

# Betriebssysteme Praxis

WS 2011/12

**Hans-Georg Eßer**  
Dipl.-Math., Dipl.-Inform.

Foliensatz A (22.09.2011)  
Einführung, Virtualisierung, Linux



## Hilfreiche Vorkenntnisse:

- **Linux-Shell** – Benutzung der Standard-Shell *bash* unter Linux  
→ Bash-Crashkurs

## **Praktikum:**

Systemadministration unter Linux

## **Prüfung und Benotung**

1. Lernfortschrittskontrolle (LFK)
2. Klausur über 120 Minuten

## **Fragen:**

- direkt in der Vorlesung (Handzeichen)
- oder danach
- oder per E-Mail

## Termine

Do. 22.09.2011	M/FOM-A02	Mi. 14.12.2011	M/FOM-A04
Mi. 19.10.2011	M/FOM-A04	Mi. 21.12.2011	M/FOM-A03
Mi. 26.10.2011	M/FOM-A02	Mi. 11.01.2012	M/FOM-A03
Do. 10.11.2011	M/FOM-A07	Do. 19.01.2012	M/FOM-A03
Do. 01.12.2011	M/FOM-A08	Do. 26.01.2012	Klausur

**Service / Web-Seite: <http://fom.hgesser.de>**

- Folien und Praktikumsaufgaben
- Vorlesungs-Videos („*test, test*“)
- Probeklausur gegen Semesterende

## Hans-Georg Eßer

- Dipl.-Math. (RWTH Aachen, 1997)  
Dipl.-Inform. (RWTH Aachen, 2005)
- Chefredakteur Linux-Zeitschrift (seit 2000) und  
Autor diverser Computerbücher
- LPI-zertifiziert (LPIC-1 und LPIC-2)
- seit 2006 Dozent an der Hochschule München und  
an der FOM: Betriebssysteme, Rechnerarchitektur,  
IT-Infrastrukturen, Informatik-Grundlagen
- Seit 2010 Doktorand (Univ. Erlangen-Nürnberg)

# Einführung und Motivation

## Ein praktischer Kurs

- Installation
  - Verständnis für Partitionierung, Dateisysteme, Boot-Vorgang
- Wartung
  - Nutzen von Shell-Tools (Linux)
  - Software-Installation
  - Einrichtung von Geräten
  - Benutzer-Verwaltung, Sicherheit
  - Netzwerk-Konfiguration, Netzwerk-Dienste



## LPI Certified Junior Level Linux Professional

- Work at the Linux command line
- Perform easy maintenance tasks: help out users, add users to a larger system, backup & restore, shutdown & reboot
- Install and configure a workstation (including X) and connect it to a LAN, or a stand-alone PC via modem to the Internet.



## LPI Certified Advanced Level Linux Professional

- Administer a small to medium-sized site
- Plan, implement, maintain, keep consistent, secure, and troubleshoot a small mixed (MS, Linux) network, including a LAN server (samba), Internet Gateway (firewall, proxy, mail, news), Internet Server (webserver, FTP server)
- Supervise assistants
- Advise management on automation and purchases



## LPI Certified Senior Level Linux Professional

- hauptsächlich: LDAP
- dazu: verschiedene Spezialisierungen
  - LPI 302: Mixed Environments
  - LPI 303: Security
  - LPI 304: Virtualization and High Availability
  - es kommen vielleicht noch weitere

Im Rahmen dieser Vorlesung:  
teilweise (!) Vorbereitung  
auf die LPIC-1-Prüfungen  
101 und 102



- Für die Zertifizierung: Stoff aus dieser Veranstaltung reicht nicht
- insbesondere: „passives Konsumieren“ reicht nicht
- Prüfungen sind so gestaltet, dass Administratoren sie leicht bestehen können  
→ üben, üben, üben :)
- Kosten: 145 € pro Prüfung (LPIC-1 = 2 Prüfungen)

