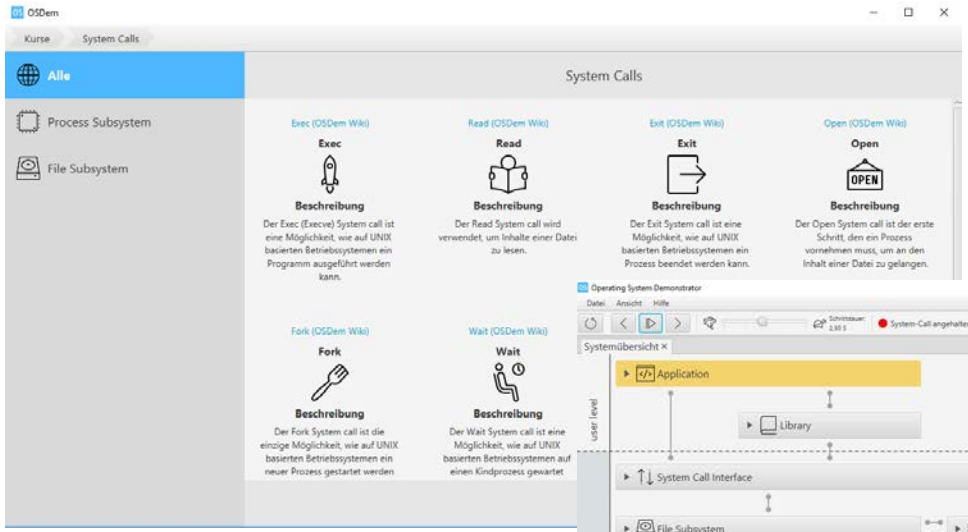
Fachhochschule Nordwestschweiz
Hochschule für Technik

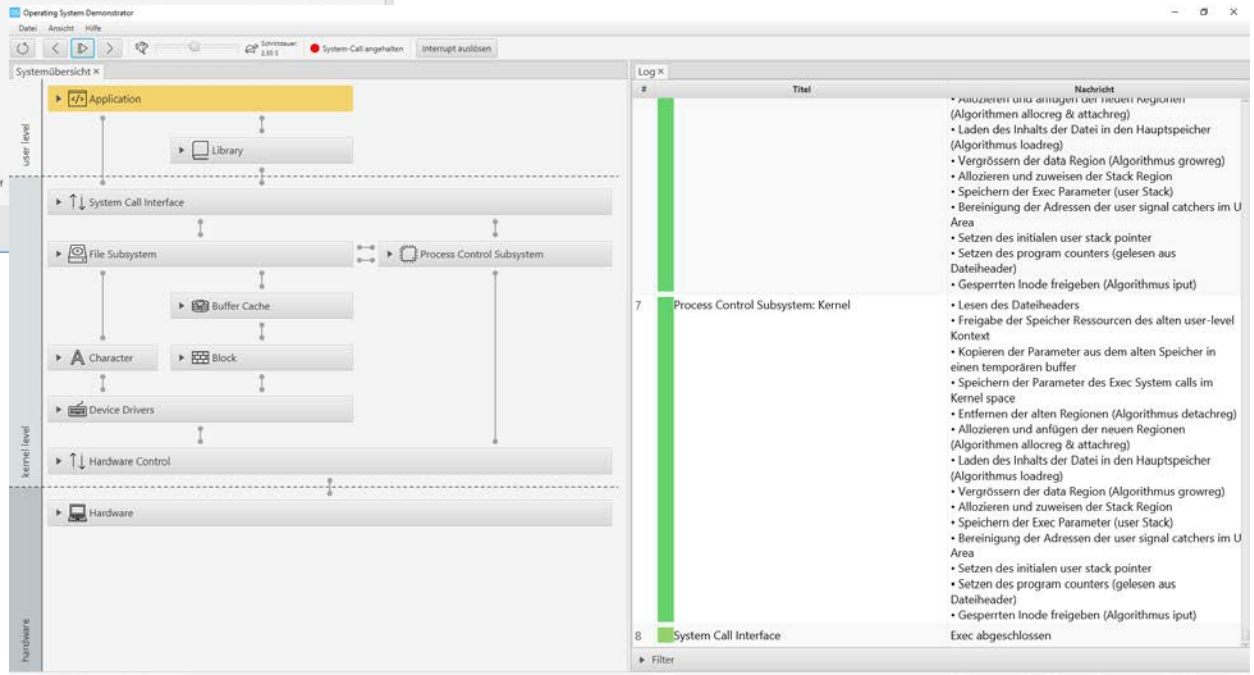
Modul Betriebssysteme (bsys-iC)

Feedback aus der Hausaufgabe

- Was ist Ihnen aufgefallen?
- Gab es grundlegende neue Erkenntnisse?
- Was hat gefehlt?
- Wieviel Zeit haben Sie aufgewendet?

