Virtuelle Maschinen

Dr.-Ing. Volkmar Sieh

Department Informatik 4
Verteilte Systeme und Betriebssysteme
Friedrich-Alexander-Universität Erlangen-Nürnberg

WS 2022/2023

Organisatorisches

Vorlesungen:

- Dr.-Ing. Volkmar Sieh
- Di. 14:15-15:45, 0.031

Übungen:

- Dr.-Ing. Volkmar Sieh
- Mo. 12:15-13:45, 0.031

Erweiterte Übungen:

- Dr.-Ing. Volkmar Sieh
- (fast) jederzeit...

Folien und Übungsblätter im Netz.

Bei Fragen: E-Mail, Telefon, persönlich, ... (fast) jederzeit...

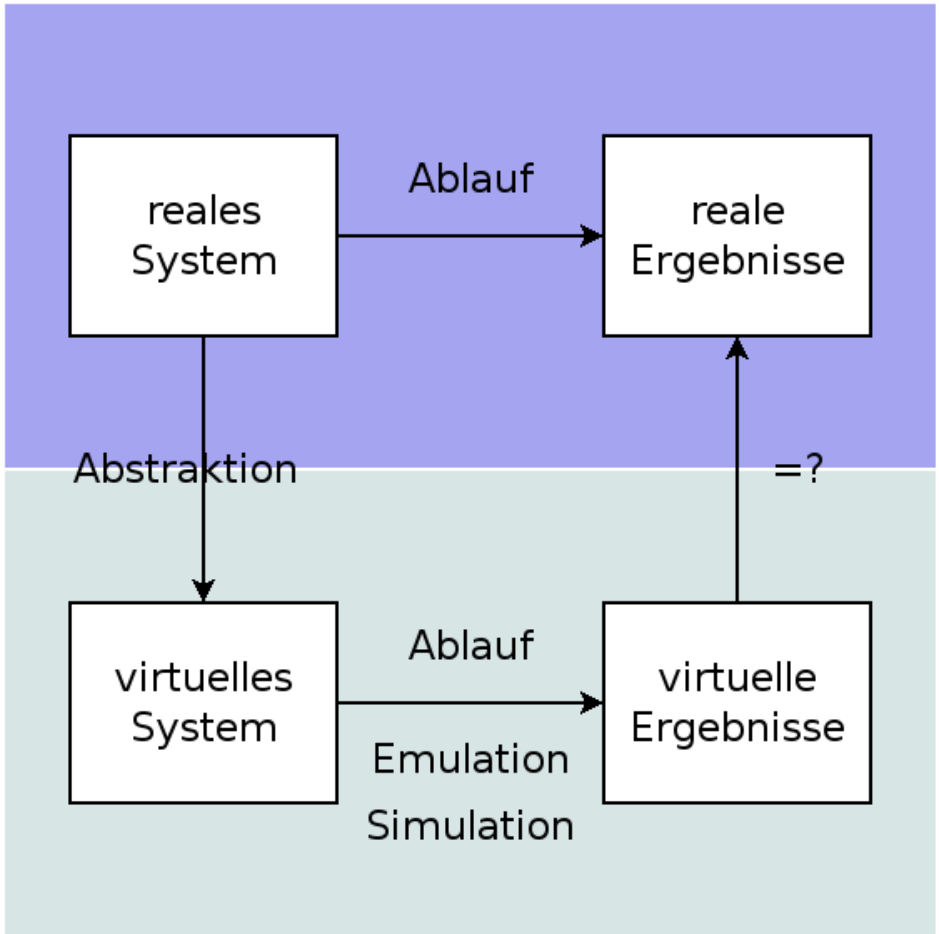
Bitte per Waffel anmelden (siehe Homepage)!

Organisatorisches

Prüfungen:

- 5 ECTS:
  - Abgabe von Programmieraufgaben erwünscht(!) (Gruppenarbeit)
  - mündliche Prüfung in den Semesterferien
- 7,5 ECTS
  - Abgabe von Programmieraufgaben obligatorisch(!) (Gruppenarbeit)
  - mündliche Prüfung in den Semesterferien

Einleitung


reales System
Ablauf
reale Ergebnisse
Abstraktion
=?
virtuelles System
Ablauf
virtuelle Ergebnisse
Emulation
Simulation

Definition: Virtuelle Maschine

Wikipedia:

Virtuelle Maschine (VM) ... bezeichnet entweder ein simuliertes Betriebssystem oder eine simulierte Laufzeitumgebung für Programme innerhalb eines Computers.

Definition: Virtuelle Maschine

Andere Sichtweise:

Aus Sicht einer Anwendung auf einer *normalen* Hardware werden die Instruktionen der Anwendung (in Hardware) interpretiert und ausgeführt.

