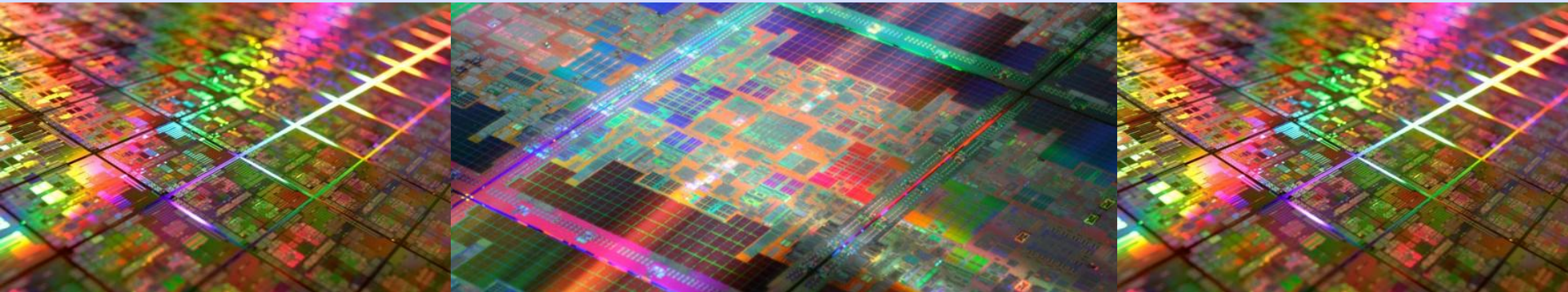


Betriebssysteme

Technische Grundlagen der Informatik für
Wirtschaftsinformatik

Stefan Podlipnig

TU Wien



Lernziele

- Kennenlernen der grundlegenden Terminologie
- Verständnis entwickeln für die generellen Prinzipien im Bereich Betriebssysteme

Was wurde in dieser LV bisher besprochen?

- Allgemeine Konzepte (z.B. von Neumann)
- Prozessor (anhand der MIPS-Architektur)
 - Befehlssatz
 - Rechnerarithmetik
 - Datenpfad und Pipelining
- Speicherhierarchie
- Elementare Prinzipien der Ein-/Ausgabe
- Grundlagen der Rechnernetze

**Wie bringt man jetzt alle Komponenten eines Rechners zum Laufen?
Betriebssystem!**

- Definition nach DIN 44300
- *Unter Betriebssystem versteht man alle Programme eines digitalen Rechensystems, die zusammen mit den Eigenschaften der Rechenanlage die Basis der möglichen Betriebsarten des digitalen Rechensystems bilden und insbesondere die Abwicklung von (Anwendungs-)Programmen steuern und überwachen.*
- Ohne ein Betriebssystem funktioniert die Hardware nicht

Betriebssystem - Start

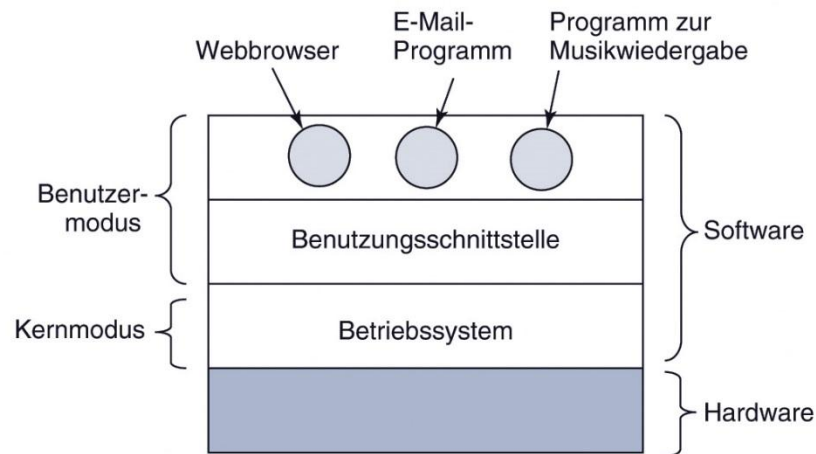
- Betriebssystem ist aber auch nur ein Programm
- Wer startet das Betriebssystem?
- Beim Start des Rechners beginnt der Prozessor an einer bestimmten Adresse mit der Ausführung
 - Dort steht Programmcode für das Ausführen eines Startprogramms, das dann auch das eigentliche Betriebssystem startet
 - BIOS, UEFI
- Erst danach kann der Rechner benutzt werden
- Bei speziellen Rechnern kann das Betriebssystem im ROM stehen

