

BETRIEBSSYSTEME WEITERE THEMEN

Prof. Dr. Jörg Mielebacher
joerg.mielebacher@mosbach.dhbw.de

Die folgenden Folien behandeln weitere Themen rund um das Thema Betriebssysteme. Zu jeder Folie sind Notizenseiten erfasst.

Verbesserungsvorschläge und Fehlerhinweise können Sie gerne an die Adresse joerg.mielebacher@mosbach.dhbw.de senden.

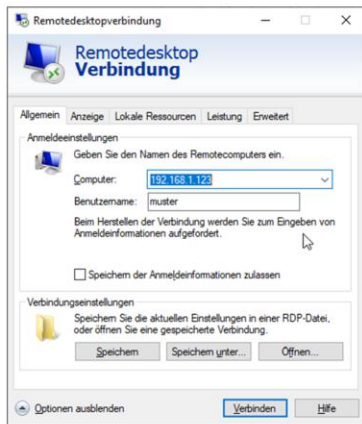
Rechtliche Hinweise: Die Rechte an geschützten Marken liegen bei den jeweiligen Markeninhabern. Alle Rechte an diesen Folien, Notizen und sonstigen Materialien liegen bei ihrem Autor, Jörg Mielebacher. Jede Form der teilweisen oder vollständigen Weitergabe, Speicherung auf Servern oder Nutzung in Lehrveranstaltungen, die nicht von dem Autor selbst durchgeführt werden, erfordert seine schriftliche Zustimmung. Eine schriftliche Zustimmung ist darüber hinaus für jede kommerzielle Nutzung erforderlich. Für inhaltliche Fehler kann keine Haftung übernommen werden.

Wiederholung

- Kommandozeilenparametern werden beim Aufruf eines Programms mit übergeben.
- Die Ein- und Ausgabe ist meist gepuffert.
- Threads werden in C++ durch Objekte der Klasse `thread` erzeugt.
- Race conditions müssen verhindert werden.
- Die kritischen Bereiche werden durch mutex-Objekte geschützt; so kann stets nur ein Thread darauf zugreifen, während die anderen warten.
- Deadlocks können auftreten, wenn Threads sich gegenseitig sperren.

Thin Clients und Remote-Zugriff

Remote Desktop Protocol (RDP)



- Proprietäres Protokoll von Microsoft
- Übertragung des Bildschirms
- Nutzung von Tastatur, Maus, Sound usw.
- TCP/UDP auf Port 3389
- Windows Terminal Services / Remote Desktop Services
- RDP-Server auch für Linux und als Teil von VirtualBox
- Clients für viele Plattformen
- Sicherheit teils problematisch

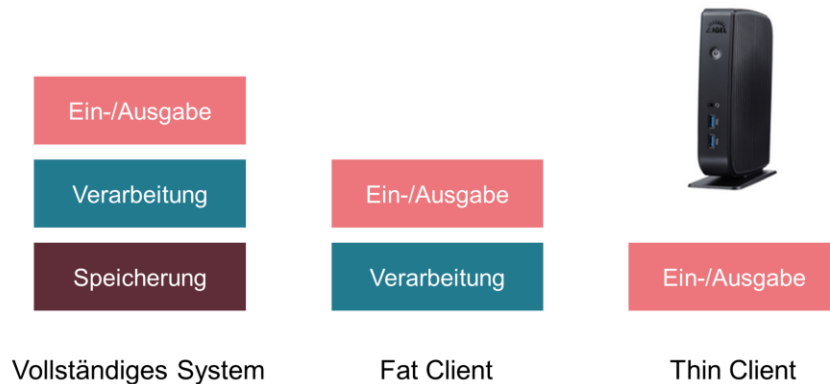
Das Remote Desktop Protocol ist ein proprietäres Protokoll von Microsoft, das dem Zugriff auf einen Computer über das Netzwerk dient. Mit RDP lässt sich der Inhalt des Bildschirms übertragen, aber auch Tastatur, Maus, Sound usw. über das Netzwerk nutzen. RDP nutzt TCP oder UDP auf Port 3389. Es ist die Grundlage für die Windows Terminal Services (heute: Remote Desktop Services); bei denen Server Desktopsitzungen für mehrere Benutzer bereitstellen. Der Vorteil dieses Ansatzes liegt darin, dass die einzelnen Clients nur wenig Rechenleistung benötigen und eine flexible Anmeldung auf unterschiedlichen Clients (auch z.B. über VPN) möglich ist. RDP-Server sind in Windows integriert, stehen aber auch für Linux zur Verfügung (FreeRDP). In der Virtualisierungslösung VirtualBox ist ein RDP-Server integriert. RDP-Clients gibt es für viele Plattformen (z.B. rdesktop unter Linux). RDP-Sitzungen lassen sich leicht aufbauen, allerdings haben sich in den vergangenen Jahren immer wieder erhebliche Sicherheitslücken gezeigt, z.B. bei der Verschlüsselung der Verbindungen.