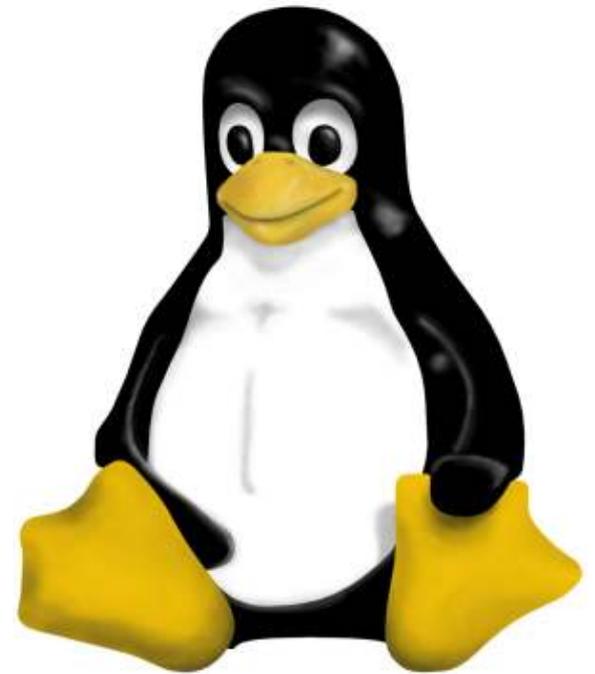
- keine grafischen Werkzeuge – auch wenn es welche gibt
- also: nicht YaST & Co., sondern Kommandozeilentools, Konfigurationsdateien, Shell-Skripte
- verstehen, was im Hintergrund abläuft

- Windows setzt auch bei der Administration überwiegend auf grafische Tools
- „GUI-Administration“ leichter (schneller) zu erlernen, bietet aber weniger Möglichkeiten, wenn etwas schief geht
- Windows ist im Rahmen dieser Vorlesung kein Thema

# Gliederung

1. Einleitung
2. Virtualisierung
3. Allgemeine Grundlagen
  - Partitionen, RAID
3. Linux-Administration
4. Windows-Administration

- Etabliertes Standardsystem für sehr viele Plattformen (PC Desktop / Server, Embedded etc.)
- vor allem auf Servern weit verbreitet
- Image eines virtuellen Linux-PCs für VMware / VirtualBox



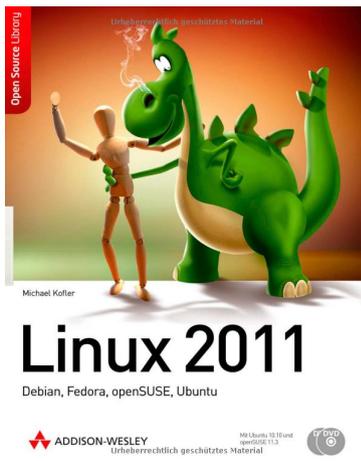


## Grundlagenbuch Linux

Grundlagen, Techniken, Lösungen  
(Eßer, Dölle)

Data Becker, 2007

→ als PDF-Dokument im Campus-System



## Linux 2011

Debian, Fedora, openSUSE, Ubuntu  
(Kofler)

Addison-Wesley, 2010

49,80 €; E-Book: 39,80 €

## 2. Virtualisierung

- Idee: Ein Betriebssystem läuft nicht direkt auf der Hardware, sondern als Gast in einer virtuellen Maschine – unter einem Host-Betriebssystem
- Verschiedene Arten der Virtualisierung
  - Full Virtualization (z. B. VMware, VirtualBox)
  - Hypervisor ohne darunter liegendes Betriebssystem (z. B. VMware ESX Server)
  - Paravirtualisierung (z. B. Xen)

