



Bild von Mike by Pexels

Betriebssysteme | I.1



Franz J. Hauck | Institut für Verteilte Systeme, Univ. Ulm



Bild von Mike by Pexels

I | Virtualisierung Betriebssysteme



Franz J. Hauck | Institut für Verteilte Systeme, Univ. Ulm

Überblick

Überblick der Themenabschnitte

- **A** – Organisatorisches
- **B** – Zahlendarstellung und Rechnerarithmetik
- **C** – Aufbau eines Rechnersystems
- **D** – Einführung in Betriebssysteme
- **E** – Prozessverwaltung und Nebenläufigkeit
- **F** – Dateiverwaltung
- **G** – Speicherverwaltung
- **H** – Ein-, Ausgabe und Geräteverwaltung
- **I** – Virtualisierung
- **J** – Verklemmungen
- **K** – Rechteverwaltung



Inhaltsüberblick

Virtualisierung

- Begriff
- virtuelle Maschine
- virtuelles Betriebssystem
- Mechanismen für Virtualisierung

Was ist Virtualisierung?

Etwas existiert gar nicht so wie es erscheint

- z.B. virtuelle Abbilder (Optik)
- z.B. Virtuelle Realität (Informatik)
- z.B. virtueller Speicher (Betriebssystem)
 - Adressen und Speichersegmente weichen von den realen ab
- u.v.m.

Fokus hier: virtuelle Maschinen, virtuelle Betriebssysteme

Virtuelle Maschinen

Virtualisierung der realen Hardware

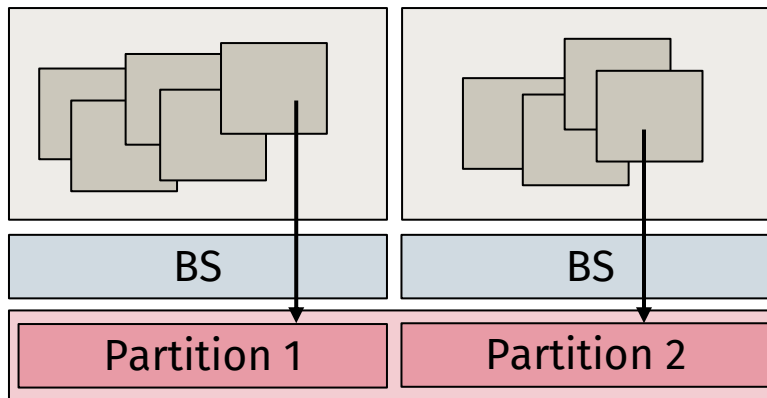
■ Motivation

- **mehrere** virtuelle Maschinen auf einer realen Maschine
 - bessere Ressourcennutzung (z.B. Cloud Computing)
 - verschiedene Betriebssysteme (Linux vs. Windows vs. iOS)
- **Isolation** verschiedener Anwendungen oder Kunden
 - virtuelle Maschinen sind voneinander entkoppelt

Hardware-Partitionierung

Aufteilung der Hardware für mehrere virtuelle Maschinen

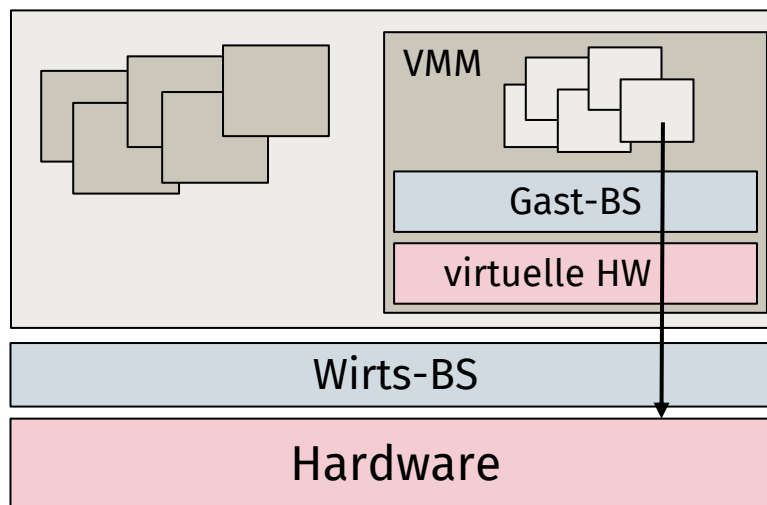
- Vorkehrungen in der Hardware
 - dynamisch im Betrieb möglich
 - typisch: teure **Mainframe-Systeme**



Virtual-Machine-Monitor

Eine virtuelle Maschine als Anwendung

- Anwendungsprozess enthält komplette virtuelle Maschine
 - vollständig virtualisierte Hardware für das Gastbetriebssystem



- Beispiele: VMware Server, MS Virtual PC, Virtual Box
- VMM wird auch Typ-2-Hypervisor oder Typ-2-VMM genannt

Virtual-Machine-Monitor (2)