Virtual Network Computing (VNC)

- Übertragung des Bildschirms über Remote Frame Buffer protocol (RFB)
- Nutzung von Tastatur, Maus, Sound usw.
- Quelloffen, unterschiedliche Forks
- VNC Server vs. VNC client/viewer
- TCP auf Port 5900
- Einsatz v.a. für Fernwartung
- Sicherheit sehr gemischt

Eine freie Alternative zu RDP ist das Virtual Network Computing (VNC), das über das Remote Frame Buffer Protocol die Übertragung des Bildschirminhalts erlaubt und wie bei RDP die entfernte Nutzung von Ein-/Ausgabegeräten erlaubt. Die Verbindung nützt üblicherweise TCP auf Port 5900. VNC ist quelloffen und steht in unterschiedlichen Forks zur Verbindung. Es ist besonders im Bereich der Fernwartung beliebt, auch wenn es in den letzten Jahren oft durch andere Lösungen (z.B. TeamViewer) verdrängt wurde. Die Sicherheit variiert stark zwischen den Implementierungen, da RFB alleine keine gute Sicherheit bietet, daher wird VNC oft in sichere Kanäle eingebettet (SSH, VPN usw.) bzw. um Verschlüsselungsschichten erweitert.

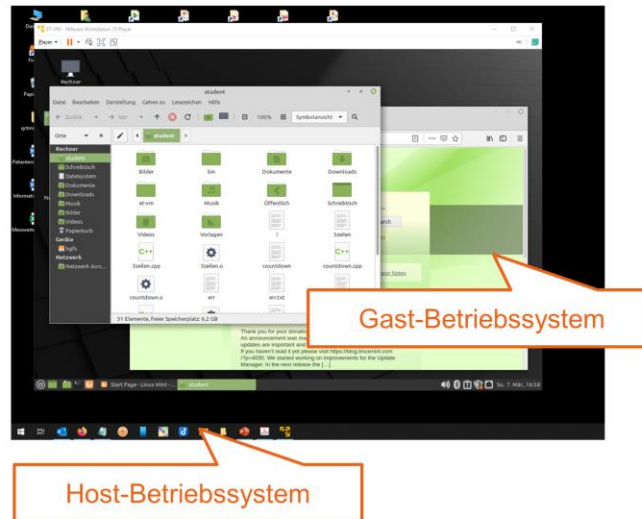
Thin Clients



Die Nutzung von Protokollen wie RDP erlaubt es, die Clients sehr schlank zu halten, während die Speicherung und Verarbeitung von einem Server übernommen werden. Ein vollständiger Client übernimmt die Speicherung der Daten, die Verarbeitung sowie die Ein-/Ausgabe. Verbleibt die Speicherung auf dem Server und der Client übernimmt lediglich Verarbeitung und Ein-/Ausgabe, so spricht man von einem Fat Client. Oft anzutreffen ist eine Steigerung hiervon: Bei sog. Thin Clients erfolgen Speicherung und Verarbeitung auf dem Server, lediglich die Ein-/Ausgabe ist Aufgabe des Clients. Dementsprechend benötigen die Thin Clients nur wenig Rechenleistung und verfügen meist nur über ein sehr schlankes Betriebssystem (oft auf Basis von Linux), das lediglich die Aufgabe hat, die Netzwerkverbindung zu ermöglichen und das entsprechende Fernzugriffsprotokoll umzusetzen. Daher sind sie meist kostengünstig und wartungsfreundlich, allerdings muss der Server über eine der Zahl der Client angemessene Leistung verfügen und das Netzwerk muss entsprechende Bandbreite und Stabilität besitzen.