OSDem


OSDem
Kurse
System Calls
Alle
Process Subsystem
File Subsystem
System Calls
Exec (OSDem Wiki)
Exec
Beschreibung
Der Exec (Execve) System call ist eine Möglichkeit, wie auf UNIX basierten Betriebssystemen ein Programm ausgeführt werden kann.
Read (OSDem Wiki)
Read
Beschreibung
Der Read System call wird verwendet, um Inhalte einer Datei zu lesen.
Exit (OSDem Wiki)
Exit
Beschreibung
Der Exit System call ist eine Möglichkeit, wie auf UNIX basierten Betriebssystemen ein Prozess beendet werden kann.
Open (OSDem Wiki)
Open
OPEN
Beschreibung
Der Open System call ist der erste Schritt, den ein Prozess vornehmen muss, um an den Inhalt einer Datei zu gelangen.
Fork (OSDem Wiki)
Fork
Beschreibung
Der Fork System call ist die einzige Möglichkeit, wie auf UNIX basierten Betriebssystemen ein neuer Prozess gestartet werden
Wait (OSDem Wiki)
Wait
Beschreibung
Der Wait System call ist eine Möglichkeit, wie auf UNIX basierten Betriebssystemen auf einen Kindprozess gewartet


Operating System Demonstrator
Datei
Ansicht
Hilfe
System-Call angehalten
Interrupt auslösen
Systemübersicht
Application
Library
System Call Interface
File Subsystem
Process Control Subsystem
Buffer Cache
Character
Block
Device Drivers
Hardware Control
Hardware
user level
kernel level
hardware
Log
Titel
Nachricht
7
Process Control Subsystem: Kernel
• Lesen des Dateiheaders
• Freigabe der Speicher Ressourcen des alten user-level Kontext
• Kopieren der Parameter aus dem alten Speicher in einen temporären buffer
• Speichern der Parameter des Exec System calls im Kernel space
• Entfernen der alten Regionen (Algorithmus detachreg)
• Allozieren und anfügen der neuen Regionen (Algorithmen allocreg & attachreg)
• Laden des Inhalts der Datei in den Hauptspeicher (Algorithmus loadreg)
• Vergrössern der data Region (Algorithmus growreg)
• Allozieren und zuweisen der Stack Region
• Speichern der Exec Parameter (user Stack)
• Bereinigung der Adressen der user signal catchers im U Area
• Setzen des initialen user stack pointer
• Setzen des program counters (gelesen aus Dateiheader)
• Gesperrten Inode freigeben (Algorithmus iput)
8
System Call Interface
Exec abgeschlossen
Filter

Lektion 2: Aufbau & Blockstruktur eines Betriebssystems, Abgrenzung zu anderen HW/SW-Komponenten

Inhalt

- Die Blockstruktur eines Betriebssystems und Aufgabenzuordnung zu funktionalen Blöcken
- Anforderungen & Aufgabenteilung zwischen Betriebssystem und Hardware / Peripherie
- Anforderungen & Aufgabenteilung zwischen Betriebssystem und darauf aufbauender Middleware, Datenbanken und Applikationen

Motivation

Anforderungen an das Betriebssystem

- 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 & Rechte
- Einheitliche Schnittstelle für die System- & Anwendungsprogram-mierung
- Ereignisprotokollierung

Logische Blockstruktur

Aufgabenzuordnung zu funktionalen Blöcken

- Dateisystem
  - Struktur des Dateisystems (Baum, Graph, flach, …)
  - Strukturelemente (Directories)
  - Zugriffsrechte auf Directories & Dateien
  - Anlage, Suche, Manipulation, Löschen von Dateien
  - Verwaltung von Datenblöcken auf Speichermedien
  - Kombination von Dateisystemen (mounting)
  - Benutzerschnittstelle und Navigation
  - Backup / Restore