Aus Sicht einer Anwendung auf einer *simulierten* Hardware werden die Instruktionen der Anwendung (in Software) interpretiert und ausgeführt.

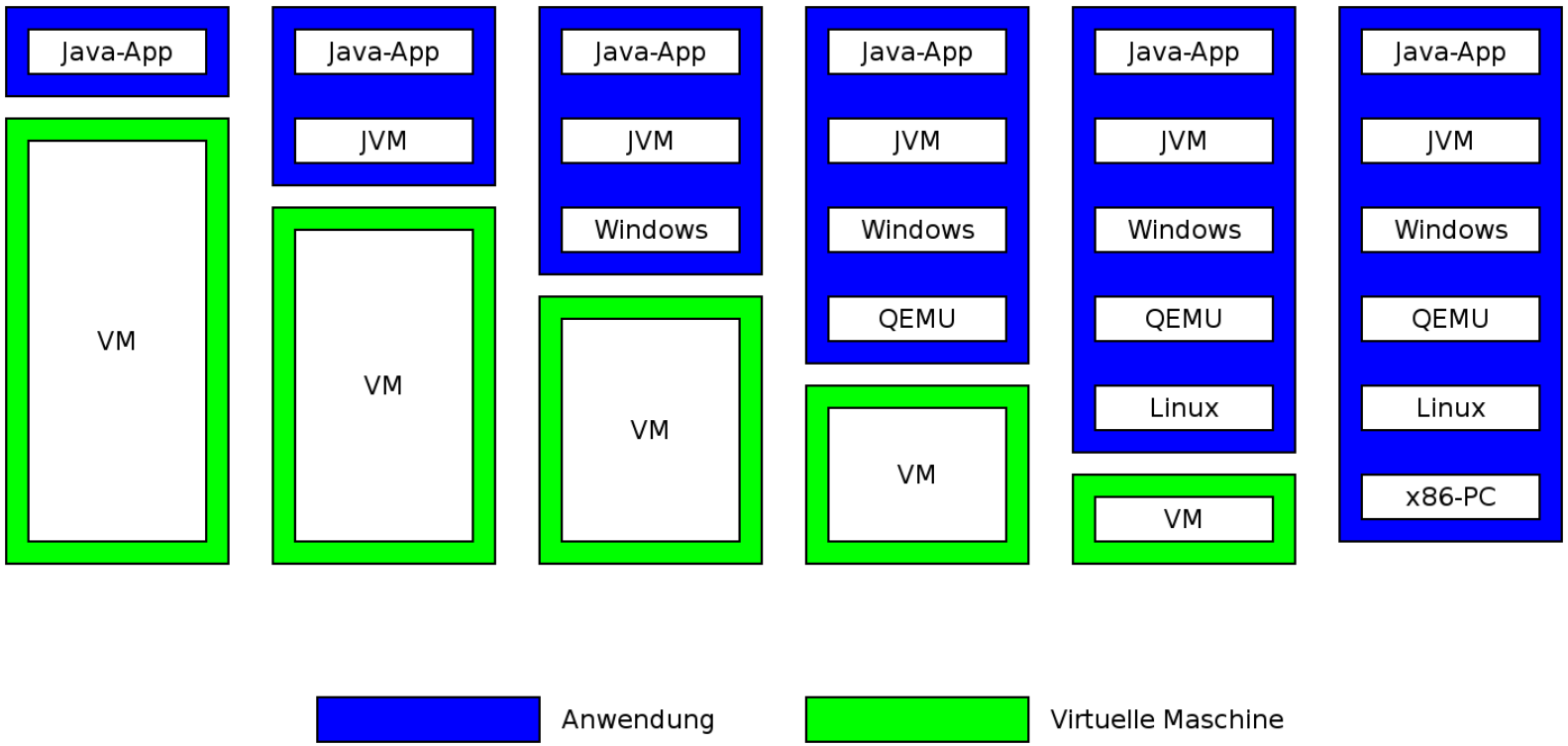
Aus Sicht einer Anwendung auf einem *normalen* Betriebssystem werden die Instruktionen der Anwendung (in Hardware und mit Betriebssystem-Funktionen) interpretiert und ausgeführt.

...

D.h.: jeder Software- und Hardware-Stack unterhalb einer Anwendung kann als (virtuelle) Maschine angesehen werden!

Definition: Virtuelle Maschine

Beispiel:

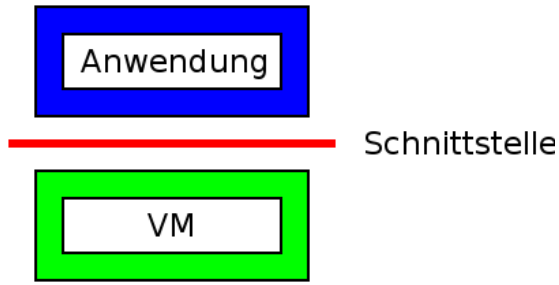


Definition: Virtuelle Maschine

Schnittstelle:

ISA: Instruction Set Architecture

ABI: Application Binary Interface



Schnittstelle definiert:

- mögliche Befehle
- Kodierung der Befehle
- Semantik der Befehle
- Daten für die Befehle
- ...