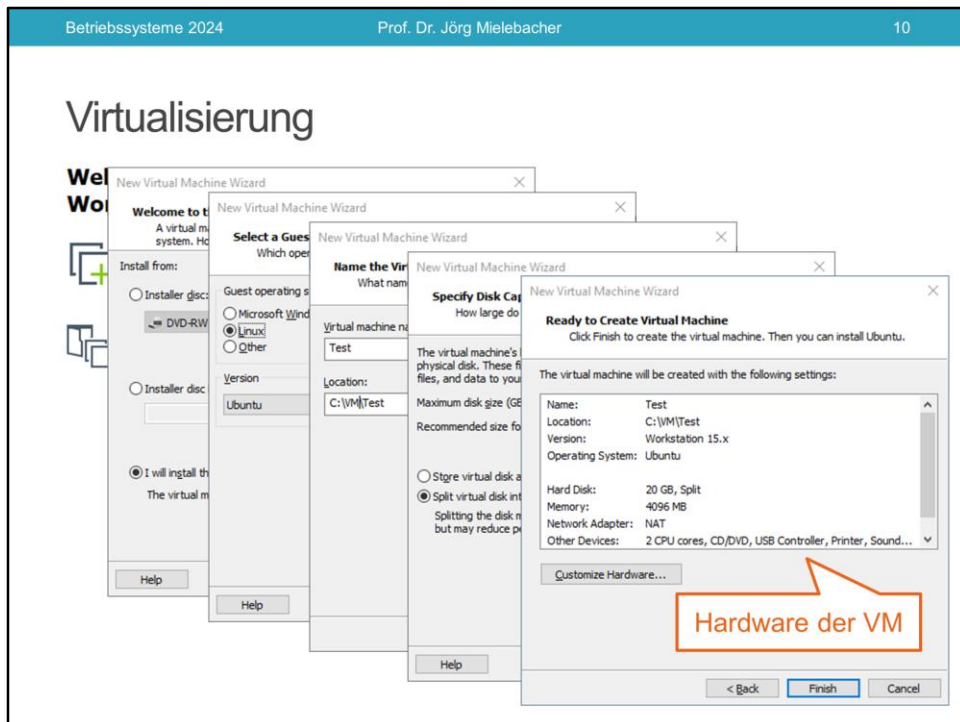
Virtualisierung

Virtualisierung



Virtualisierung erlaubt es allgemein, ein Gastbetriebssystem (z.B. Linux) in einem Host-Betriebssystem (z.B. Windows) auszuführen, wobei das Gastsystem in einer sog. virtuellen Maschine ausgeführt wird und auf virtualisierte Hardware zugreift.

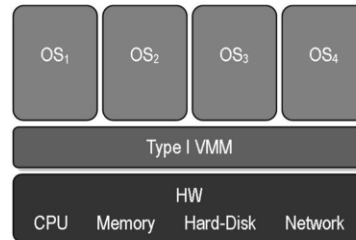
Verbreitete Werkzeuge im Desktop-Umfeld sind z.B. VMWare Workstation, VirtualBox und Parallels. Im Server-Umfeld ist besonders VMWare vSphere sehr verbreitet.



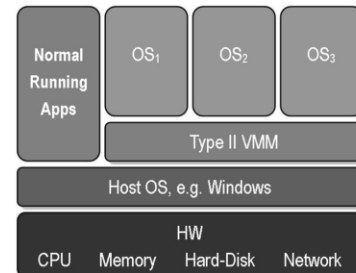
Das Anlegen einer virtuellen Maschine ist meist vergleichsweise einfach: Man weist die Installations-ISO-Datei (oder –Medium) zu, vergibt Name und Speicherort der virtuellen Maschine (bei VMWare: die VMX-Datei + zugehörige Festplatten-Dateien .vmdk), legt die Hardware fest (Festplatten, Prozessor, Hauptspeicher, Netzwerkinterface usw.). Anschließend beginnt die Installation des Gastbetriebssystems.

