

Einführung in die IT

Teil 5:

Virtualisierung
Betriebssysteme
Interrupts

Bernd Schöner

DHBW Mannheim

Exkurs: Quantencomputing/1

IBM hat einen neuen Quantenprozessor vorgestellt, der deutlich leistungsfähiger als bisherige Systeme sein soll. Der Technologiekonzern teilte am Dienstag in Armonk (US-Bundesstaat New York) mit, der »Eagle« genannte Chip sei so komplex, dass er von traditionellen Supercomputern nicht nachgeahmt werden könne. Für eine Simulation durch konventionelle Hochleistungsrechner wären nach Darstellung des US-Konzerns »mehr klassische Bits notwendig, als es Atome in allen menschlichen Wesen auf dem Planeten gibt«.

Damit unterstreicht der IT-Konzern den elementaren Unterschied zwischen herkömmlichen Computern und Quantenrechnern. Traditionelle Computer arbeiten mit Bits, die jeweils nur einen von zwei Zuständen annehmen können: »Eins« oder »Null« beziehungsweise »An« und »Aus«. Quantencomputer hingegen verwenden sogenannte Qubits (»Quanten-Bits«), die nicht nur die zwei unterschiedliche Zustände darstellen können, sondern theoretisch unendlich viele unterschiedliche Zustände – und das gleichzeitig. Jedes zusätzliche Qubit, das ein Quantencomputer nutzen kann, verdoppelt die Anzahl der gleichzeitig darstellbaren Zustände. Daher gilt die Zahl der Qubits derzeit als Leistungsmerkmal für die neuartigen Rechner. Allerdings wird die Leistung eines Quantencomputers auch von etlichen weiteren Faktoren bestimmt.

Wirtschaft und Wissenschaft erhoffen sich von Quantencomputern enorme Fortschritte, etwa bei der Entwicklung neuer Arzneimittel. Zudem könnte sie zum einen abhörsichere Datenübertragungen ermöglichen, zum anderen genutzt werden, um komplexe Verschlüsselungen zu knacken. Noch sind Quantencomputer aber in erster Linie Forschungsobjekte, eine kommerzielle Nutzung ist bislang kaum möglich.

<https://www.spiegel.de/netzwelt/gadgets/quantencomputer-eagle-von-ibm-zu-komplex-fuer-supercomputer-a-2b2e703d-ba94-48ce-8140-94618118bd78>

Exkurs: Quantencomputing/2

Bislang nur ein Forschungsthema

Mit dem neuen Chip hat IBM einen weiteren Schritt [auf seiner Roadmap](#) zur Entwicklung real nutzbarer Quantencomputer geschafft. Das Unternehmen betont, dass es mit »Eagle« erstmals die Schwelle von 100 Qubits überschreite, der neue Chip verfügt über 127 Qubits.

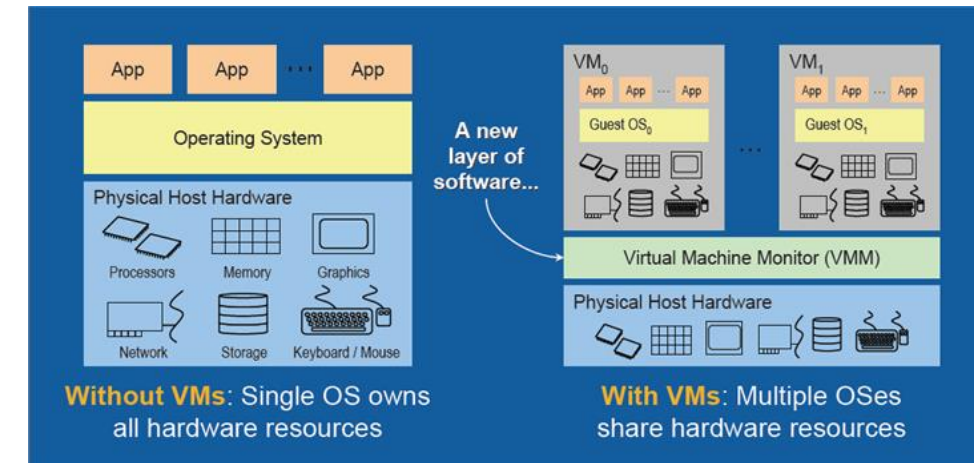
IBM hat die Anzahl der Qubits in seinen Quantenprozessoren in den vergangenen Jahren kontinuierlich in die Höhe geschraubt. Das ermögliche es Nutzern, bei Experimenten und Anwendungen Probleme auf einem neuen Komplexitätsniveau zu erforschen, betonte IBM. Dazu gehöre die Optimierung des maschinellen Lernens und die Modellierung neuer Moleküle und Materialien für den Einsatz in Bereichen, die von der Energiewirtschaft bis zur Arzneimittelentwicklung reichten.

»Die Einführung des »Eagle«-Prozessors ist ein wichtiger Schritt auf dem Weg zu dem Tag, an dem Quantencomputer klassische Computer in bedeutendem Umfang übertreffen können«, sagte IBM-Forschungschef Darío Gil und schwärmte: »Quantencomputer haben das Potenzial, nahezu jeden Sektor zu verändern und uns dabei zu helfen, die größten Probleme unserer Zeit anzugehen.«

Die »Canary«-Quantenchips von IBM aus dem Jahr 2017 verfügten über fünf Qubits. Im Februar 2020 stellte IBM die »Falcon«-Serie mit 27 Qubits vor. Im September 2020 schraubte der Konzern die Marke mit dem »Hummingbird«-Prozessor auf 65 Qubits hoch. Im Jahr 2023 will der Konzern einen Chip mit mehr als 1000 Qubits betriebsbereit haben. Den ersten Quantencomputer in Deutschland betreibt die Fraunhofer-Gesellschaft gemeinsam mit IBM in einem Rechenzentrum des Konzerns in Ehningen bei Stuttgart. An dem »Q System One« – ebenfalls von IBM – sollen die Technologie an sich, ihre Anwendungsszenarien und Algorithmen erforscht werden. Dieses System nutzt 27 Qubits.