Aufgaben des Betriebssystems

- Schnittstelle zwischen Mensch und Hardware
 - Bedienschnittstelle (*User Interface*)
 - Konsole
 - Grafische Benutzeroberfläche
 - Programmierschnittstelle (API, *Application Programming Interface*)
 - Versteckt die Komplexität der Maschine hinter einfachen Befehlen (Systemaufrufe)
- Verwalter von Ressourcen
 - Auf einem Rechner laufen mehrere Programme gleichzeitig ab
 - Betriebssystem garantiert eine geordnete, kontrollierte und schnelle Zuteilung von Prozessoren, Speichereinheiten und Peripheriegeräten unter den verschiedenen Programmen

Operationsmodi

- Kernmodus (*kernel mode*)
 - Betriebssystem läuft im Kernmodus
 - Vollständiger Zugang zur gesamten Hardware
 - Jeder Befehl der Maschine kann ausgeführt werden
- Benutzermodus (*user mode*)
 - Rest der Software läuft im Benutzermodus
 - Nur eine Teilmenge der Befehle steht zur Verfügung
 - Keine Befehle für die Kontrolle über die Maschine und die Ein- und Ausgabe stehen zur Verfügung



Dienste eines Betriebssystems

- Prozessmanagement
- Speichermanagement
- Dateiverwaltung
- Steuerung und Abstraktion der Hardware, Geräteverwaltung, E/A-Steuerung
- Bereitstellen der Benutzeroberfläche

Schichten

- Betriebssysteme besitzen normalerweise einen Schichtenaufbau
- Höhere Schichten benutzen niedrigere Schichten (aber nicht umgekehrt)
- Erleichtert die Übersichtlichkeit, Wartung und Anpassung

Programmierschnittstelle (API)		Bedienschnittstelle (UI / GUI)	
Fehlerbehandlung und -verwaltung			
Speicher- und Dateiverwaltung			
I/O-Verwaltung, Netzwerk-Dienste			
Kernel-Dienste	Zeitdienste, Interrupt-Verwaltung		
	Task-Synchronisation und -Kommunikation		
	Task-Verwaltung, Scheduling		
HW-Support / BSP (Interrupts, I/O, Initialisierung, ...)			

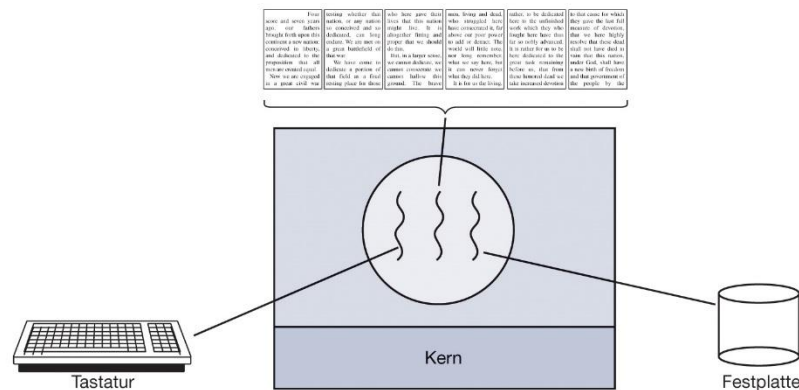
- Anwendungsprogramme werden in Form von Prozessen verwaltet und ausgeführt
 - Prozess ist ein Programm in Ausführung
- Jeder Prozess besitzt einen Adressraum
 - Beinhaltet das ausführbare Programm, die Programmdateien und den Stack (Keller)
 - Virtueller Speicher
- Daneben besitzt ein Prozess eine Menge an Informationen für die Ausführung
 - Register (z.B. aktueller Befehlszähler)
 - Liste geöffneter Dateien
 - Liste verbundener Prozesse
 - ...
- Informationen werden in einer Prozesstabelle abgelegt

