linus Torvalds (1991).

Linux ist frei und Quellcode ist verfügbar.

kollaborative Weiterentwicklung durch Open Source Community.

kein Mikrokern-Ansatz, jedoch modulare Struktur.

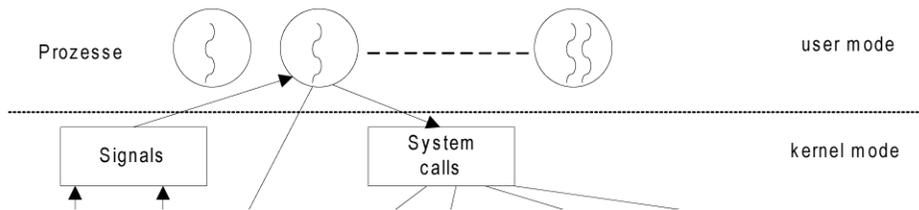
dynamic linking.

Module sind hierarchisch organisiert.

jeder Modul ist durch 2 Datenstrukturen beschrieben.

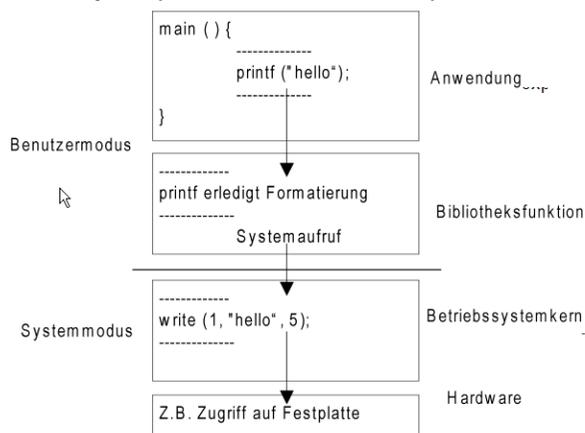
Modulbeschreibung, u.a Modulname, Größe, Zahl der exportierten Symbole und referenzierte Module.

Symbol-Tabelle.



Aufruf von BS Funktionalität durch Anwenderprogramme mittels Systemaufrufe.

in Benutzerprogrammen werden Systemaufrufe nicht direkt verwendet, sondern dies erfolgt über die Einbindung von Systembibliotheken, z.B. C-Library.

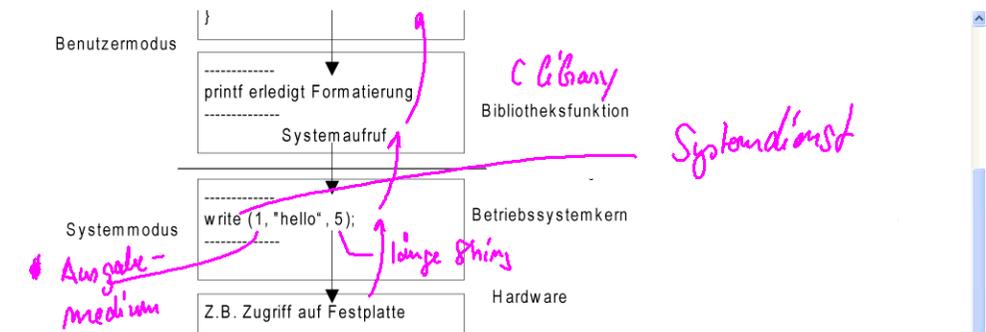
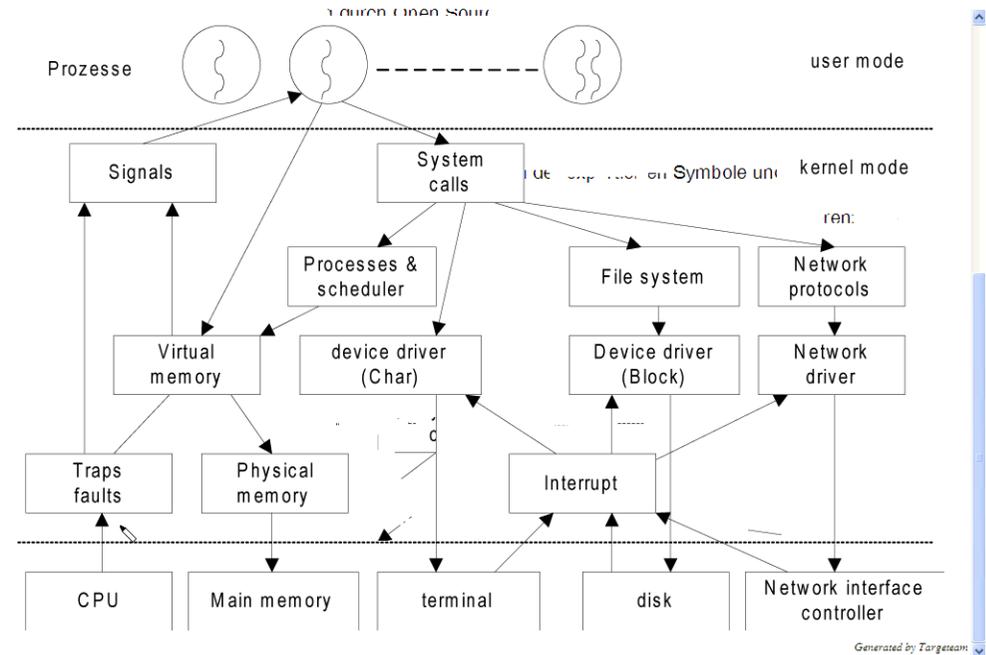