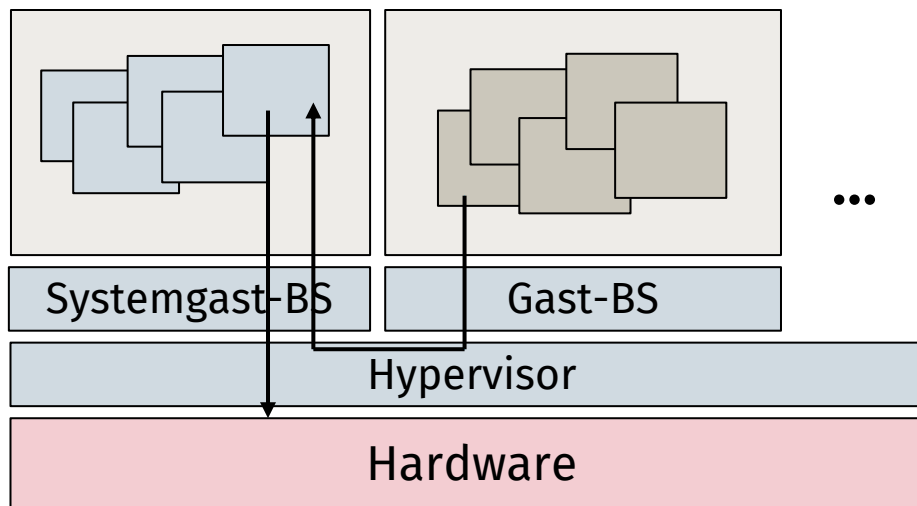
Aspekte

- Gastbetriebssystem und Anwendungen laufen unverändert
 - jedoch nur innerhalb des VMM
- VMM kann andere als real vorhandene Hardware abbilden
 - genannt **Emulation**
 - z.B. C64, Amiga, Nintendo, PowerPC auf x86-Hardware, Apple Rosetta (x86 auf ARM)
- teilweise Emulationen bei Virtualisierung
 - Hardwarebausteine, die nicht durch mehrere Gast-BS verwendbar sind
 - z.B. Emulation einer Standard-Netzwerkkarte und Abbildung auf eingebaute Karte

Paravirtualisierung

Hypervisor als Vermittlungsschicht

- Multiplexing ermöglicht Virtualisierung
 - nur APIs für Gastbetriebssysteme durch Hypervisor
 - Hypervisor als Metabetriebssystem



- Beispiele: Xen, Citrix, VMware ESX Server

Paravirtualisierung (2)

Gastbetriebssysteme angepasst

- statt direkter HW-Zugriff erfolgt **Hypervisor-Aufruf**
 - ähnlich wie Systemaufruf
 - Hardware-Zugriffe über spezielles System-Gastbetriebssystem
 - z.B. **Xen: Domain0**
- Hypervisor **verwaltet** Hardware-Ressourcen direkt
 - CPU-Scheduling
 - Speicher
 - wird auch Typ-1-Hypervisor oder Typ-1-VMM genannt

Paravirtualisierung (3)

Vor- und Nachteile gegenüber VMM

◆ Vorteil

- effizienter, da auf Virtualisierung optimiert und Emulationen vermieden

◆ Nachteil

- Anpassung des BS notwendig
- z.B. eigene Kernmodule

Hardware-Virtualisierung

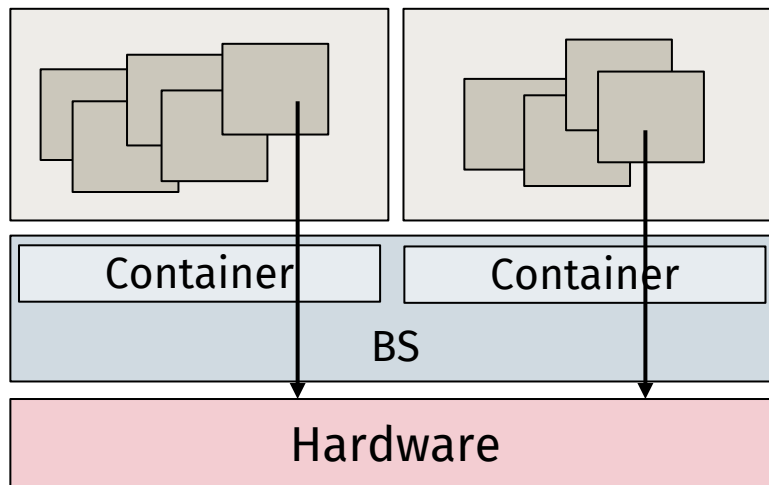
Reale Systeme

- häufig **Mischung** aus VMM und Paravirtualisierung
 - z.B. **KVM/QEMU** ist Typ-2-VMM mit Kernelmodul und Anpassung im Gast-BS
 - z.B. **Xen** ist Typ-1-Hypervisor mit zusätzlicher Hardware-Virtualisierung

Betriebssystemvirtualisierung

Aufteilung des Betriebssystems in Partitionen

- voneinander isolierte Anwendungsdomänen



- Beispiele: BSD Jails, Solaris Container, Linux Vserver, OpenVZ
- bekannter Container-Manager: Docker