Prozesstabelle (typische Einträge)

Prozessverwaltung	Speicherverwaltung	Dateiverwaltung
Register	Zeiger auf Textsegment	Wurzelverzeichnis
Befehlszähler	Zeiger auf Datensegment	Arbeitsverzeichnis
Programmstatuswort	Zeiger auf Stacksegment	Dateideskriptor
Kellerregister		Benutzer-ID
Prozesszustand		Gruppen-ID
Priorität		
Scheduling-Parameter		
Prozess-ID		
Elternprozess		
Prozessgruppe		
Signale		
Startzeit des Prozesses		
Benutzte CPU-Zeit		
CPU-Zeit der Kindprozesse		
Zeitpunkt des nächsten Alarms		

Threads

- Threads
 - Quasi-parallele Abläufe innerhalb eines Prozesses
 - Teilen sich alle Ressourcen eines Prozesses
- Vor- und Nachteile
 - Weniger Aufwand bei Wechsel zwischen Threads
 - Gemeinsamer Speicher
 - Schnelle Kommunikation
 - Synchronisation erforderlich
 - Kein Speicherschutz
- Beispiel: Textverarbeitung (ein Prozess) mit drei Threads



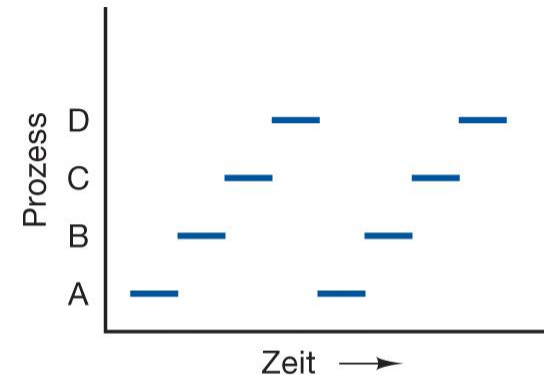
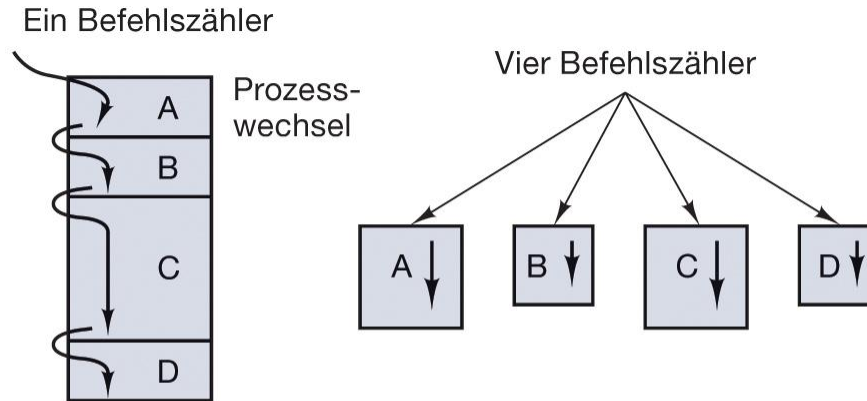
- Beispiele für Elemente pro Prozess
 - Adressraum
 - Globale Variablen
 - Geöffnete Dateien
 - Verwaltungsinformationen
- Beispiele für Elemente pro Thread
 - Befehlszähler (PC)
 - Register
 - Stack
 - Zustand