# Virtualisierung: Warum? (1)

- Ungenutzte Rechenkapazitäten besser ausnutzen (vgl.: Multi-Processing)
- einheitliche Hardware: BS auf virtueller Maschine muss nicht an konkrete Hardware angepasst werden
- Server-Bereich:
  - Konsolidierung – wenige große Maschinen statt vielen kleinen
  - vereinfachtes Backup aller virt. Maschinen

# Virtualisierung: Warum? (2)

- Einfachere Bereitstellung einer neuen Maschine, einfaches Duplizieren
- Snapshots erlauben Rückkehr zu funktionierendem Zustand einer Maschine
- Sicherheit: Voreinander zu schützende Anwendungen besser in separaten VMs als auf einem logischen Rechner laufen lassen
- Für Entwickler: Test auf verschiedenen Plattformen, ohne dafür jeweils 1 PC zu benötigen
- Legacy-Support (alte BS)

## Warum im Rahmen dieser Vorlesung?

- einheitliche Hardware-Umgebung für ein zu testendes Linux-System (Debian)
- Systemverwaltung ohne das Risiko, Ihre Notebooks „kaputt“ zu konfigurieren

# Virtualisierung: Konzepte

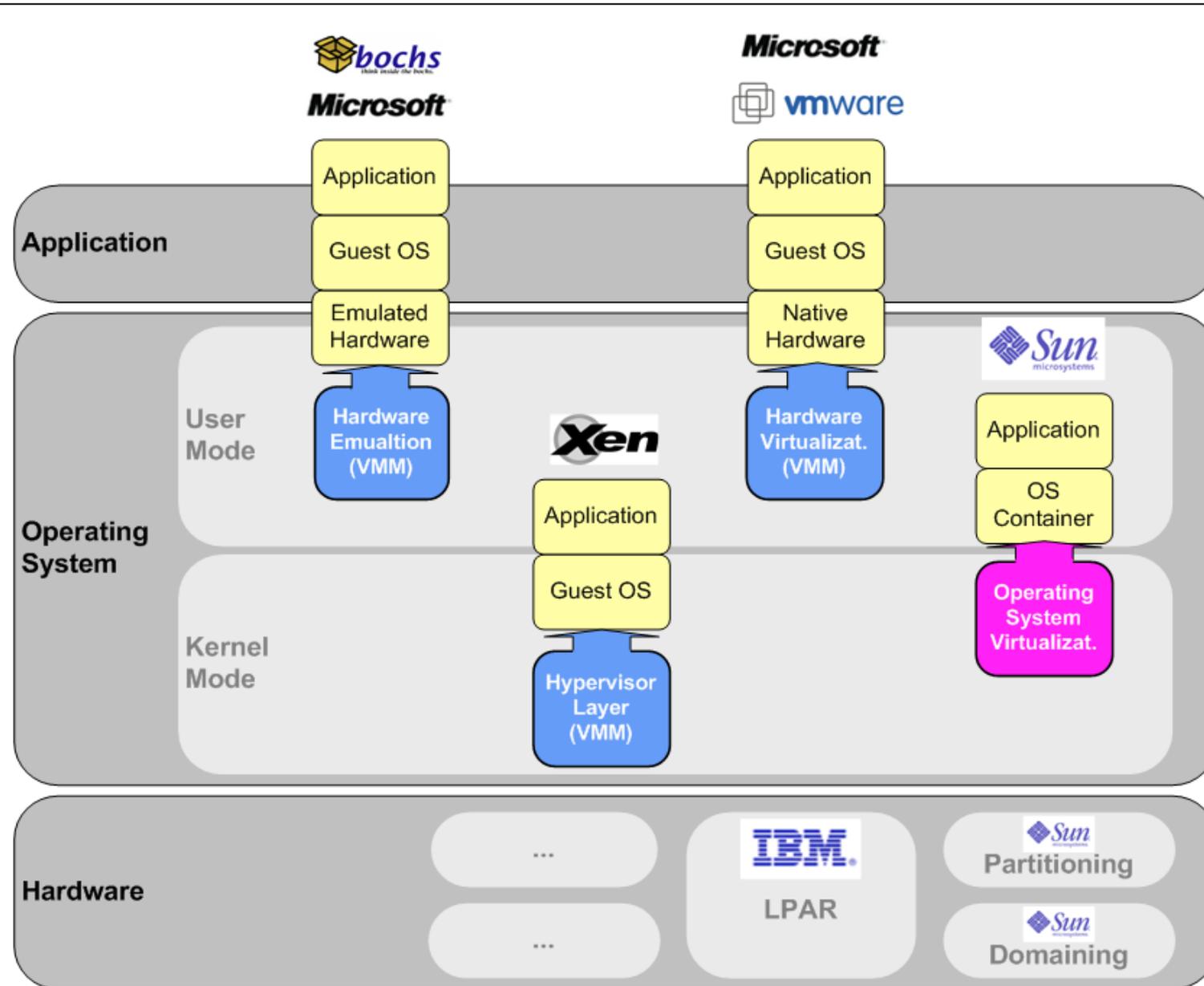


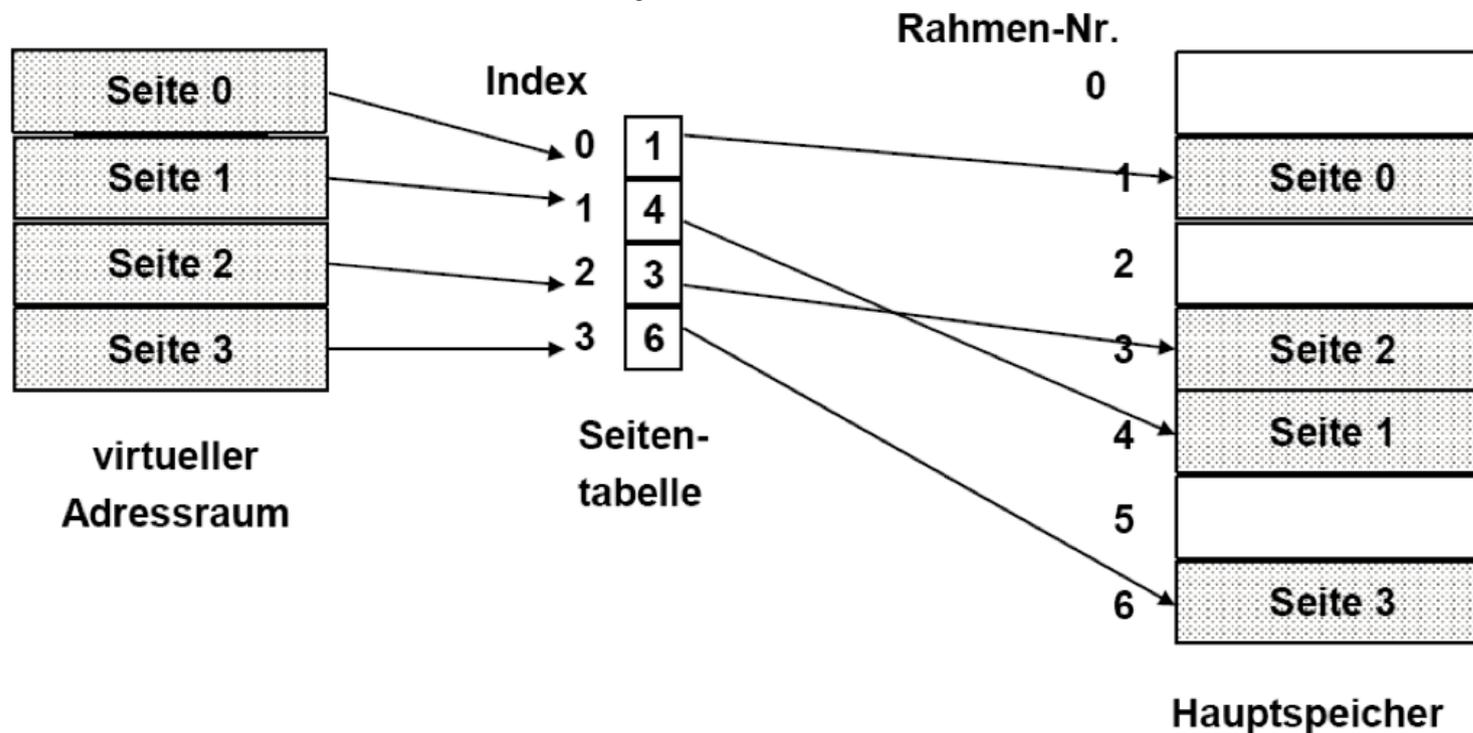
Bild: Daniel Hirschbach, Lizenz: CC-by-sa 2.0/de  
(Quelle: Wikipedia, Virtualisierung)