Hypervisor (Virtual Machine Monitor, VMM)

Typ 1-Hypervisor (Bare Metal)



Typ 2-Hypervisor (Hosted)



Die Virtualisierung basiert auf dem Einsatz eines sog. Hypervisors (Virtual Machine Monitor, VMM) und spezieller Maschinenbefehle, die die Prozessoren unterstützen. Allgemein unterscheidet man den Typ 1- und Typ 2-Hypervisor. Typ 1 (auch Bare Metal) bedeutet, dass der Hypervisor direkt auf der Hardware aufsetzt und in der Art eines sehr schlanken Betriebssystems den VMs den Zugriff auf die virtualisierte Hardware ermöglicht. VMWare vSphere (vormals ESXi) arbeitet auf diese Weise.

Ein Typ 2-Hypervisor (auch Hosted Hypervisor) wird als Anwendung innerhalb des Host-Betriebssystems ausgeführt und stellt darüber die Umgebung für die VMs bereit. Beispiele sind VMWare Workstation, VirtualBox und Parallels.

Anwendung und Nutzen

- Bessere Auslastung der Hardware
- Einfachere Wartung (Snapshots, Verschieben usw.)
- Software-Entwicklung
- Parallele Nutzung mehrerer Betriebssysteme

Erheblichen Nutzen brachte die Virtualisierung vor allem im Serverumfeld, da die Server-Hardware sich besser auslasten lässt. Auch lassen sich wichtige Wartungsaufgaben drastisch vereinfachen, z.B. Sicherung und Wiederherstellung durch sog. Snapshots, das Replizieren oder auch das Verschieben von VMs zwischen physischen Servern ohne Downtime.

In der Software-Entwicklung ist es möglich, mehrere Betriebssysteme parallel auszuführen, um auf unterschiedlichen Plattformen zu entwickeln und zu testen.

Auch für Anwender bringt die Virtualisierung Vorteile, wenn Anwendungen nur für ein bestimmtes Betriebssystem zur Verfügung stehen. Früher war hierfür umständliche Dual-Boot-Systeme notwendig.

Emulation

▪ Emulation

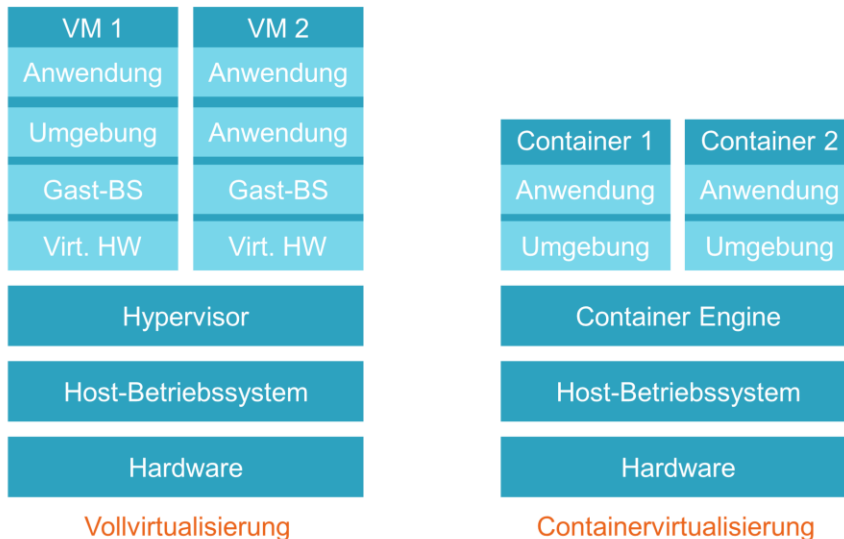
- Software bildet Hardware **vollständig** nach
- Beispiele: QEMU, DOSBox usw.