Scheduling

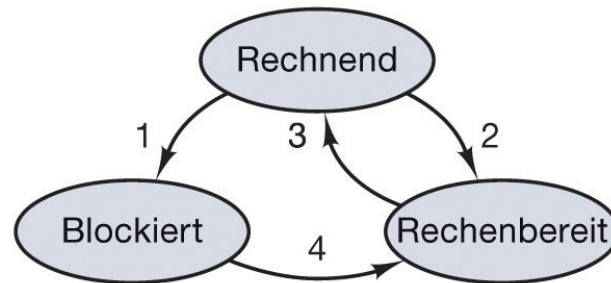
- Heutige Betriebssysteme unterstützen Multitasking
 - Mehrere Prozesse können abwechselnd bearbeitet werden
 - Jeder Prozess bekommt ein Zeitquantum (z.B. 10 ms)
- Scheduler
 - Wählt den nächsten zu bearbeitenden Prozess (oder Thread)
 - Schaltet zwischen Prozessen um
 - Läuft zu bestimmten Zeitpunkten oder bei bestimmten Ereignissen
 - Unterbricht den gerade laufenden Prozess
 - Startet einen anderen Prozess
- Umschalten (Prozess-Wechsel)
 - Informationen zum Prozess (Prozessor-Kontext) speichern
 - Angehaltener Prozess besteht aus
 - Adressraum (Speicherabbild)
 - Prozesstabelleneintrag mit Informationen (Register etc.)

Multitasking

- Mehrere Prozesse (Ablauf)



- Prozesszustände

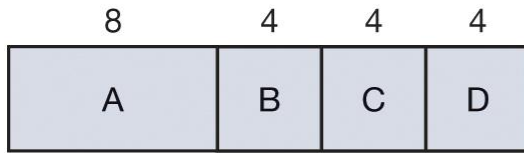


1. Prozess blockiert, weil er auf Eingabe wartet
2. Scheduler wählt einen anderen Prozess aus
3. Scheduler wählt diesen Prozess aus
4. Eingabe vorhanden

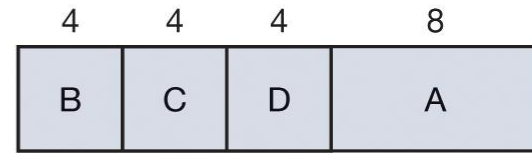
Scheduling - Beispiele

- Shortest-Job-First-Scheduling

- a) Originalfolge, b) Shortest-Job-First-Scheduling



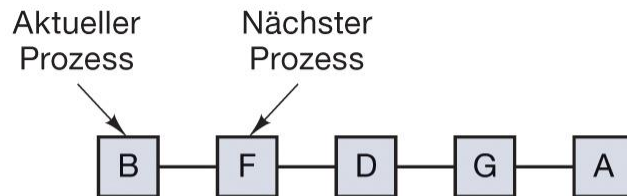
a



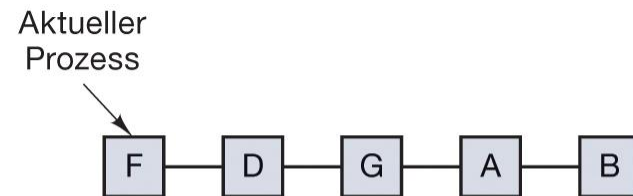
b

- Round-Robin-Scheduling

- a) Liste der Prozesse in Ausführung, b) Liste nachdem B sein Quantum aufgebraucht hat