Aufgabenzuordnung zu funktionalen Blöcken

- Prozess-Steuersystem (System- und Benutzer-Prozesse bzw. Threads)
  - Prozesse kreieren
  - Prozesse starten
  - Prozesse schedulen, 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

Aufgabenzuordnung zu funktionalen Blöcken

- System Call Interface
  - Einzige Schnittstelle zwischen Kern und Benutzer
  - Normierung der Syntax & Semantik (POSIX 1003)
  - Parametrisierung und Übergabe
  - Übergabe der Kontrolle → Betriebsmodi

Aufgabenzuordnung zu funktionalen Blöcken

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

Aufgabenzuordnung zu funktionalen Blöcken

- 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 & Syntax für Nicht-Systemspezialisten
  - Austauschbarkeit der Shell und der System-befehle (Applikationen)

Betriebssystem und Hardware / Peripherie

- Das Betriebssystem muss:
  - die Fähigkeiten der Hardware optimal ausnutzen (z.B. Mehrprozessor-Architektur)
  - die Komplexität und Inhomogenität der Hardware und Peripherie verbergen (z.B. einheitlicher Systemaufruf für Daten Lesen oder Schreiben)
  - die Hardware und Peripherie vor unerlaubter Nutzung (Durchgriff auf die Hardwaresteuerung) schützen
- Peripheriegeräte sind per Definition unterschiedlich, sollen aber möglichst einfach und einheitlich in das System integrierbar sein.

Das Filesystem als generelle Schnittstelle

- Die Idee von Unix (inzwischen auch in anderen Betriebssystemen verwendet):
  - Das Dateisystem ist die einheitliche Schnittstelle für möglichst viele (alle) Subsysteme:
    - Dateien, Directories
    - Prozesssynchronisation (Lock Files, …)
    - Prozesskommunikation (Pipes, Sockets)
    - Peripheriegeräte (Device Special Files)
    - Kommunikationsprotokolle (TCP/IP, …)
    - Prozesse (/proc Dateisystem)
  - Bedingt eine zusätzliche Abstraktionsschicht

Betriebssystem und darauf aufbauende Komponenten

- Middleware
  - Aus Sicht des Betriebssystems sind Middleware-Komponenten Applikationselemente, d.h. sie laufen primär im „user space“ ab und machen Systemaufrufe wie andere Applikationen auch.
  - Intern können Middleware-Umgebungen (Java, .NET usw.) betriebssystem-typische Aufgaben (Ressourcenverwaltung, Scheduling usw.) wahrnehmen (jedoch innerhalb des Prozess-Adressraums).
- Datenbanken
  - Meist ebenfalls im „user space“, jedoch oft mit speziellen, intern geschützten Zugriffen auf das Dateisystem, Hauptspeicher etc.
- Applikationen
  - Laufen im „user space“, Systemaufrufe wo nötig, Scheduling

Pause

Übung
(ca. 30 min.)

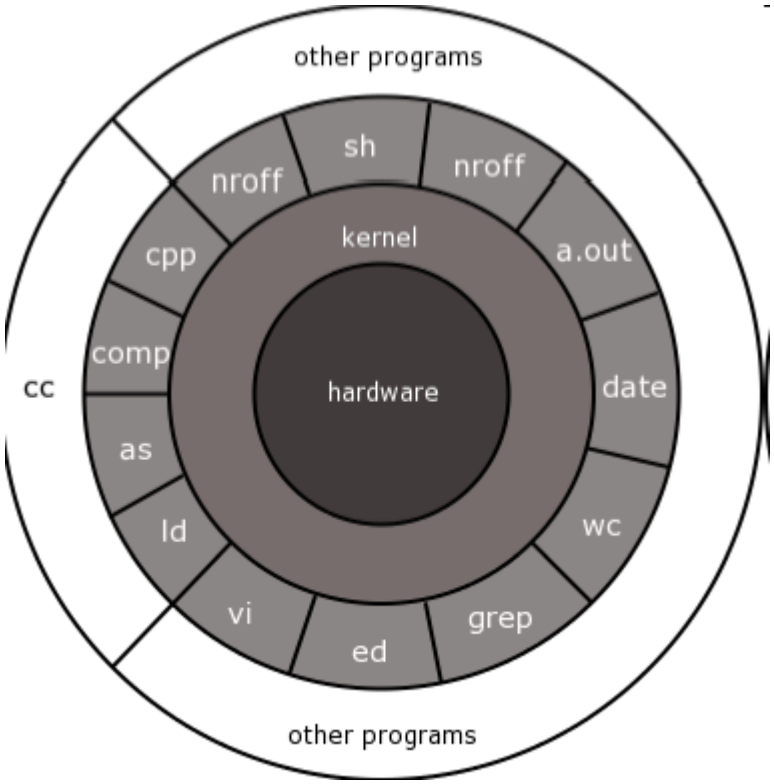
- Aufgabe(n) gemäss separatem Aufgabenblatt
- Lösungsansatz: Einzelarbeit oder Gruppen von max. 3 Personen
- Hilfsmittel: beliebig
- Besprechung möglicher Lösungen in der Klasse (es gibt meist nicht die eine «Musterlösung»)