▪ Virtualisierung

- Software bildet Hardware **teilweise** nach
- Befehle werden direkt auf der Hardware ausgeführt
- Beispiele: VMWare ESXi/Workstation, VirtualBox, Parallels usw.

Häufig vermischt wird die Virtualisierung mit der Emulation. Bei der Emulation wird die Hardware vollständig durch Software nachgebildet. Beispiele hierfür sind z. QEMU und DOSBox. Bei der Virtualisierung wird die Hardware dagegen nur teilweise nachgebildet, während viele Befehle direkt (bzw. unter Nutzung des Betriebssystems) auf der Hardware ausgeführt werden.

Containervirtualisierung



Ein heute sehr wichtiger Ansatz ist die sog. Containervirtualisierung. Mit Werkzeugen wie Docker lassen sich Container erzeugen und ausführen, wobei Container Anwendungen und die notwendige Umgebung (Sprachen, Bibliotheken, Dienste usw.) enthalten. Bei der Vollvirtualisierung enthalten die einzelnen VMs jeweils ein vollständiges Gastbetriebssystem. Dies entfällt bei der Containervirtualisierung. Hier werden die Container lediglich gegenseitig isoliert und können z.B. durch das Dateisystem oder Netzwerkverbindungen miteinander kommunizieren. Die Container enthalten jedoch kein eigenes Betriebssystem.

Containervirtualisierung hat sich in den letzten Jahren sehr stark verbreitet und wird erfolgreich bei der Bereitstellung von Anwendungen und in der agilen Software-Entwicklung eingesetzt.

Beispiel: Docker

Hintergrund

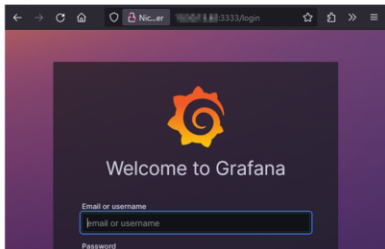
```
docker run -d -p 3333:3000 --name=grafana-test grafana/grafana-enterprise
```

Container starten

Ports Host/Container

Name

Image



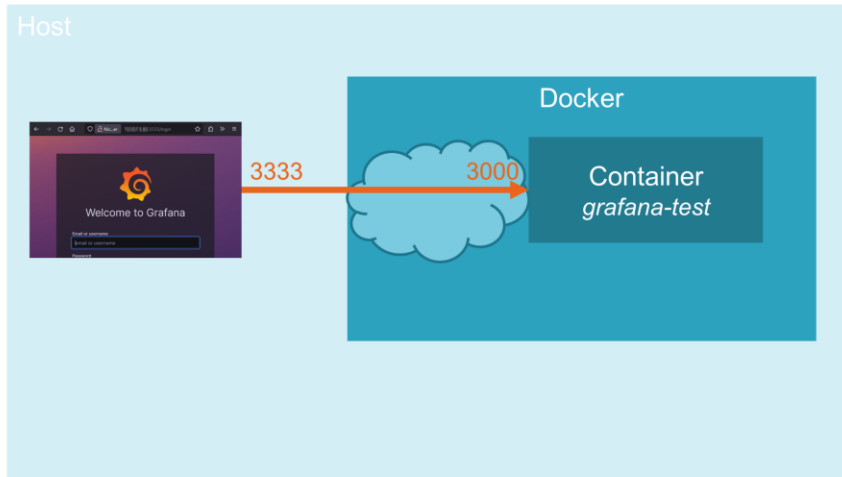
Docker ist für die Container-Virtualisierung besonders beliebt. Es steht für verschiedene Plattformen sowie mit und ohne grafische Oberfläche zur Verfügung. Für viele Anwendungen und Dienstprogramme gibt es bereits fertige Container, z.B. DBMS, Webserver oder auch komplexere Systeme.

Im hier gezeigten Beispiel wird ein Grafana-Container gestartet (Grafana ist ein webbasiertes Tool für die Visualisierung verschiedenster Daten). Der Container wird unter dem Namen `grafana-test` verwendet. Er nutzt das Container-Image `grafana-enterprise`. Beim Start des Containers wird hier auch festgelegt, dass der Container im Hintergrund ausgeführt wird (`-d`). Das von Docker aufgebaute Container-Netzwerk wird so konfiguriert, dass der Container vom Host-System über den Port 3333 erreichbar ist, der im Container auf Port 3000 abgebildet wird (`-p`).