D.h. die Definition einer Schnittstelle beschreibt eine Virtuelle Maschine!

Definition: Virtuelle Maschine

Normalerweise: Schnittstellen-Definition

- beschreibt funktionale Eigenschaften
- ignoriert nicht-funktionale Eigenschaften; z.B.:
  - Zeit
  - Stromverbrauch
  - Wärmeerzeugung
  - Fehler-Verhalten
  - ...

Definition: Gast/Gastgeber

Gastgeber-Hardware/Host-Hardware:
: die reale Hardware

Gastgeber-Betriebssystem/Host-OS:
: das auf der realen Hardware laufende Betriebssystem

Gastgeber-Applikation/Host-Application:
: die auf der realen Hardware laufende Applikation

Gast-Hardware/Guest-Hardware:
: die virtuelle Hardware

Gast-Betriebssystem/Guest-OS:
: das auf der virtuellen Hardware laufende Betriebssystem

Gast-Applikation/Guest-Application:
: die in der virtuellen Maschine laufende Applikation

Motivation – Rechenzentrum

Server-Konsolidierung: In Rechenzentren gibt es viele dedizierte Rechner. Häufig: ein Rechner pro Applikation. In vielen Fällen reicht ein virtueller Rechner pro Applikation.

Lastverteilung: Laufen mehrere Applikationen auf einem Server, kann dieser überlastet sein. Sie von einem Server auf einen anderen zu verschieben, ist u.U. schwierig. Virtuelle Maschinen zu verschieben, ist einfach (Kopieren des Platten-Images).

Motivation – mehrere OS

Windows auf Linux: Linux-User wünschen u.U. Windows-Software (z.B. Office-Suite).

Linux auf Windows: Windows-User wünschen u.U. Linux-Sicherheit (z.B. beim Surfen).

...

Motivation – Hardware-Abstraktion

Hardware-Abstraktionsschicht: Eventuell läuft Software aufgrund von fehlenden Treibern nicht auf einem echten Rechner. Eine virtuelle Maschine kann andere Hardware (virtuell) zur Verfügung stellen.

Ersatz für alte Hardware: Software, die für ältere Systeme entwickelt worden ist, kann auf modernen Systemen weiter laufen. Alte Konsolenspiele aus den frühen achtziger Jahren können dank geeigneter Emulatoren wie z. B. M.E.S.S. auf moderner Hardware laufen.

Ersatz für noch nicht existierende Hardware: Test-Umgebung für Software, die für noch nicht existierende Hardware geschrieben werden soll.

Architektur-Unabhängigkeit: Z.B. laufen Java-Programme mittels virtueller Maschinen auf vielen OS auf verschiedener Hardware.

Motivation – Software-Entwicklung

Programmierung Netzwerk-Software: Mit Hilfe mehrerer virtueller Maschinen kann man sich ein kleines Rechner-Netzwerk auf einem einzelnen echten Rechner aufbauen. Nützlich für das Testen während der Entwicklung von Netzwerk-Software.

OS-Debugging: Ein OS zu debuggen ist schwierig, da man keine Debugger zur Verfügung hat. Häufig existiert nicht einmal ein `print`. Virtuelle Maschine kann z.B. einen Trace der ausgeführten Instruktionen ausgeben.

Software-Entwicklung: Entwicklung von Software für nicht wirklich vorhandene Hardware oder Hardware-Konfigurationen.

Motivation – Misc

Ausbildung: Virtuelle Rechner sind schnell zu installieren (Platten-Image kopieren). Daher ideal für Ausbildungszwecke. SchülerInnen können virtuellen Rechner ruhig einmal „kaputt"-konfigurieren. Zum Beispiel sind Root-Rechte als normaler User damit kein Problem.

Fehlerinjektion: Echte Hardware ist praktisch nicht bezüglich Fehlertoleranz-Verhalten untersuchbar. Fehler treten nicht auf, wenn man auf sie wartet. Ein virtueller Rechner geht genau dann kaputt, wann man es möchte.

Ergonomie: Software, die normalerweise nur auf Systemen mit unergonomischen Ein-/Ausgabegräten läuft (LC-Displays), kann auf Systemen mit komfortablen Bildschirmen laufen. Beispiel: Die Bildwiedergabe bei Game-Boy-Emulatoren auf einem PC ist besser als bei einem realen Game Boy.

