



UNIVERSITÄT ZU LÜBECK  
INSTITUT FÜR TELEMATIK

# Betriebssysteme

## Kapitel 06: Einführung Virtualisierung

Stefan Fischer, *Regine Wendt*  
Universität zu Lübeck  
Institut für Telematik  
13. Juli 2020



**Virtualität** ist die Eigenschaft einer Sache, nicht in der Form zu existieren, in der sie zu existieren scheint, aber in ihrem Wesen oder ihrer Wirkung einer in dieser Form existierenden Sache zu gleichen.<sup>(1)</sup>

**Virtuell** ist das Gegenteil von physisch – nicht von real (!).





*Die **Virtualisierung** abstrahiert IT-Ressourcen durch das Einfügen einer zusätzlichen Ebene zwischen Anwendung und Hardware. Für den Anwender verhält sich das virtuelle Objekt wie ein dediziertes Hard- oder Softwareobjekt (2)*

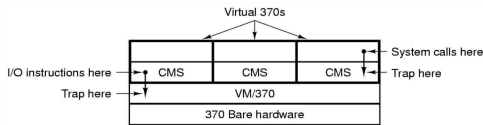
- ▶ Virtuelle Nachbildung von Geräten, Betriebssystemen oder Services
- ▶ Kein Unterschied für Anwender
- ▶ Verhalten nach außen wie physische Pendants



UNIVERSITÄT ZU LÜBECK  
STIFTUNGUNIVERSITÄT  
SEIT 1015

## Betriebssystem-Architekturen (III)

- Virtuelle Maschinen:
  - Trennung der Funktion multi-programming von den übrigen Funktionen eines Betriebssystems.
  - Emulation der Hardware zu sogenannten virtuellen Maschinen (VM).
  - Verschiedene Betriebssysteme können gleichzeitig auf den virtuellen Maschinen laufen.
  - Die virtuellen Maschinen werden durch ein Monitorprogramm (virtual machine monitor, VMM) voneinander getrennt.
- Beispiel: IBM VM/370



CMS: Conversational Monitor System

- ▶ Wirt (Host): Wirtsystem stellt Ressourcen bereit
- ▶ Gast: auf dem Host ausgeführte Instanz
- ▶ Hypervisor/VMM: erstellt und verwaltet virtuelle Hardware

- ▶ 1959: Time sharing
- ▶ 1964: IBM Großrechner System/360
  - ▶ Zentrale Steuereinheit
  - ▶ Hypervisor CP-40/CMS
- ▶ 1970: IBM VM/370





- ▶ 1959: Time sharing
- ▶ 1964: IBM Großrechner System/360
  - ▶ Zentrale Steuereinheit
  - ▶ Hypervisor CP-40/CMS
- ▶ 1970: IBM VM/370
- ▶ 1974 UCLA Popek und Goldberg:  
"Formal Requirements for Virtualizable Third Generation Architectures"
- ▶ 1978: x86-Architektur
- ▶ 80er/90er Popek und Goldberg nicht erfüllt

- ▶ 1974 UCLA Popek und Goldberg:  
"Formal Requirements for Virtualizable Third Generation Architectures"
- ▶ 1978: x86-Architektur
- ▶ 80er/90er Popek und Goldberg nicht erfüllt
- ▶ 1999: VMware Workstation 1.0



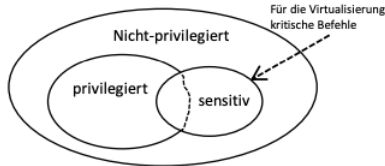
# Anforderungen an VMM



- ▶ Sicherheit: volle Kontrolle über Ressourcen
- ▶ Effizienz: wenige Eingriffe
- ▶ Wiedergabetreue: identisches Programmverhalten

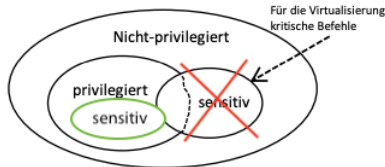
## Problem:

- ▶ sensitive Befehle  
-> Verhalten abhängig von Modus
- ▶ privilegierte Befehle  
-> Sprung ins Betriebssystem aus Usermodus



## Problem:

- ▶ sensitive Befehle  
-> Verhalten abhängig von Modus
- ▶ privilegierte Befehle  
-> Sprung ins Betriebssystem aus Usermodus



## Lösung:

- ▶ Popek & Goldberg: Sensitive Befehle Teilmenge der privilegierten Befehle

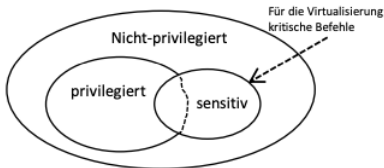
# Prozessoren ohne Unterstützung



Kritische Befehle:

- ▶ Sensitive nicht privilegierte Befehle
- ▶ Werden durch Hypervisor überschrieben
- ▶ Sprung in Hypervisor und Emulation