Grafana wäre jetzt auf dem Hostsystem unter Port 3333 erreichbar. Allerdings müsste man den Container manuell erneut starten, wenn man keinen automatischen Neustart festlegt.

Ob der Container ausgeführt wird (bzw. welche Container ausgeführt werden), kann man mit dem Befehl `docker ps` überprüfen. Stoppen lässt sich der Container mit `docker stop grafana-test`, löschen mit `docker rm grafana-test`.

Beispiel: Docker



Deutlich wird das Zusammenwirken in dem gezeigten Schema.

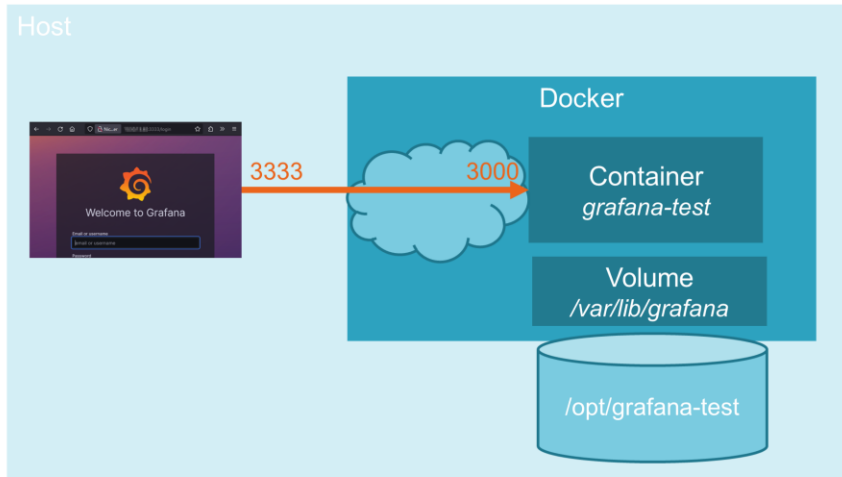
Beispiel: Docker

The diagram illustrates the mapping of host paths to container paths using Docker volumes. Three orange speech bubble boxes are positioned above the command: 'Volume' points to the volume specification, 'Host' points to the host path, and 'Container' points to the container path.

```
docker run -d -p 3333:3000 --volume "/opt/grafana-test:/var/lib/grafana" \
--name=grafana-test grafana/grafana-enterprise
```

Daten des Containers lassen sich in sog. Volumes ablegen. Das ist wichtig, wenn Daten des Containers persistent gespeichert werden sollen. Im hier gezeigten Beispiel wird ein Ordner des Hostsystems (`/opt/grafana-test`) auf einen Ordner des Containers (`/var/lib/grafana`) abgebildet.

Beispiel: Docker



Wie man sieht, wird das Verzeichnis `/var/lib/grafana` des Containers abgebildet auf das Verzeichnis `/opt/grafana-test` des Host-Systems abgebildet. Mit anderen Worten: Die Daten sind auch vom Host-System leicht einsehbar, z.B. wenn man Konfigurationen anpassen möchte.

Wo stehen wir?

Bisherige Themen

- Geschichte und Merkmale

Die Vorlesung Betriebssysteme begann mit einem historischen Abriss, der Entstehung und Entwicklung der Betriebssysteme in Bezug zu der Entwicklung der Computertechnik gesetzt hat. In diese Zuge wurden auch wichtige Begriffe und Merkmale von Betriebssystemen vorgestellt, darunter auch die typischen Aufgaben.