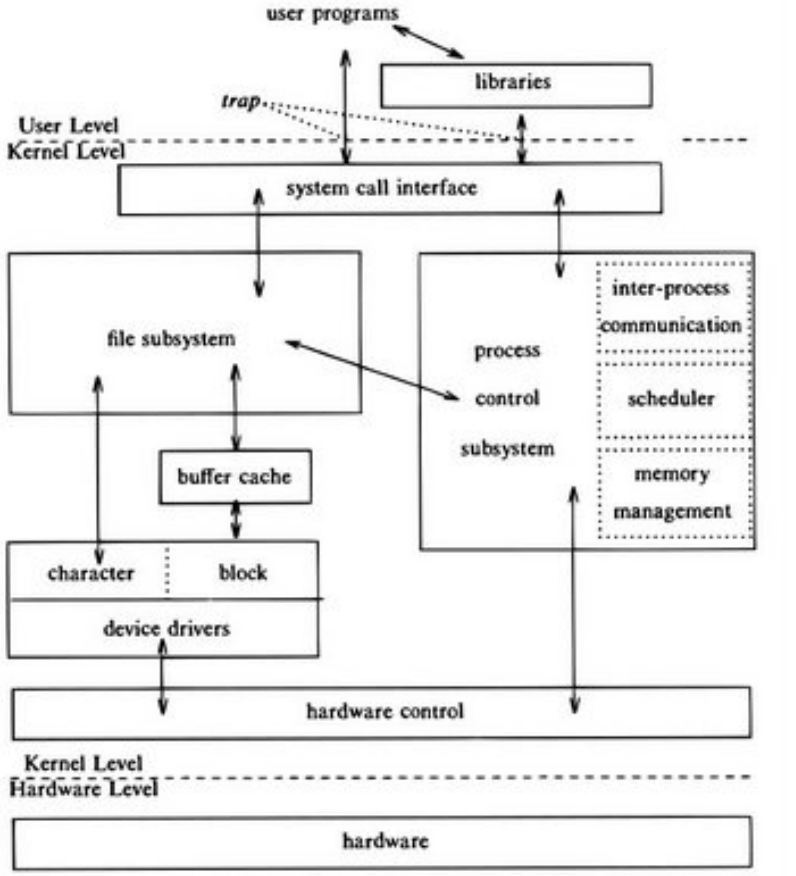
Übungsbesprechung (ca. 15 min.)

- Stellen Sie Ihre jeweilige Lösung der Klasse vor.
- Zeigen Sie auf, warum ihre Lösung korrekt, vollständig und effizient ist.
- Diskutieren Sie ggf. Design-Entscheide, Alternativen oder abweichende Lösungsansätze.
- Gibt es Unklarheiten? Stellen Sie Fragen.