Virtualisierung

- Ausgangsposition: die IT-Infrastruktur wird immer größer und komplexer, die Rechner immer leistungsfähiger.
- Rechnervirtualisierung
- Speichervirtualisierung
 - Abstraktion der physischen Speicherform eines Datenspeichers wird als Speichervirtualisierung bezeichnet.
- Netzwerkvirtualisierung
 - Ein logisches Teilnetz innerhalb eines Switches bzw. eines gesamten physischen Netzwerks.
- Betriebssystemvirtualisierung mittels OS-Container



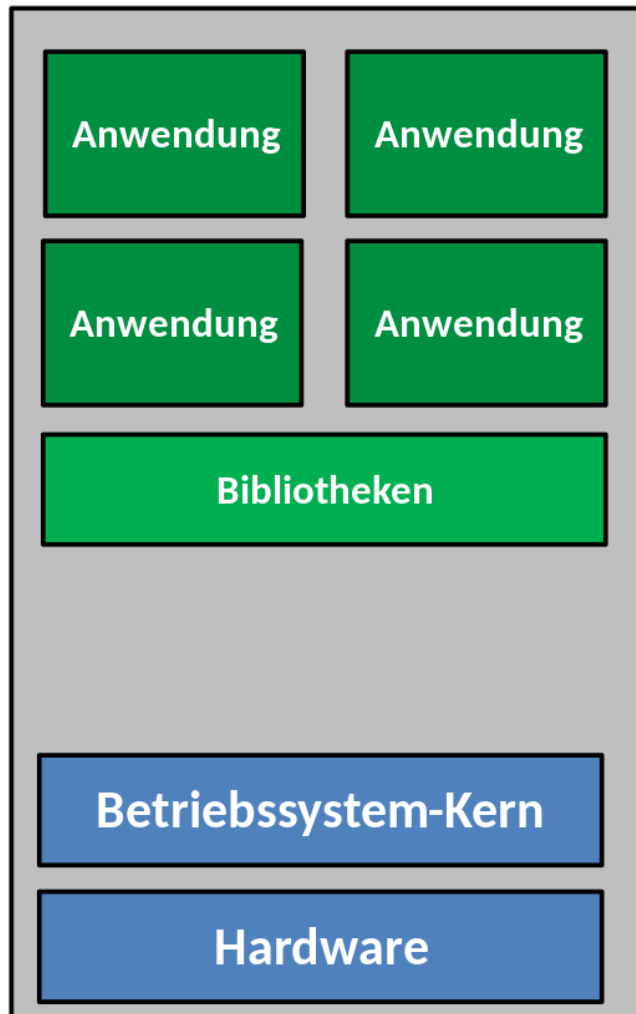
Virtualisierung/1

- Virtualisierung bezeichnet die Nachbildung eines Hard- oder Software-Objekts durch ein ähnliches Objekt vom selben Typ mit Hilfe eines Abstraktions-Layers.
- Auf einem physischen Rechner laufen mehrere logische Rechner mit oder ohne eigenes Betriebssystem.
- Computer-Ressourcen (besonders im Server-Bereich) können transparent zusammengefasst oder aufgeteilt werden, z.B.
 - Ein logisches Teilnetz innerhalb eines Switches bzw. eines gesamten physischen Netzwerks.
 - Speicherbereiche eines grossen Speichersystems
 - Mehrere Betriebssysteme, bzw. Anwendungen auf einem Server
- Virtualisierung mittels Container

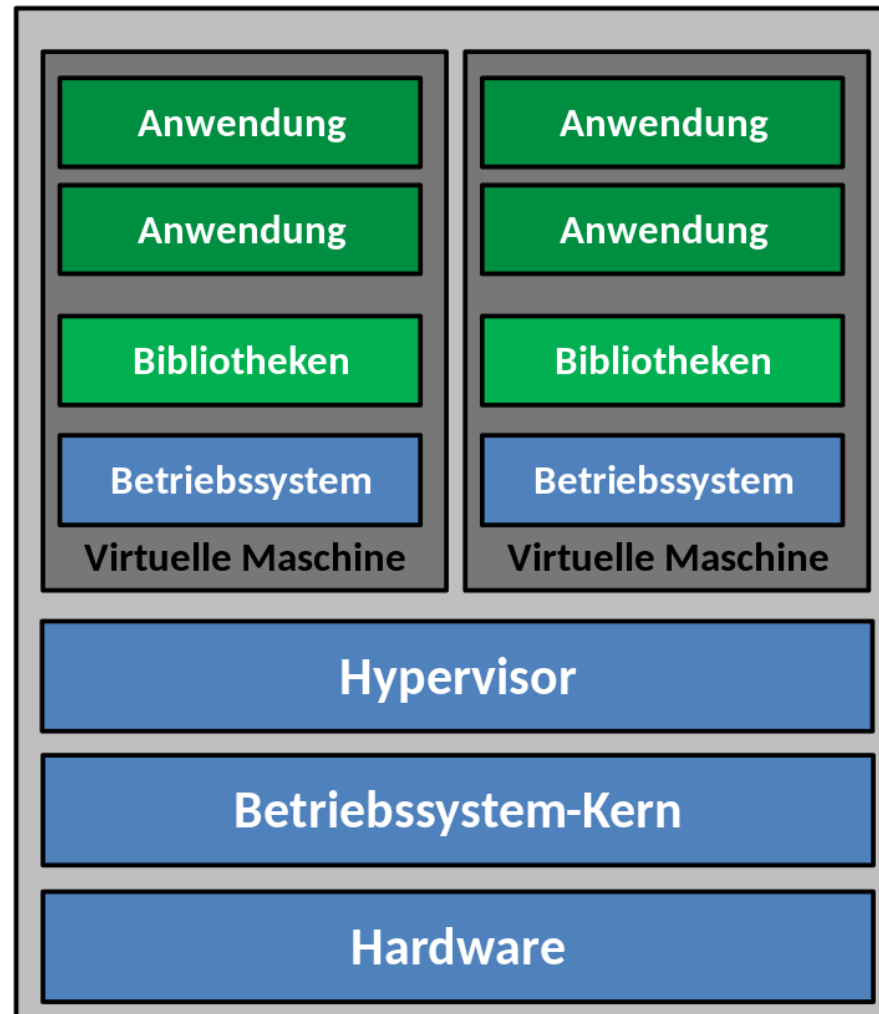
Virtualisierung/2

- Virtualisierung mittels Hypervisor: wird auch Virtual Machine Monitor (VMM) genannt. Er sorgt dafür, dass auf einer realen Hardware eine oder mehrere virtuelle Maschinen ausgeführt werden können Auf einem physischen Rechner laufen mehrere logische Rechner.
- Computer-Ressourcen (besonders im Server-Bereich) können transparent zusammengefasst oder aufgeteilt werden.
- Eine virtuelle Maschine (VM) ist eine Art virtueller Computer mit seiner virtuellen Hardware und den virtuellen Betriebsmitteln. Beim Erstellen die Betriebsmittel definiert, wie Hauptspeicher, CPUs oder Festplattenplatz Netzwerk, etc.
- Dynamische Zuordnung der Ressourcen über die Zeit hinweg.
- Jede VM hat ihr eigenes Betriebssystem.
- Wichtige Funktion sind auch Klonen und Snapshots.