# Virtualisierung: Probleme

- Einfache User-Mode-Instruktionen (arithmetische und logische Instruktionen, Speicherzugriff bei virtuellem Speicher/Paging) sind direkt ausführbar
- Andere Instruktionen müssen abgefangen werden (im User Mode: Software-Interrupt, OS Call; im System Mode: Hardware-Zugriffe)
- „virtualisierbare“ vs. „nicht-virtualisierbare“ Instruktionen

# Paging in der VM (1)

- Paging, bekannt aus BS-Theorie (3. Sem.):
  - Für jeden Prozess verwaltet das BS eine Seitentabelle, die virtuelle Seiten auf physikalische Seitenrahmen abbildet



- Beim Prozesswechsel wird das Seitentabellenregister (Page Table Pointer) in der CPU aktualisiert

# Paging in der VM (2)

- Was tun bei einem virtualisiertem BS? (1/3)
  - Annahme: Host und Gast arbeiten beide mit Paging und verwenden dieselbe Seitengröße
  - Speicherzugriff im User-Mode (aus einer Anwendung in der virtuellen Maschine) ist eine nicht-privilegierte Operation – es muss also eine Seitentabelle geben, die Seiten auf die echten Seitenrahmen (auf dem Hostsystem) abbildet
  - Gastsystem darf diese Tabelle nicht selbst erzeugen (liegt außerhalb der VM) und nicht selbst das echte CPU-Register verändern

- Was tun bei einem virtualisiertem BS? (2/3)

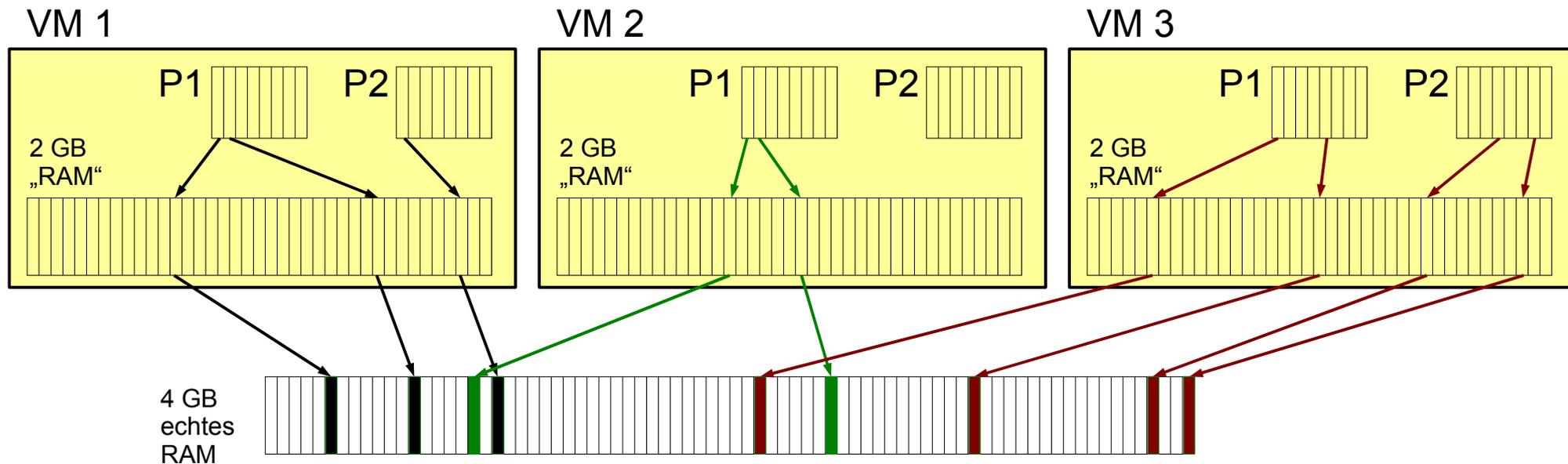
- Eigentliche zweistufige Übersetzung nötig:
  - Zuordnung  
virtuelle Seite (Gast) → physik. Frame (Gast)  
(durch „virtuelle“ MMU, mit Seitentabelle im Gast)
  - Zuordnung  
physik. Frame (Gast) → physik. Frame (Host)