Systemaufruf führt zum Übergang vom Benutzermodus in den Systemmodus.

Beispiele von Systemaufrufen

Prozessmanagement: fork, waitpid, exit.

Dateimanagement: open, close, read, write.



Systemaufruf führt zum Übergang vom Benutzermodus in den Systemmodus.

Beispiele von Systemaufrufen

Prozessmanagement: fork, waitpid, exit.

Dateimanagement: open, close, read, write.

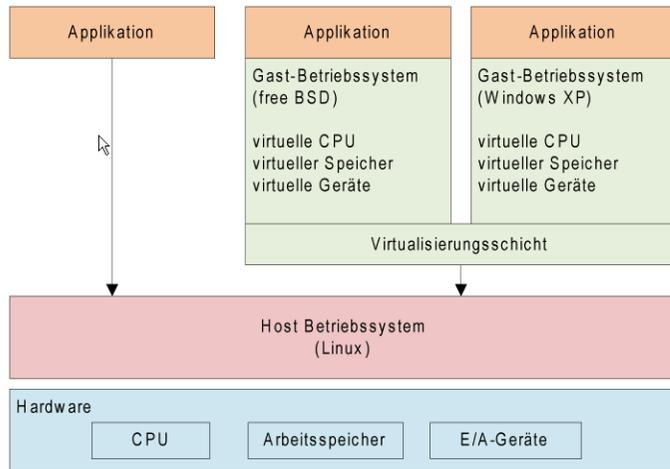
Verzeichnismangement: mkdir, rmdir, link, mount.

Gerätemangement: request/release device, get/set device attributes.

Kommunikationsmanagement: send/receive messages, create/delete connection.

Virtualisierungskonzept erlaubt die gleichzeitige Bereitstellung mehrerer Betriebssysteme auf einem Rechner.
 Isolierung der virtuellen Maschinen.
 gemeinsame Nutzung von Dateien möglich.

Beispiel **VMware**.



In der Praxis findet man einige verschiedene BS-Architekturkonzepte, wobei der monolithische Ansatz und zunehmend auch der Mikrokern-Ansatz am weitesten verbreitet sind.

[Monolithischer Ansatz](#)

[Mikrokern-Ansatz](#)

Beispiel: BS-Architekturen

[Unix Betriebssystem](#)

[Windows NT Betriebssystem](#)

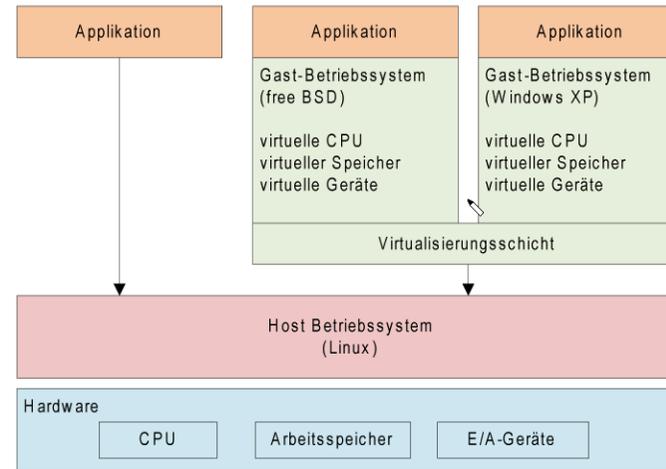
[Linux Betriebssystem](#)

[Systemaufrufe](#)

[Virtuelle Maschine](#)

virtualisierungskonzept erlaubt die gleichzeitige Bereitstellung mehrerer Betriebssysteme auf einem Rechner.
 Isolierung der virtuellen Maschinen.
 gemeinsame Nutzung von Dateien möglich.

Beispiel **VMware**.



Java Virtual Machine (JVM): abstrakter Computer zur Ausführung von Java-Programmen; implementiert in