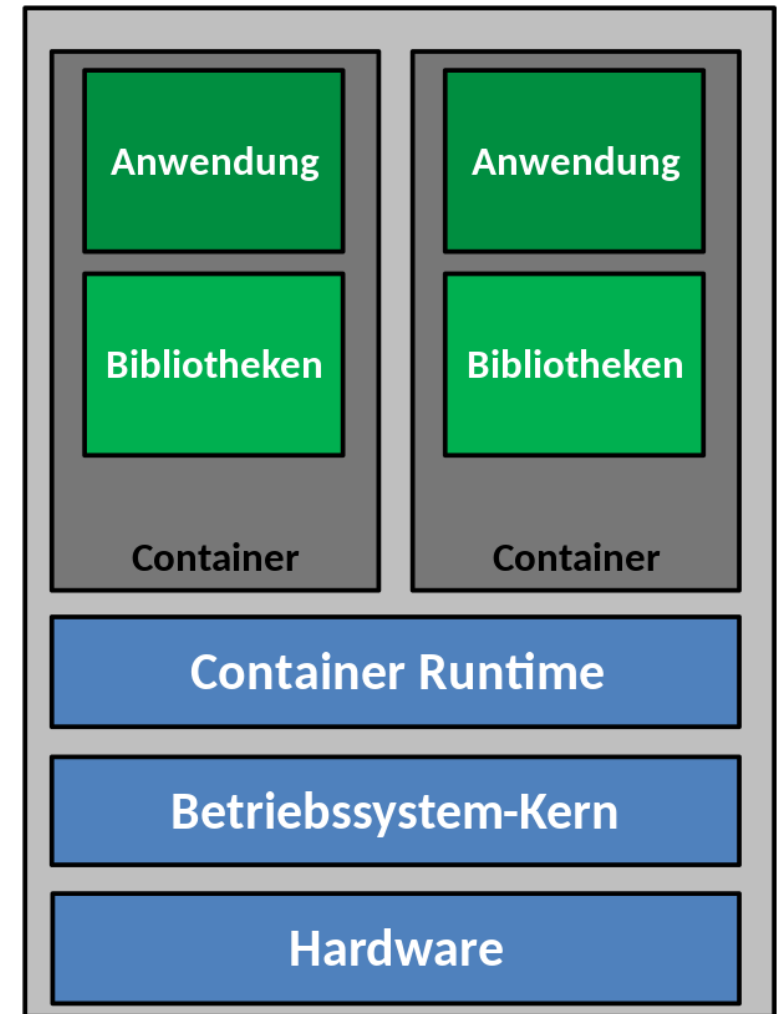
Vergleich traditioneller Ansatz vs. virtuelle Maschinen vs. Containerisierung von Anwendungen



a) Traditionelles System



b) Virtuelle Maschinen



c) Containervirtualisierung

Virtualisierung mit Containern/1

- Container in der IT sind vergleichbar mit Frachtcontainern aus der Logistik. Die Container in der IT werden mit einer Anwendung und allen dazugehörigen Komponenten gefüllt, so dass diese Anwendung jederzeit und ortsunabhängig ausgeführt werden kann.
- Container (etwa seit 2013) sind eine leichtgewichtige Form der Virtualisierung. Im Gegensatz zur Virtualisierung mittels eines Hypervisors hat Containervirtualisierung zwar einige Einschränkungen in der Art ihrer Gäste. Sie sind von einander isoliert, genau wie virtuelle Maschinen. Die Container nutzen den Kernel des OS und gelten aber als besonders ressourcenschonend.
- Ein Container ist ein Softwarepaket, das alles Wichtige zum Ausführen von Software enthält: Code, Laufzeit, Konfiguration und Systembibliotheken, damit das Programm auf jedem Hostsystem ausgeführt werden kann.
- Die Container Plattform läuft immer auf einem Host OS. Deshalb sind Container deutlich kleiner als virtuelle Maschinen,
- Container enthalten die Programme, Bibliotheken und die Anwendung. Kein OS. Ein Image enthält alles was benötigt wird, um eine Anwendung ausführen zu können.
- Images können auf andere Rechner portiert werden, allerdings nur auf gleichen Plattformen.