VMM müsste also jeden Speicherzugriff abfangen und „umbiegen“, d. h. in Software die Funktion der MMU nachbilden

- Was tun bei einem virtualisiertem BS? (3/3)
  - Lösung: zwei Seitentabellen – eine virtuelle Page Table (im Gast) und eine zugehörige „Shadow Page Table“ (im Host)
  - Wenn das Gast-BS beim Context Switch die Seitentabelle austauschen will (also das PT-Register schreibt), wird dies abgefangen; dann wird im PT-Register die Adresse der passenden Shadow Page Table eingetragen

# Paging in der VM (5)

- Beispiel:
  - Host: 4 GByte RAM
  - 3 Gäste, jeweils 2 GByte virtuelles RAM
  - je zwei Prozesse pro Gastsystem



## Ablauf beim Context Switch im Gast-BS:

- Gast-BS führt Scheduler aus und wählt einen neuen Prozess aus
- Gast-BS setzt PT-Register auf neue Seitentabelle für diesen Prozess
- VMM fängt diesen Aufruf ab und sucht passende Shadow Page Table
- VMM setzt PT-Register und gibt Kontrolle an Gast-BS zurück
- Gast-BS vollendet Context Switch und gibt Kontrolle an neuen Prozess

# Emulation vs. Virtualisierung

- Emulation (von Hardware) ist ein anderes Konzept und nicht mit Virtualisierung zu verwechseln
- Emulator ahmt Hardware inkl. CPU vollständig nach
- Instruktionen werden also im Emulator nicht ausgeführt, sondern „interpretiert“
- Emulatoren können auch Hardware mit abweichenden CPUs emulieren (z. B. C64-Emu)

- Beispiel WINE („WINE Is Not an Emulator“)
- WINE ersetzt auf Linux-Systemen den Programm-Loader und diverse Bibliotheken (DLLs), die Software unter Windows erwartet.
- WINE ist also kein Hardware-Emulator und stellt auch keine virtuelle Maschine bereit.
- Windows-Programme laufen dank WINE fast „nativ“ auf dem Linux-System.

## Praxisteil

- Installation VirtualBox auf Ihrem Notebook / Netbook
- Anlegen einer virtuellen Maschine (5 GByte Plattenplatz, 512 MByte RAM)
- Installation von Debian Linux 5 in der virtuellen Maschine
- erste Experimente mit dem installierten Linux

## Praxisteil

- Arbeiten mit der Shell
- Verzeichnisnavigation, -Listings
- Dateien kopieren, umbenennen, verschieben
- Verzeichnisse erstellen, löschen etc.
- Dateien öffnen
- Der Editor „vi“