→ Binärübersetzung



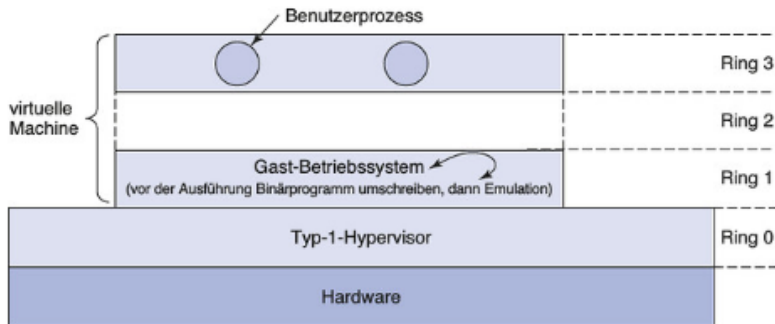
# Prozessoren ohne Unterstützung



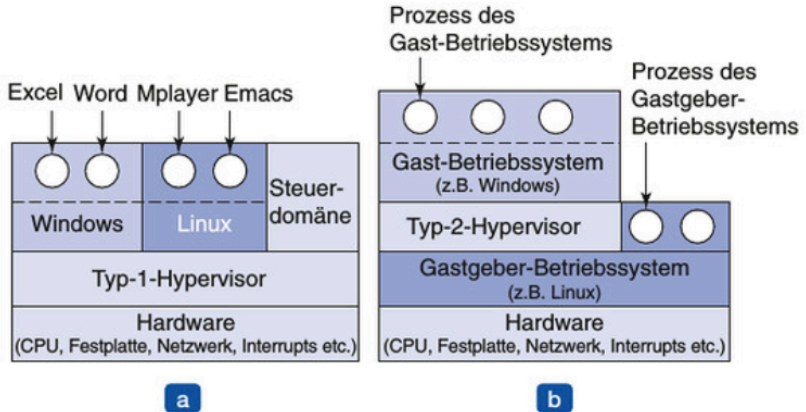
Kritische Befehle:

→ Binärübersetzung

→ Schutzringe nutzen



# Typ-1- und Typ-2-Hypervisor





- ▶ Para: griechisch nebenbei, entlang
  - ▶ Maschinenähnliche Softwareschnittstelle
  - ▶ VM nicht identisch mit tatsächlicher Hardware
  - ▶ Modifikation des Gastbetriebsystems nötig
  - ▶ Gast-BS kommuniziert mit Hypercalls
  - ▶ Gast-BS "weiß" von Hypervisor
- 
- ▶ Vorteil: Durch Hypercalls schnelleres Gesamtsystem
  - ▶ Nachteil: Gast-BS auf Hypervisor zugeschnitten

# Virtualisierungsmethoden



- ▶ Typ-1: läuft direkt auf Hardware
- ▶ Typ-2: nutzt Dienste existierenden Gastgeber-BS
- ▶ Paravirtualisierung: Gast-BS "weiß" von Hypervisor

Virtualisierungsmethode	Typ-1-Hypervisor	Typ-2-Hypervisor
Virtualisierung ohne Hardware-unterstützung	ESX Server 1.0	VMware Workstation 1
Paravirtualisierung	Xen 1.0	
Virtualisierung mit Hardware-unterstützung	vSphere, Xen, Hyper-V	VMware Fusion, KVM, Parallels

# Multicomputer vs. VM



- ▶ Unternehmen mit Multicomputern:
  - ▶ Viele Server, viele Rechner, über Netzwerk verbunden
  - ▶ Wg. Lastverteilung, Zuverlässigkeit, IT-Sicherheit
  - ▶ Nachteil: teuer, Verwaltung
- ▶ Vorteil Einsatz VM:
  - ▶ Partielles Ausfallsystem: Fehler in VM keine Konsequenz auf andere VM
  - ▶ "Multicomputer" mit geringeren Kosten und einfacher Wartung
  - ▶ Erhöhte Sicherheit durch virtuelle Maschinenisolation
  - ▶ Eine leistungsfähige Plattform günstiger als viele
- ▶ Nachteil Einsatz VM: "Alles auf eine Karte"

- ▶ Viele VM auf einem Computer:  
Mehrere (isolierte) Betriebssysteme gleichzeitig
- ▶ Portabilität: Umgebung nach Entwicklung "mitnehmen"
- ▶ Migration vereinfacht z.B. bei Lastausgleich
- ▶ Ausführung veralteter Anwendungen
- ▶ Cloud: Mehrere Kunden teilen eine Maschine

- ▶ Ausfall des Wirtsystems oder Hardware betrifft automatisch alle VM
- ▶ Effizienzverlust durch Hypervisor
- ▶ Hypervisoren angreifbar
- ▶ Gegenseitige Beeinflussung
  - ▶ Lastverhalten nicht vorhersehbar
  - ▶ Teilung beschränkter Ressourcen
- ▶ Lizenzierung von Betriebssystemen

## Nächste Vorlesung:

Docker und Clouds

### Quellen:

Andrew S. Tanenbaum / Herbert Bos: Moderne Betriebssysteme, 4. Auflage, Pearson Studium, 2016

1: <https://educalingo.com/de/dic-de/virtualitat>

2: <https://www.cloudcomputing-insider.de/was-ist-virtualisierung-a-756279/>

3: [https://www.wirtschaftsinformatik-muenchen.de/wp-content/uploads/Peter20Mandl/Lehrveranstaltungen/SoSe2015/Wirtschaftsinformatik/14\\_Betriebssystemvirtualisierung.pdf](https://www.wirtschaftsinformatik-muenchen.de/wp-content/uploads/Peter20Mandl/Lehrveranstaltungen/SoSe2015/Wirtschaftsinformatik/14_Betriebssystemvirtualisierung.pdf)