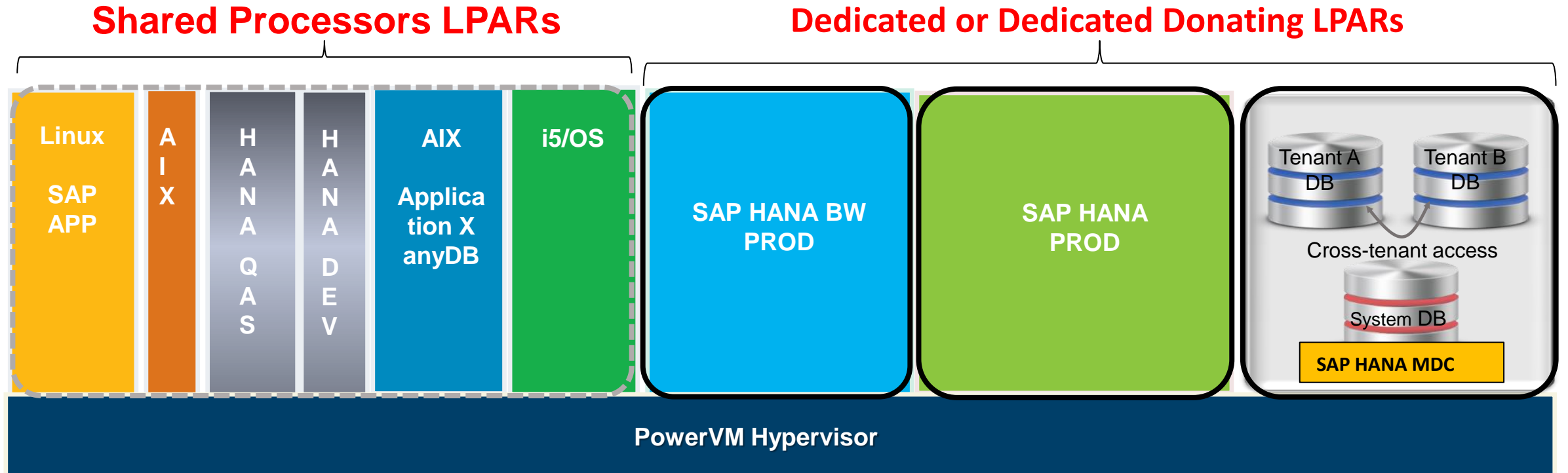
Virtualisierung mit Containern/2

- Docker (2013) ist eine Containervirtualisierungsplattform, die viele der Techniken einsetzt und um benutzerfreundliche Werkzeuge und Dienste ergänzt. Dazu gehört beispielsweise eine Beschreibung von Images (Dockerfiles) oder ein Repository, das solche Images verwaltet.
- Im Unterschied zu einer Virtuellen Maschine teilen sich Container und Host einen gemeinsamen Betriebssystem-Kernel. Dies verbessert einerseits die Leistung erheblich, vergrößert andererseits aber auch das Risiko, dass erfolgreiche Angriffe gegen den Kernel auch den Host kompromittieren.
- Kubernetes ist ein Open-Source-System zur Automatisierung der Bereitstellung, Skalierung und Verwaltung von Container-Anwendungen, das ursprünglich von Google (2014) entworfen wurde.
- Eine „Plattform für das automatisierte Bespielen, Skalieren und Warten von Anwendungscontainern auf verteilten Hosts“. Es unterstützt eine Reihe von Container-Tools, einschließlich Docker. Kubernetes übernimmt das Bereitstellen (Deployment) der Container und überwacht deren Status.
- Die OpenShift Container Platform ist die lokale private Plattform von Red Hat als Serviceprodukt, die auf einem Kern von Anwendungscontainern aufgebaut ist, die von Docker bereitgestellt werden.
- Die OpenShift Container Platform stellt Kubernetes-Umgebungen für Unternehmen zur Verfügung, die zum Erstellen, Bereitstellen und Verwalten von containerbasierten Anwendungen auf jedem Rechenzentrum auf denen Red Hat Enterprise Linux unterstützt wird. Im Vergleich zu virtuellen Maschinen enthalten Container kein eigenes Betriebssystem und benötigen weniger Ressourcen. Man spricht von einer Virtualisierung von Anwendungen, da es sich nicht um eine reine Betriebssystemvirtualisierung handelt.
- Anwendungen wie SAP und Container?

Vorteile der Virtualisierung

- Verbesserte Systemauslastung
 - Weniger physikalische Resources (Platz, Power, CPU, Adapter, etc.)
 - System/Server Kosten werden reduziert, weil Wartung und Betrieb weniger Geräte betreuen müssen.
 - Virtualisierungstechnologie reduziert die Betriebsprozesse
 - Kosten eines Data Centers werden weniger, weil weniger Resources benutzt werden.
 - Antwortzeiten können stabil gehalten werden, kurzzeitige Lastspitzen werden einfach abgefangen.
 - LPARs/VMs können verändert werden bezüglich Cores und Memory (grosser oder kleiner)
- Speicher Virtualisierung unterstützt Speichersysteme verschiedener Hersteller.

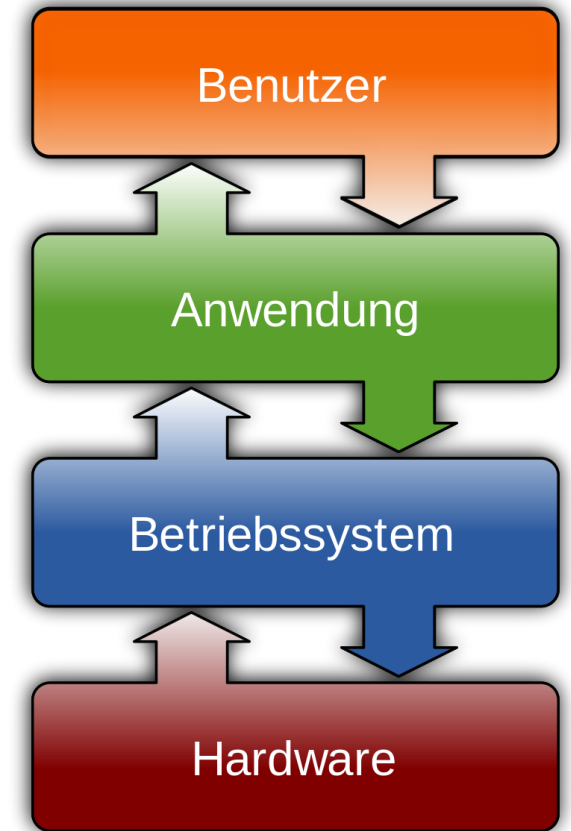
Beispiel: PowerVM und SAP HANA Virtualisierung



Dedicated Donating heist, dass unbenutzte CPUs einem Shared Prozessor Pool zur Verfügung gestellt werden. Solche CPUs werden sofort (innerhalb von Microsekunden) wieder zurück gegeben, wenn sie wieder gebraucht werden.

Das Betriebssystem/1

- „Das Betriebssystem wird gebildet durch die Programme eines digitalen Rechensystems, die zusammen mit den Eigenschaften der Rechanlage die Grundlage der möglichen Betriebsarten des digitalen Rechensystems bilden und insbesondere die Ausführung von Programmen steuern und überwachen.“
 - – DIN 44300
- **Betriebssystem** (OS: Operating System): eine Zusammenstellung von Programmen, die die Systemressourcen eines Computers, wie Arbeitsspeicher, Festplatten, Netzwerk, Ein- und Ausgabegeräte verwaltet und diese Anwendungsprogrammen zur Verfügung stellt.
 - Spezifisch für eine Hardware Architektur
 - Varianten, z.B. zOS, Windows, Linux, MacOS, Unix Varianten
- Das Betriebssystem bildet die Schnittstelle zwischen der Hardware und der Anwendungssoftware. Betriebssysteme bestehen in der Regel aus einem Kernel (deutsch: Kern), der die Hardware des Computers verwaltet, sowie speziellen Programmen, die beim Start unterschiedliche Aufgaben übernehmen, z. B. das Laden von Gerätetreibern. Die Interpretation ist in der Literatur nicht eindeutig.
- Verschiedene Arten von Betriebssystemen:
 - **Für allgemein zur Informationsverarbeitung verwendete Computersysteme**
 - Spezialisierte Betriebssysteme, oft mit Echtzeitfähigkeit, z.B. Spielecomputer, Mobiltelefone, Maschinen der Maschinenbaubranche, Flugzeuge, Autos u. v. a..