Pause

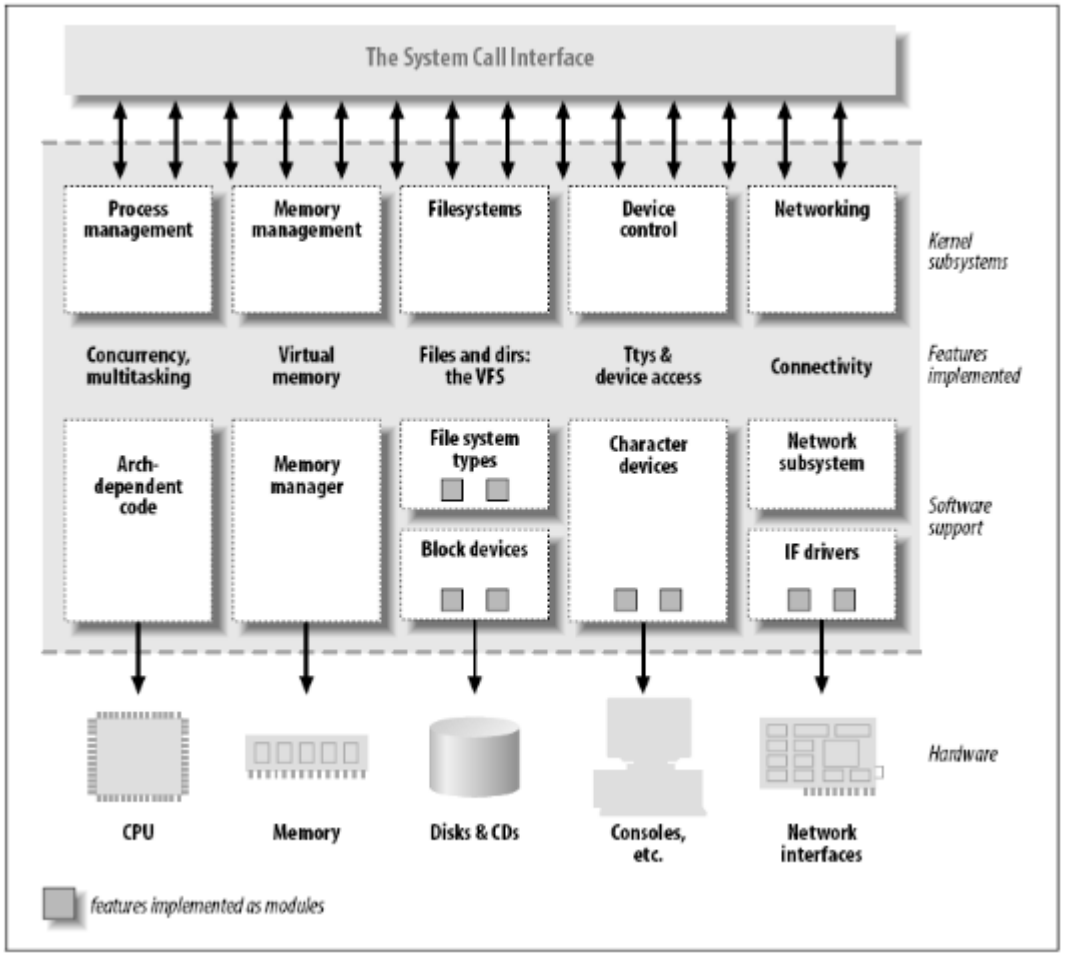
Die Unix-Schichtenarchitektur



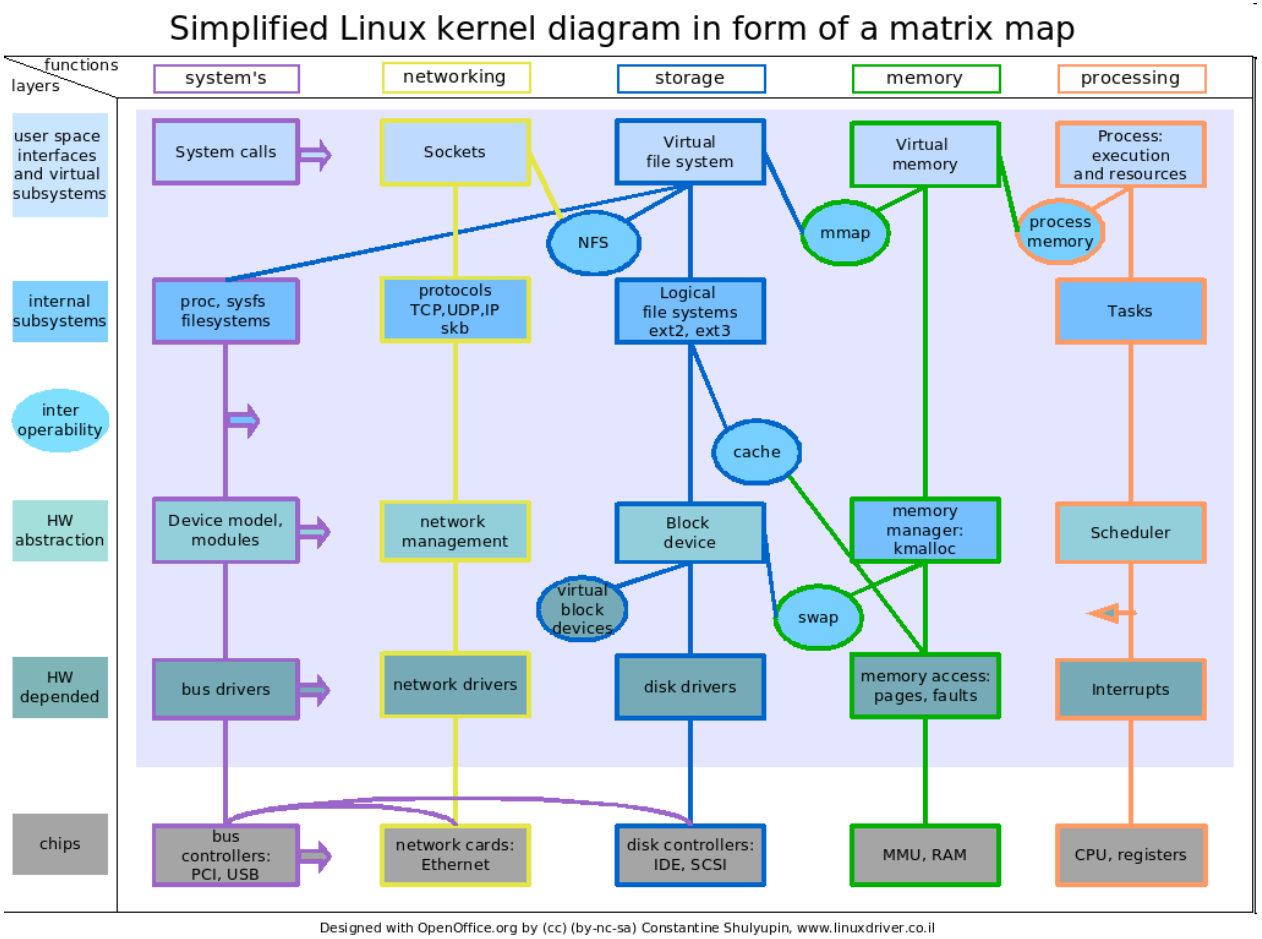


Maurice Bach: The Design of the Unix Operating System

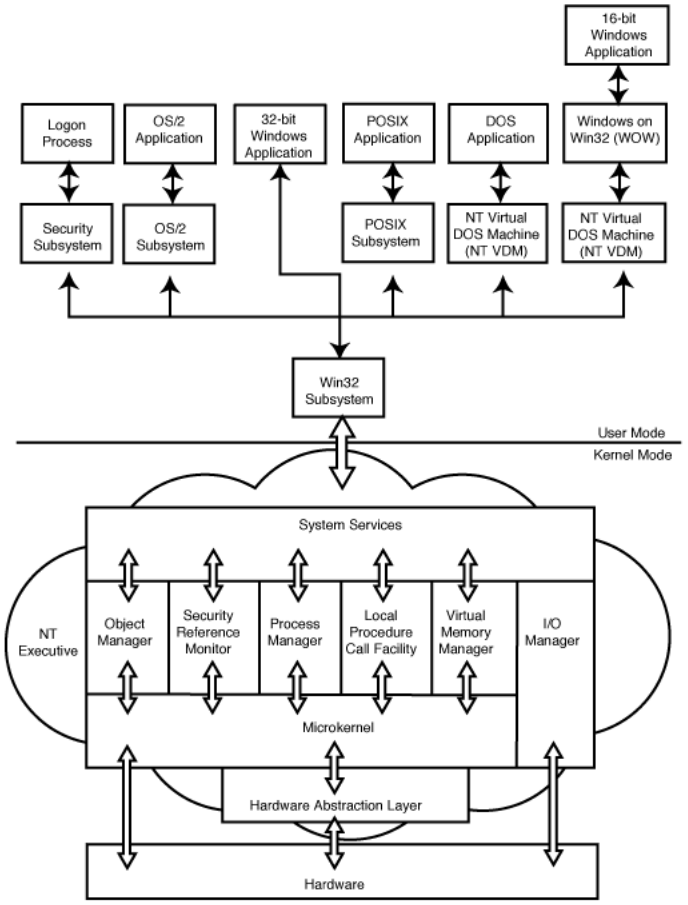
Die Linux-Schichtenarchitektur



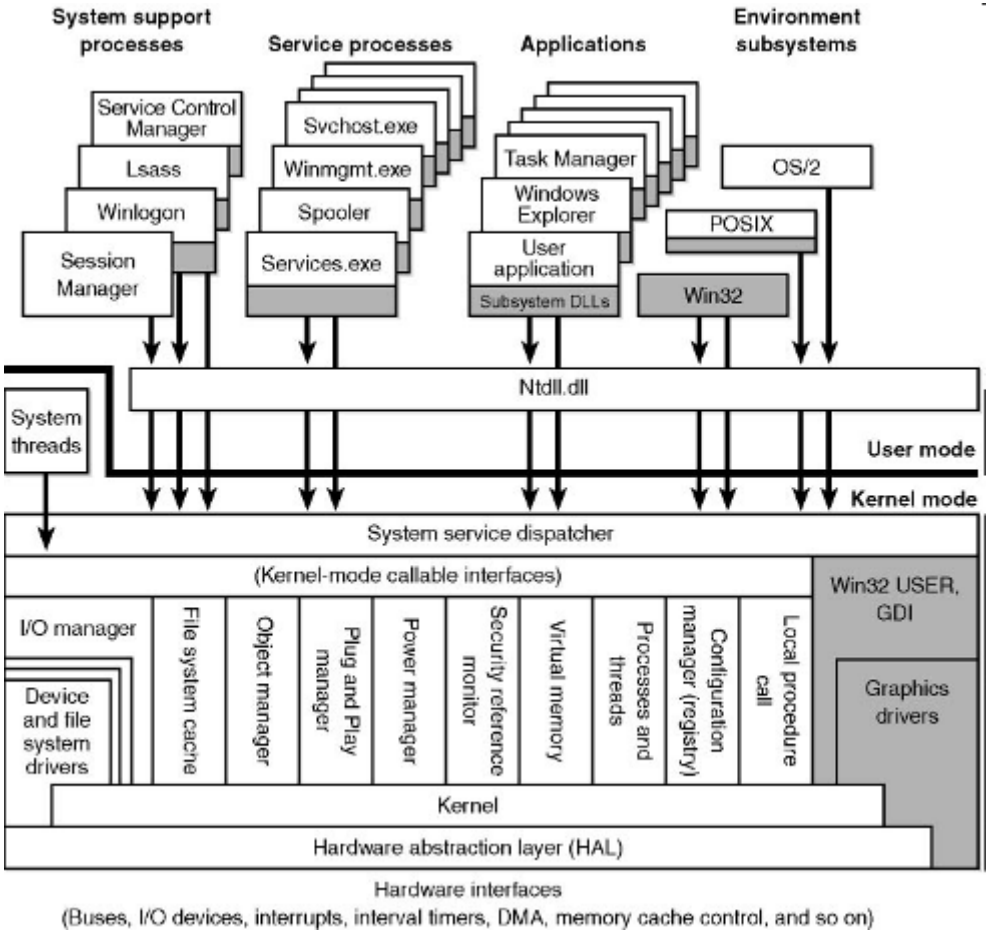