Aufgaben des Betriebssystems



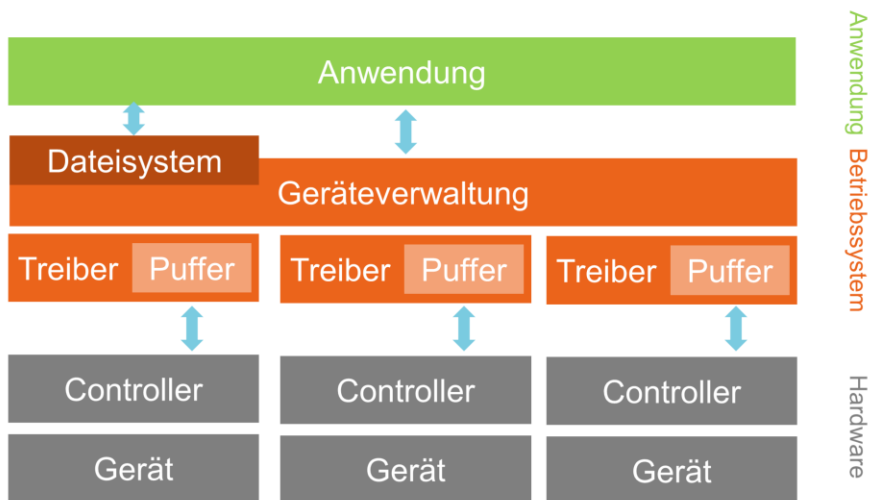
Zu den typischen Aufgaben gehört – wie seinerzeit beschrieben – die Verwaltung der Hardware, insbesondere das Abstrahieren von selbiger im Sinne einer virtuellen Maschine. Außerdem gehören zu den Aufgaben die Verwaltung der Prozesse, insbesondere auch das Scheduling und die Interprozesskommunikation sowie das Speichermanagement – einerseits des Hauptspeichers, andererseits aber auch der Zugriff auf das Dateisystem. Und schließlich sind Betriebssysteme heute typischerweise Mehrbenutzerbetriebssysteme, sodass auch die Verwaltung der Benutzer und die Zugriffskontrolle wichtige Aufgaben sind.

Bisherige Themen

- Geschichte und Merkmale
- Prozessverwaltung
- Speicherverwaltung
- Ein- und Ausgabe
- Dateisystem

Von diesen Aufgaben wurden die Prozessverwaltung, die Speicherverwaltung, die Ein- und Ausgabe (vor allem im Sinne der Geräteverwaltung) und das Dateisystem besprochen, die Benutzerverwaltung nur am Rande, z.B. am Beispiel von Berechtigungen.

Geräte und Dateisystem



In der Praxis sehr unmittelbar ist der Kontakt mit der Geräteverwaltung und dem Dateisystem.

Wichtige Erkenntnisse

Technik und Anforderungen treiben Entwicklung der Betriebssysteme.

Abstraktion verringert Komplexität.

Das Gewährleisten der Sicherheit gestaltet sich sehr aufwendig.

Die Vielfalt der Anwendungsszenarien wächst rapide.

Zusammenfassung

- RDP und VNC sind verbreitete Protokolle für den Remote-Zugriff auf Computer.
- Thin Clients nutzen RDP etc., um Anwendungen zu nutzen, die von einem Server bereitgestellt werden.
- Für die Remote Administration gibt es zahlreiche Lösungen, z.B. RDP, VNC, SSH und kommerzielle Tools.
- Virtualisierung bringt zahlreiche Vorteile.
- Man unterscheidet Typ 1 und Typ 2-Hypervisors.
- Emulation bildet die Zielhardware vollständig nach, Virtualisierung nur teilweise.
- Containervirtualisierung verpackt Anwendungen und ihre notwendige Umgebung in Container; diese enthalten aber kein Gastbetriebssystem.

Aufgaben

1. Wofür wird RDP verwendet?
2. Was ist ein Thin Client? Was ist notwendig, um ihn einzusetzen?
3. Nennen Sie Beispiel für Tools in der Fernwartung.
4. Nennen Sie Beispiele für Typ-1- und Typ-2-Hypervisor.
5. Worin liegen die Gemeinsamkeiten und Unterschiede von Vollvirtualisierung, Containervirtualisierung und Emulation?
6. Sie wollen MariaDB als Docker-Container nutzen. Was müssen Sie auf der Linux-Konsole hierfür machen?
7. Worin liegen die Vorteile eines benannten Volumes in Docker? Wie nutzt man es?