Das Betriebssystem - Abstraktion

- Abstraktion:
 - Abstraktion der Maschine
 - Verbergen der spezifischen Hardware Architektur und der Geräte vor dem Anwender
 - Verbergen der Komplexität der Maschine vor dem Anwender
 - Vereinfachung der Programmierung von Anwendungen
 - Abstraktion des Maschinenbegriffes
 - Reale Maschine = Zentraleinheit + Geräte (Hardware)
 - Abstrakte Maschine = Reale Maschine + Betriebssystem
 - Benutzermaschine = Abstrakte Maschine + Anwendungsprogramm
- Geräte
 - Geräte aus der Sicht eines Betriebssystems sind alle, die über Ein-/Ausgabekanäle angesprochen werden. Dazu gehören auch interne Erweiterungen wie Grafikkarten, Netzwerkkarten und anderes.
 - Die (Unter-)Programme zur Initialisierung und Ansteuerung dieser „Geräte“ bezeichnet man zusammenfassend als Gerätetreiber (Device Driver).

Das Betriebssystem - Aufgaben

- Aufgaben des OS:
 - Speicherverwaltung
 - Verwaltung der Systemressource Hauptspeicher.
 - Protokollierung der Speichernutzung.
 - Reservierung und Freigabe von Speicher.
 - (Prozess)-Verwaltung
 - Überwachung der Speicherzugriffe und gegebenenfalls Beenden von Prozessen bei einer Schutzverletzung
 - Erzeugung neuer Prozesse (entweder auf Anforderung des Betriebssystems oder auf Aufforderung anderer schon existierender Prozesse) und Reservierung des von den Prozessen benötigten Speichers.
 - Kommunikation und Synchronisation von Prozessen untereinander (Interprozesskommunikation)
 - Geräte- und Dateiverwaltung
 - Effiziente Zuweisung von Ein-/Ausgabegeräten und Vermittlungseinheiten (Datenkanäle, Steuereinheiten), Vermeidung von Konflikten
 - Initiierung, Überwachung der Ausführung, Terminierung von Ein-/Ausgabevorgängen
 - Verwaltung des Dateisystems. Erzeugung eines Namensraums mit zugehörigen Speicherobjekten und gegebenenfalls weiteren Objekten.
 - Rechteverwaltung
 - Voneinander unabhängige Benutzer/Programme dürfen sich gegenseitig nicht stören.
 - Schutzfunktionen z. B. durch Zugriffsbeschränkungen

Das Betriebssystem – Überblick über Charakteristika und Services

- Dialog- und Stapelverarbeitung
- Single-User, Multi-User
- OS Prozess Konzept
- Multitasking - Multiprocessing
- Interrupts (Unterbrechungen)
- Memory Management and Protection, Virtual Memory

Dialog- und Stapelverarbeitung

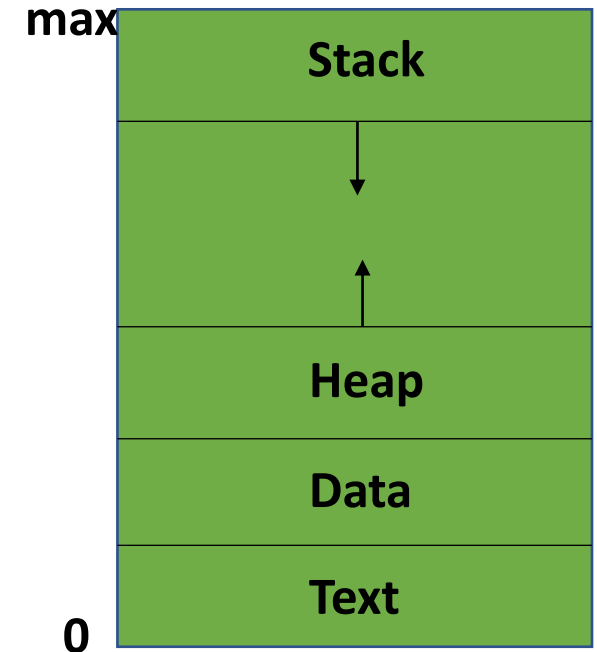
- Die vorherrschende Betriebsart auf den ersten großen Computern der 1950er und 1960er Jahre war zunächst die Stapelverarbeitung (Batch Processing). Ein auszuführendes Programm zusammen mit Steueranweisungen als Job an das Betriebssystem übergeben. Anfänglich wurden die Jobs in Form von Lochkarten erstellt, so dass tatsächlich ein Stapel (engl. Batch) Lochkarten von dem Betriebssystem verarbeitet wurde.
- Interaktive Eingriffe waren dabei nicht möglich, alle Eingabedaten mussten in Dateien vorab bereitgestellt werden. Ergebnisse standen erst nach der vollständigen Abarbeitung des gesamten Jobs zur Verfügung. Das galt übrigens auch für Programmierfehler.
- Auch heute noch werden Batch-Programme geschrieben, diese erledigen Aufgaben, die keine Interaktion mit dem Benutzer erfordern, z. B. die Erstellung und der Druck der Rechnungen bei einem Telekommunikations-Unternehmen am Monatsende.
- Im Unterschied zur Stapelverarbeitung werden im Dialogbetrieb (Interactive Processing) Programme interaktiv in einem Dialog durch den Benutzer gestartet.
- Auch während der Verarbeitung sind Eingaben von Daten und Steuerkommandos sowie Ausgaben von Zwischenergebnissen möglich. Der Benutzer interagiert über eine textuelle oder grafische Oberfläche mit den Programmen und dem Betriebssystem.

Single-User, Multi-User

- **Single-User:** ein Anwender kann das System nutzen und ein Programm starten
 - z.B. MS DOS, Windows, spezialisierte Betriebssysteme
- **Multi-User:** mehrere Benutzer können gleichzeitig mit demselben Rechner arbeiten.
- **Mehrprogramm-Betrieb** (Multiprogramming oder Multitasking): mehrere Programme eines oder mehrerer Benutzer können ‚gleichzeitig‘ ausgeführt werden.
 - Als erstes Mehrnutzer-Betriebssystem wurde bei IBM bereits Mitte der 1960er Jahre OS/360 eingeführt.
- **Teilnehmer- und Teilhaberbetrieb:**
 - Teilnehmerbetrieb: die einzelnen Nutzer arbeiten in der Regel unterschiedlichen Programmen
 - Teilhaberbetrieb: auf großen Servern arbeiten sehr viele Benutzer gleichzeitig am selben Programm und nutzen insbesondere dieselben Datenbestände.
 - in der Regel wird eine Software oberhalb des Betriebssystems benötigt, die zusätzliche Betriebsmittelverwaltung übernimmt.
 - diese Software wird auch Applikationsserver genannt. Beispiele sind Java EE Server oder SAP Appserver.