Die Linux Kernel-Matrix



Die Microsoft Windows-Schichtenarchitektur

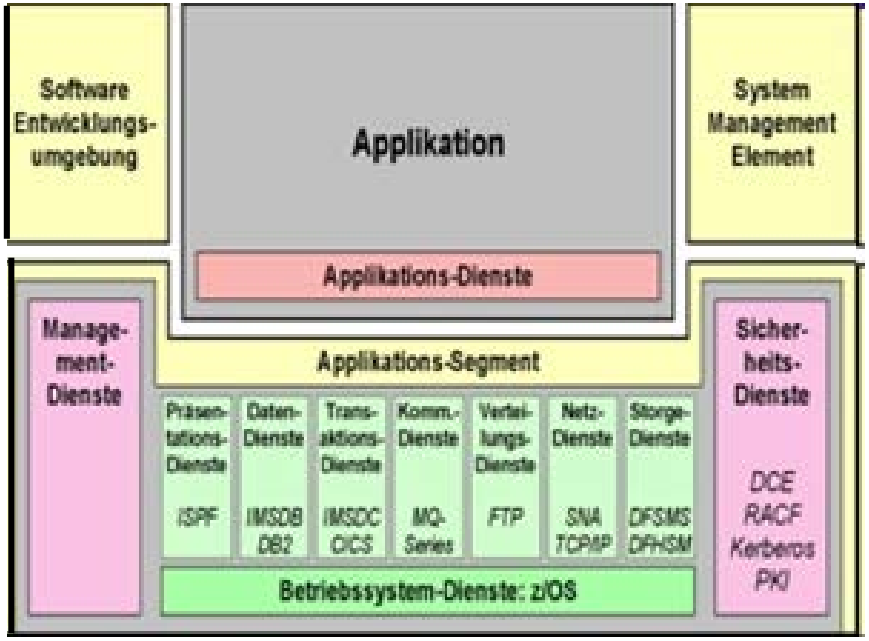


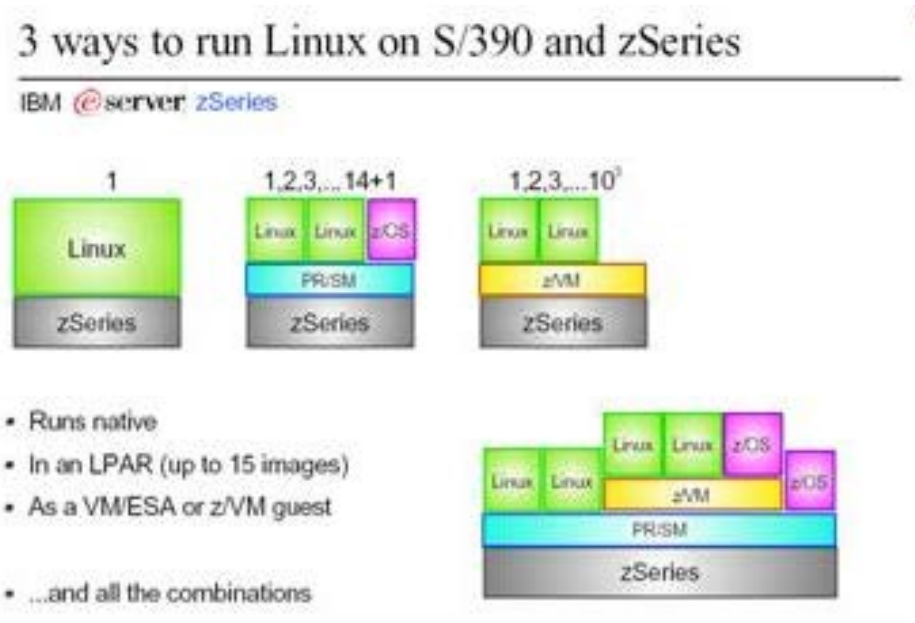
Win NT: http://www.softlookup.com/tutorial/winnt/02fig04.gif



http://p.blog.csdn.net/images/p_blog_csdn_net/hongmy525/339986/o_WinNtKernel2-3.bmp

Die IBM z/OS-Schichtenarchitektur





http://www.bsi.bund.de/gshb/deutsch/bilder/baust/db03107_01_e07.jpg

http://www.linuxfocus.org/Deutsch/March2004/article328.shtml

Zusammenfassung der Lektion 2 und Hausaufgabe

- Die Blockstruktur eines Betriebssystems und Zuordnung funktionale Elemente zu den einzelnen Blöcken.
- Abgrenzung und Aufgabenteilung zwischen Betriebssystem, Hardware, Peripherie, Middleware, Datenbanken und Applikationen.
- Hausaufgabe:
  - Repetieren Sie den Stoff dieser Lektion.
  - Studieren Sie die Webseite „http://en.wikipedia.org/wiki/Unix“.
  - Beschäftigen Sie sich mit dem Demonstrator „OSDem“