Betriebssystemvirtualisierung (2)

Vor- und Nachteile gegenüber Paravirtualisierung

◆ Vorteil

- noch effizienter als Paravirtualisierung
- weniger Overhead

◆ Nachteil

- nur ein einziges Betriebssystem verwendbar
- geringere Isolation



Fallbeispiel: Docker

Betriebssystemvirtualisierung mit Linux

- unter **Windows** und **MacOS** mit Unterstützung eines VMM
- Docker **Image**
 - Datei beinhaltet komplette Software für einen Container
 - Software-Installation und –Konfiguration
 - erstellt anhand einer Beschreibung – **Dockerfile**
 - häufig auf Basis bestehender Images
 - evtl. gespeichert in einer Docker **Registry**
 - (zentrale) Verteilstelle für Images



Fallbeispiel: Docker (2)

Betriebssystemvirtualisierung mit Linux

■ Start eines Containers

- Angabe des **Image-Namens**
- Docker ermittelt das Image von einer Registry (konfigurierbar)
 - lädt Image in lokales Dateisystem
- Docker startet Container
 - **einmalige Ausführung** eines Kommandos oder
 - Start eines **Dienstes**



Fallbeispiel: Docker (3)

Betriebssystemvirtualisierung mit Linux

- Betrieb eines Containers
 - Weiterleitung von Kommunikation vom Host an den Container
 - Port-Forwarding
 - virtuelle Netzwerke zwischen Containern konfigurierbar
 - Betriebssystemvirtualisierung in virtuellen Maschinen
 - Einsatz im Cloud-Computing
 - Cloud-Provider vermieten virtuelle Maschinen (IaaS)
 - Einsatz mehrere Container in einer VM
- doppelte Virtualisierung



Bild von Mike by Pexels

Betriebssysteme | I.2



Franz J. Hauck | Institut für Verteilte Systeme, Univ. Ulm

Inhaltsüberblick

Virtualisierung

- Begriff
- virtuelle Maschine
- virtuelles Betriebssystem
- Mechanismen für Virtualisierung

Mechanismen für Virtualisierung

Unterbrechungen auf Prozessebene

- je nach Betriebssystem verfügbar
 - `void (*signal(int sig, void (*func)(int)))(int);`
 - **sig**: Identifikator für eine bestimmte Unterbrechung
 - **func**: Adresse einer Funktion zur Unterbrechungsbehandlung
- Aufruf der **registrierten Funktion** im Prozess durch das Betriebssystem
 - ähnlich Hardware-Unterbrechung
 - ausgelöst durch Betriebssystem
 - am Ende der Funktionsausführung fährt Prozess/Aktivitätsträger normal fort

Mechanismen für Virtualisierung (2)

Unterbrechungen auf Prozessebene (fortges.)

- typische **Auslöser** (Auswahl)
 - **SIGUSR1** – Kommunikation zwischen Prozessen
 - **SIGINT** – Drücken bestimmter Taste am Eingabegerät, z.B. STRG-C
 - **SIGSEGV** – Zugriff auf ungültige Speicheradresse
 - **SIGALRM** – Ablauf eines Zeitgebers
 - **SIGILL** – Aufruf eines illegalen/privilegierten Maschinenbefehls

Mechanismen für Virtualisierung (3)

Unterbrechungen auf Prozessebene (fortges.)

■ Einsatz bei Virtualisierung

- Emulation von **Hardware** durch Abfangen von Speicherzugriffen
- **Umschaltung** zwischen Threads (User-level Threads)
- Emulation von **privilegierten Befehlen**

Mechanismen für Virtualisierung (4)

Moderne Virtualisierungserweiterungen im Prozessor

- Beispielprozessor aus Kapitel C und D
 - S-Bit im Condition Code Register
(1 = Supervisor Mode, 0 = User Mode)
 - Problem:
 - VMM läuft als Anwendung (S=0), selbst wenn der VMM gerade im Gastbetriebssystem arbeitet
 - Gastbetriebssystem auf Hypervisor läuft als Betriebssystem (S=1), soll aber nicht alles dürfen

Mechanismen für Virtualisierung (5)

Moderne Virtualisierungserweiterungen im Prozessor

■ Lösung

- weiterer interner **Prozessorzustand**
 - real vs. virtualisiert
- im **realen** Modul arbeitet Prozessor wie bisher
- im **virtualisierten** Modus:
 - privilegierter Befehl bei **S=1**
 - führt zu einer Art Unterbrechung, die im realen Modus behandelt wird
 - spezielle im realen Modus konfigurierte Unterbrechung
 - prüft, ob VM den privilegierten Befehl ausführen darf
 - privilegierter Befehl bei **S=0**
 - führt zu Unterbrechung wie bisher

Inhaltsüberblick

Virtualisierung

- Begriff
- virtuelle Maschine
- virtuelles Betriebssystem
- Mechanismen für Virtualisierung