Betriebssystem Prozesse/1

- Ein Prozess stellt auf einem Rechnersystem die Ablaufumgebung für ein Programm zur Verfügung und ist die Instanziierung eines Programms, ergänzt um weitere Verwaltungs-Informationen und Ressourcenzuteilungen für die Ausführung.
- Er wird vom Betriebssystem dynamisch kontrolliert durch bestimmte Aktionen, mit denen das Betriebssystem ihn in entsprechende Zustände setzt.
- Als ablaufendes Programm beinhaltet ein Prozess dessen Anweisungen – eine dynamische Folge von Aktionen, die entsprechende Zustandsänderungen bewirken.
- Als Prozess bezeichnet man auch die gesamte Zustandsinformation eines laufenden Programms.
- Eine nebenläufige Ausführungseinheit innerhalb eines Prozesses wird Thread genannt. Oft werden nun nicht mehr Prozesse nebenläufig ausgeführt, sondern nur die Threads innerhalb eines Prozesses.
- Unterschied zwischen Programm und Prozess!



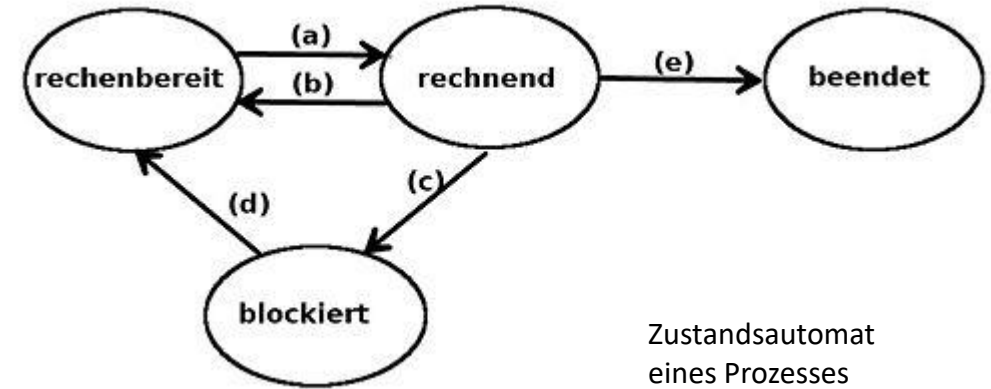
Der Aufbau eines Prozesses im Hauptspeicher mit Programmsegment (*text*), Datensegment (*data*), Heap und Stack. Im Zusammenhang mit den Speicheradressen, die ein laufendes Programm als Prozess nutzen darf, spricht man auch von einem Prozessadressraum.

Betriebssystem Prozesse/2

- Das Betriebssystem teilt einem Prozess bestimmte Ressourcen zu, ein ausreichender Anteil des Hauptspeichers, in dem der entsprechende Prozess ausgeführt werden soll.
- Ein Prozess wird gestartet, eine ausführbare Datei wird vom Betriebssystem in den Adressraum eines Prozesses geladen.
- Die Prozesse werden vom Prozess-Scheduler des Betriebssystems verwaltet.
- Ein Prozessor(kern) kann immer nur einen Prozess gleichzeitig verarbeiten. Multi-Programming.
- Ein Prozess beinhaltet insbesondere:
 - den Wert des Befehlszählers,
 - die Inhalte der zum Prozess gehörenden Prozessor-Register,
 - das Programmsegment, das den ausführbaren Code des Programms enthält,
 - das Stack-Segment, das temporäre Daten wie Rücksprungadressen und lokale Variablen enthält,
 - das Datensegment, das globale Variablen enthält, und
 - unter Umständen einen Heap, der dynamisch angeforderten Speicher umfasst.
- Im Zusammenhang mit den Speicherbereichen, die ein laufendes Programm als Prozess nutzen darf, spricht man auch von einem Prozessadressraum. Ein Prozess kann nur Daten verarbeiten, die zuvor in seinen Adressraum geladen wurden; Daten. Der Adressraum eines Prozesses ist im Allgemeinen virtuell; der virtuelle Speicher bezeichnet, er wird einem Prozess vom Betriebssystem zur Verfügung gestellt wird.

Betriebssystem Prozesse/3

- Ein Prozess durchläuft somit während seiner Lebenszeit verschiedene Zustände:
 - *Rechnend (running, auch aktiv)*: Der Prozess wird in diesem Moment auf der CPU ausgeführt, d. h. die Programm-Befehle werden abgearbeitet.
 - (Rechen)bereit *ready*: Im Zustand *bereit* befinden sich Prozesse, die gestoppt wurden, um einen anderen Prozess rechnen zu lassen. Sie können theoretisch ihren Ablauf fortsetzen und warten nun darauf, dass ihnen die CPU wieder zugeteilt wird. Ein neuer Prozess erzeugt wird, tritt dieser zunächst in den Zustand *bereit* ein.
 - Blockiert (*blocked*): Prozesse im Zustand *blockiert* warten auf bestimmte Ereignisse, die für den weiteren Prozessablauf notwendig sind. Beispielsweise sind E/A-Geräte im Vergleich zur CPU nur sehr langsam arbeitende Komponenten; hat ein Prozess das Betriebssystem mit einer E/A-Geräte-Kommunikation beauftragt, so muss er warten, bis das Betriebssystem diese abgeschlossen hat.
 - Beendet (*terminated*): Der Prozess hat seine Ausführung beendet, das Betriebssystem muss noch „aufräumen“.