Motivation – Misc

Honey-Pot: Virtuelle Maschine kann gefahrlos zum Anlocken und Untersuchen von Viren o.ä. verwendet werden.

Automatisierbarkeit: Benutzereingaben können der virtuellen Maschine per Skript übertragen werden. Tests können damit automatisch durchgeführt werden.

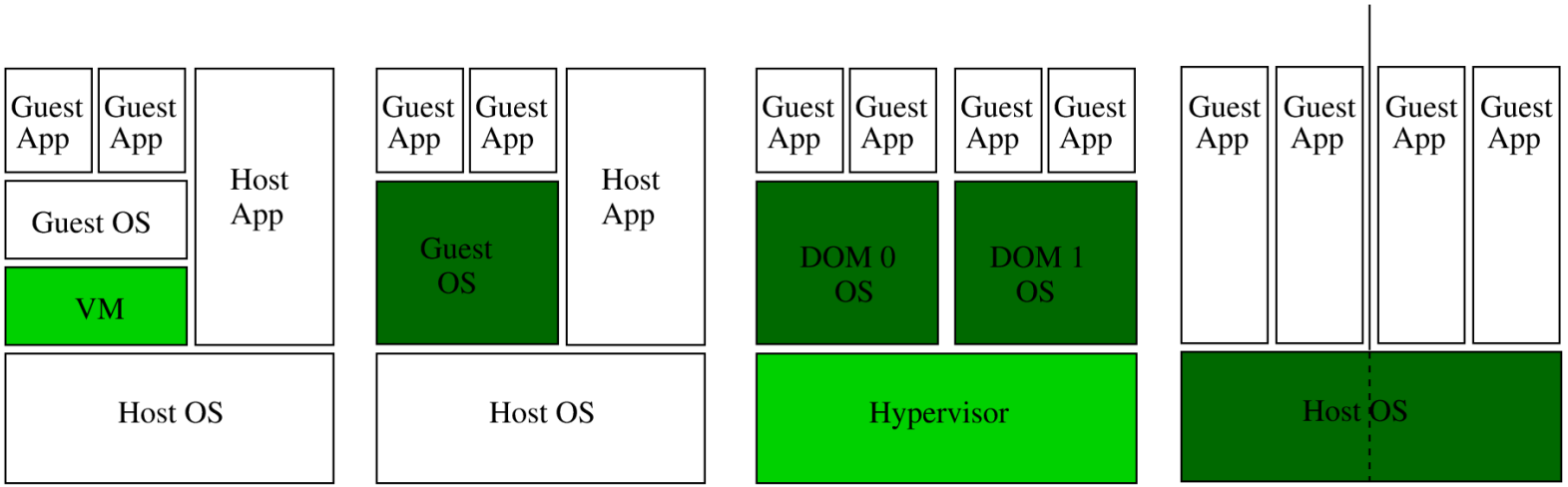
...

Virtuelle Maschinen

Verschiedene Virtuelle Maschinen; z.B.:

- ein virtueller PC
- ein virtuelles Betriebssystem
- eine virtuelle Laufzeitumgebung
- ...

Virtuelle Maschinen



Bochs
QEMU
FAUmachine

UMLinux

XEN

VServer

Virtuelle Maschinen – Ziele

- Performance
  - Server-Konsolidierung
  - Last-Verteilung
  - mehrere OS
  - Software-Entwicklung
  - ...
- Konfigurierbarkeit
  - Ersatz für alte/neue Hardware
  - Software-Entwicklung
  - Ausbildung
  - Fehlerinjektion
  - ...
- Genauigkeit
  - Ersatz für alte/neue Hardware
  - OS-Debugging
  - Fehlerinjektion
  - ...
- ...

Ziel der Vorlesung/Übung

„Virtuelle Maschinen” (**VM**) sind zur Zeit wichtiges Forschungsgebiet weltweit.

- Wir haben inzwischen viel Erfahrung (ca. 17 Jahre) mit **FAUmachine**-Projekt
- **FAUmachine** ist typische Rechner-Emulation
- viele andere **Open-Source**-Projekte sind bekannt

Weitergabe der vielen Erfahrungen

Ziel der Vorlesung/Übung

Am Ende des Semesters ist jede(r) StudentIn in der Lage,

- die Funktionsweise einer VM zu erfassen,
- aus einer Vielzahl existierender VM für eine bestimmte Aufgabe die geeignete VM herauszusuchen,
- Vor- und Nachteile von VM-Typen zu erkennen,
- eine eigene virtuelle Maschine zu entwerfen/programmieren.

Voraussetzungen

- Hardware-Kenntnisse
- Assembler-Kenntnisse
- C-/C++-/Java-Programmier-Kenntnisse
- **Motivation ...**
- **Einsatz ...**