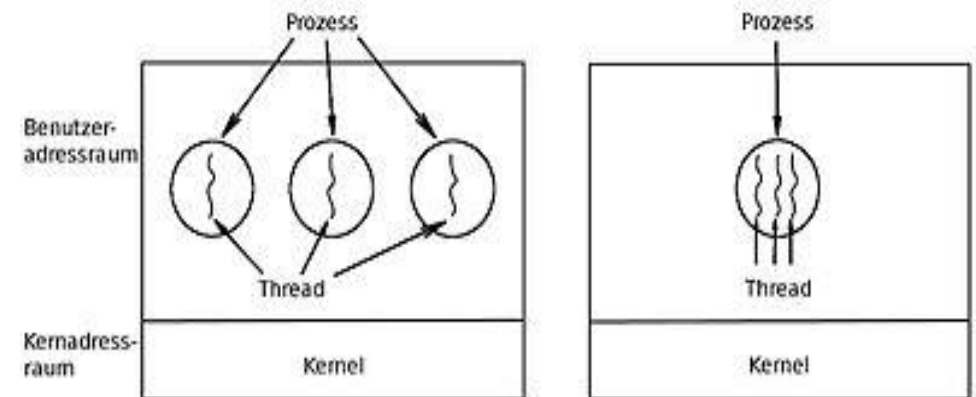
Betriebssystem Prozesse/4

- Prozessverwaltung:
 - Jeder Prozess wird durch einen **Prozesskontrollblock** (*process control block*, PCB) dargestellt. Der Prozesskontrollblock beinhaltet viele Informationen, die mit einem bestimmten Prozess verbunden sind.
 - Der Process identifier (PID, auch Prozess-ID): zur eindeutigen Identifikation von Prozessen dient.
 - Der Prozessstatus: rechenbereit, rechnend, blockiert oder beendet.
 - Der Inhalt des Befehlszählers, der die Adresse des nächsten auszuführenden Befehls enthält.
 - Die Inhalte aller anderer CPU-Register, die dem Prozess zur Verfügung stehen/mit ihm zusammenhängen.
 - Scheduling-Informationen: z.B. die Priorität des Prozesses, Zeiger auf die Warteschlangen und weitere Scheduling-Parameter.
 - Informationen für die Speicherverwaltung.
 - Buchhaltung: Das Betriebssystem führt auch Buch über einen Prozess
 - E/A-Statusinformationen: Liste der E/A-Geräte (bzw. Ressourcen), die mit dem Prozess verbunden sind, auch offene Dateien
 - Zugriffs- und Benutzerrechte
 - **Prozesstabelle:** eine Tabelle mit aktuellen Prozessen in einer eigenen Datenstruktur, der sog. Prozesstabelle.
 - Bei der Erzeugung eines neuen Prozesses wird darin ein Prozesskontrollblock als neuer Eintrag angelegt. Da PCBs Informationen über die belegten Ressourcen enthalten, müssen sie im Speicher direkt zugreifbar sein.

Threads

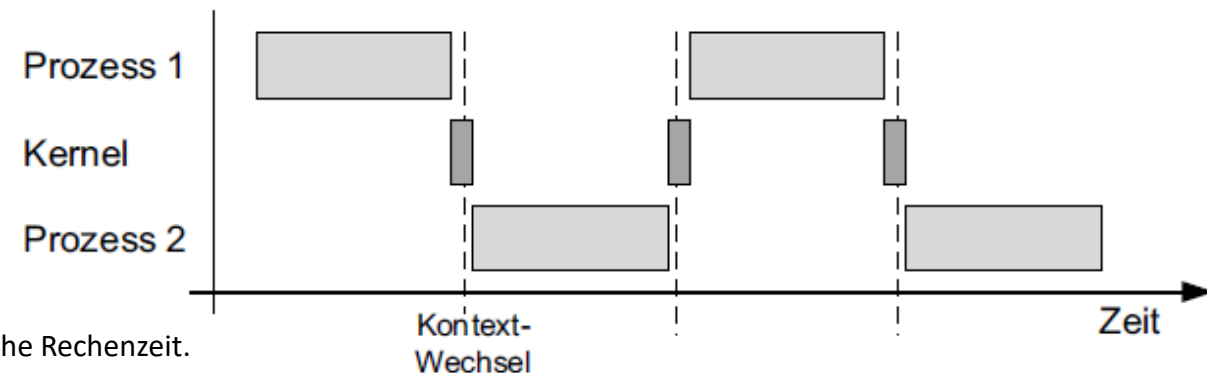
- Verwaltung von Prozessen ist relativ aufwändig, ‚heavy-weight‘.
- Moderne Betriebssysteme unterstützen deshalb Threads als ‚light-weight‘ Prozesse (10-100-mal schneller). Es ist ein Abarbeitungslauf (vom Programmierer bestimmt) innerhalb eines Prozesses.
- Threads in Programmiersprachen, z.B. Java.
- Threads existieren innerhalb von Prozessen und teilen sich deren Ressourcen, z.B. CPU, Dateien, Netzwerk.
- Ein Prozess hat seinen eigenen Adressraum, die Threads können auf den Speicher des Prozesses zugreifen.
- Threads eines Prozesses sind aber nicht gegeneinander geschützt und müssen sich daher beim Zugriff auf die gemeinsamen Prozess-Ressourcen abstimmen (synchronisieren), auch Anwendungen, um z.B. Ergebnisse zu sammeln.
- Jeder Prozess verwaltet einen privaten Thread-Kontrollblock (analog zum PCB)

Beispiel: Prozess mit einem Thread und ein Prozess mit drei Threads.



Multi-tasking – Multi-processing

- Ein Prozess wird gestartet, eine ausführbare Datei wird vom Betriebssystem in den Adressraum eines Prozesses geladen.
- Beim Programmbetrieb unterscheiden wir zwischen:
 - Beim Einprogrammbetrieb (**single-tasking**) werden einzelne Tasks von der Zentraleinheit nacheinander bearbeitet. Es befindet sich jeweils nur ein Task im Arbeitsspeicher, der für seinen gesamten Ablauf alle vorhandenen Betriebsmittel zugeteilt erhält.
 - Beim Mehrprogrammbetrieb (**multi-tasking**) können mehrere Tasks von der CPU *quasi* zeitgleich bearbeitet werden. Es befinden sich gleichzeitig mehrere Tasks ganz oder teilweise im Arbeitsspeicher, denen das Betriebssystem bei der Ausführung die benötigten Betriebsmittel abwechselnd zuteilt. Die CPU Rechenzeit muss daher unter den Prozessen verteilt werden (Zeitmultiplex).
 - zwei Verfahren beim Multitasking: nicht-unterbrechend (non-preemptive) und unterbrechend (preemptive).
 - Beim preemptive Multitasking unterbricht die CPU regelmäßig über einen Interrupt eines Zeitgebers den gerade laufenden Prozess und entscheidet welcher Prozess weitergeführt wird.
 - Die Auswahl eines Prozesses erfolgt durch den Scheduler des Betriebssystems, die Ausführung durch den Dispatcher.
- Bei mehreren CPUs oder Cores, können mehrere Prozesse tatsächlich parallel ausgeführt werden (**Multi-processing**).



Beispiel: 2 Prozesse, beide erhalten die gleiche Rechenzeit.

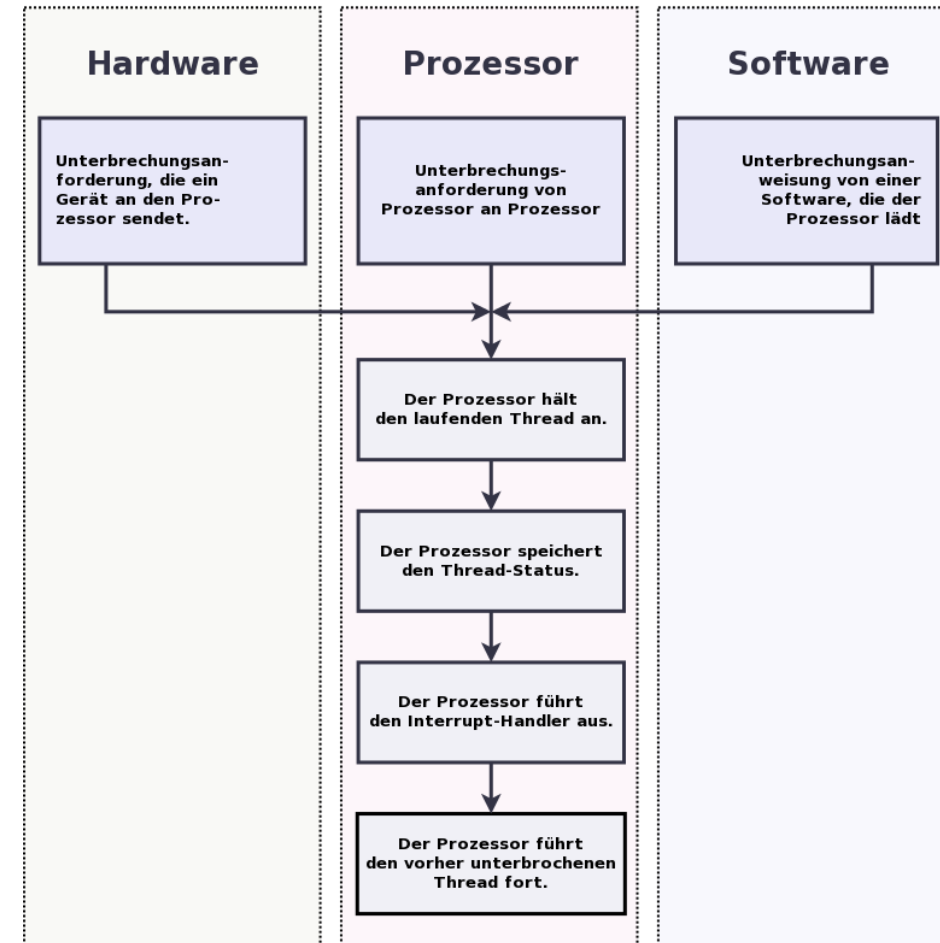
Multi-tasking Strategien

- Die Verteilung von Rechenzeit auf der/den CPU(s) kann abhängig vom Anwendungsbereich nach unterschiedlichen Strategien erfolgen. Drei wesentliche Anwendungsbereiche sind:
 1. Batch-orientierte Systeme (Stapelverarbeitung): Bei batch-orientierten Systemen findet keine direkte Interaktion mit Benutzern statt. Batch Programme laufen oft relativ lange. Über das Scheduling wird versucht, einen möglichst hohen Durchsatz an verarbeiteten Daten zu erzielen.
 2. Dialogorientierte Systeme: Ein oder mehrere Benutzer interagieren direkt mit dem System. Sie erwarten, dass das System unmittelbar auf Eingaben reagiert. Über das Scheduling wird versucht, die Antwortzeiten möglichst kurz zu halten (bei heutigen Systemen ist das meist sub-second für die meisten Interaktionen).
 3. Echtzeitanforderungen: Prozesse müssen ihre Zeitzusagen (Deadlines) einhalten, d. h. Ergebnisse müssen innerhalb eines bestimmten Zeitintervalls vorliegen. Hier muss über das Scheduling entsprechend agieren, um die Deadlines einzuhalten. Eventuell muss das Scheduling sogar so gebaut sein, dass das Zeitverhalten des Systems vorhersagbar wird.
- Dispatcher schaltet zwischen Ausführung von Prozessen um (***context switch***). Registerinhalte und PCB werden gerettet, der PCB des nächsten Prozesses wird geladen.
- Basis der heutzutage standardmäßig angewendeten Methode ist das präemptive Multitasking und Prozesspriorität.
- Ein oft benutztes Verfahren ist Round-Robin Scheduling (geeignet für Dialogverarbeitung).

Interrupts (Unterbrechungen)

- **Interrupt** (ab 1958): eine kurzfristige Unterbrechung der normalen Programmausführung, um einen, in der Regel kurzen, aber zeitlich kritischen, Vorgang abzuarbeiten. Im Gegensatz zum Polling.
- Das auslösende Ereignis wird **Interrupt Request (IRQ)** genannt. Der Prozessor führt eine **Interrupt Service Routine (ISR)** aus. Die Unterbrechungsroutine wird mit erweiterten Privilegien ausgeführt. Anschliessend wird der vorherige Zustand des Prozessors wiederhergestellt und das unterbrochene Programm wieder fortgeführt.
- Es gibt Hardware-Interrupts durch asynchrone externe Ereignisse ausgelöst und Software-Interrupts, durch den Maschinenbefehl ("INT nn").

Unterbrechungsvorgang
ausgehend von drei möglichen Quellen



Interrupts bei Prozessen

- Beispiele für Interrupts sind:
 - Tastatur, wenn eine Taste gedrückt wird
 - Netzwerkkarte: wenn Daten empfangen wurden und im Puffer bereitliegen
 - Festplatte: wenn die vorher angeforderten Daten gelesen wurden und abholbereit sind (das Lesen von der Festplatte dauert relativ lange)
 - Grafikkarte: wenn das aktuelle Bild fertig gezeichnet wurde
 - ein anderes Programm läuft, es soll auf eine Ein- oder Ausgabe (etwa von Tastatur, Festplatte, Netzwerk oder Zeitgeber) sofort reagieren.
- Die CPU muss Interrupt-fähig sein, Interrupt-Controller kann den entsprechenden Interrupt verarbeiten.
- Interrupts bei Prozessen:
 - Der gerade laufende Prozess wird nach einer bestimmten Abarbeitungszeit (time slice, wenige Millisekunden) unterbrochen durch einen Hardware-Timer, der einen Interrupt erzeugt.
 - Die Interrupt Service Routine (Teil des Betriebssystems) unterbricht den Prozess und sichert seinen Prozesskontext – der Prozess wird „schlafen gelegt“; dann übergibt sie an den Scheduler – das Betriebssystem hat (wieder) die Kontrolle erlangt.
 - Sofern kein Betriebssystem-eigener Ablauf ansteht, wählt der Scheduler nun einen rechenbereiten Prozess aus, stellt dessen Prozesskontext wieder her, startet den Hardware-Timer und übergibt dann an den Prozess.
- Beispiel: Windows 10 Interrupt Tabelle auf Schöners Laptop:
 - msinfo32.exe → Hardwareressourcen → IRQs