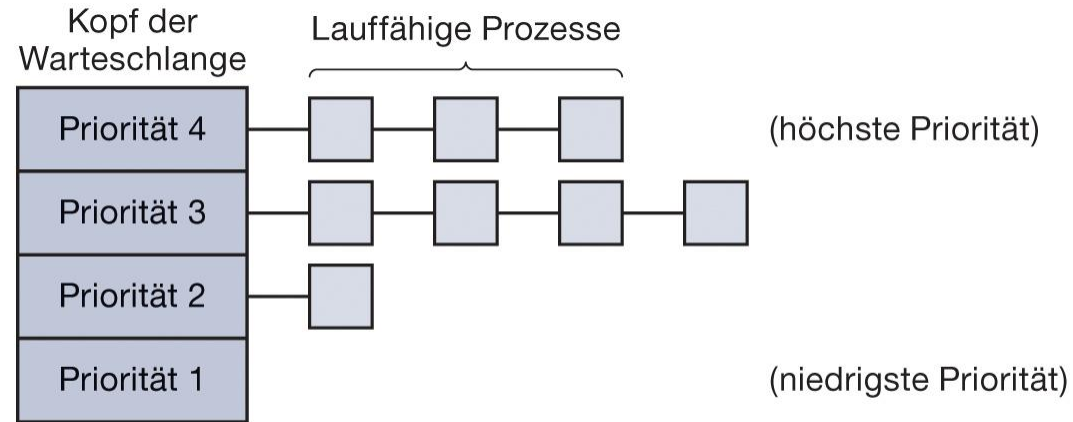
a



b

Scheduling - Prioritäten

- Vorrang mit Prioritäten realisieren



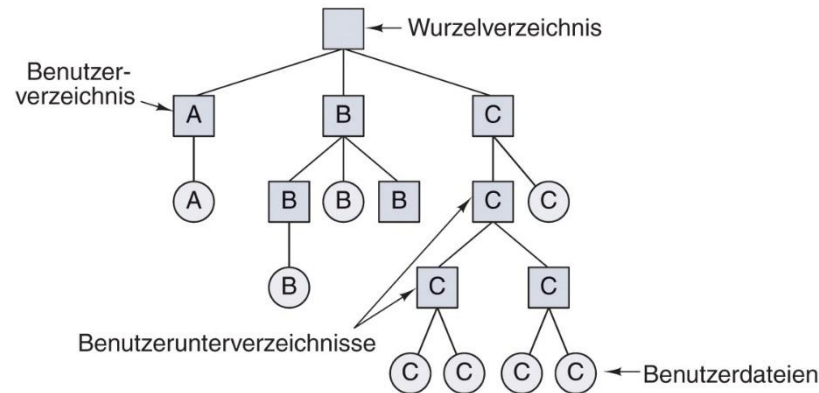
- Betriebssystem muss Programmen Speicher zuteilen und verwalten
- Speicher für Programme
 - Fixer Speicher (z.B. globale Variablen, Konstanten etc.) für die gesamte Laufzeit
 - Stack-Speicher für unterschiedliche Daten (z.B. für Funktionsaufrufe) während der Laufzeit eines Programms
 - Heap-Speicher für dynamische Belegung
- Virtueller Speicher
 - Betriebssystem und die Hardware verwalten den virtuellen Speicher
 - Betriebssystem wird hierbei hardwareseitig von der MMU (*Memory Management Unit*) unterstützt
 - Umrechnung von virtuellen in physikalische Adressen
 - Speicherschutz

Dateiverwaltung (1)

- Datei
 - Sammlung von Daten auf einem Permanentpeicher
- Dateisystem
 - Macht Daten auf einem Speichermedium zugänglich
 - Benutzer muss sich nicht um die Details der internen Datenorganisation kümmern
 - Bietet Schutzmechanismus an
 - Nur berechtigte Benutzer dürfen bestimmte Dateien lesen
 - Verwaltet die zusätzlichen Daten für jede Datei (Rechte etc.)
- Organisation der Daten
 - Daten als lineare Folge von Bytes in Dateiblöcken (z.B. 1KiB Blöcke)
 - Datei ist eine Menge von Blöcken
 - Diese Blöcke können z.B. auf einer Festplatte verteilt sein
 - Das Betriebssystem verwaltet die Abbildung der Daten der Dateien auf die einzelnen Blöcke

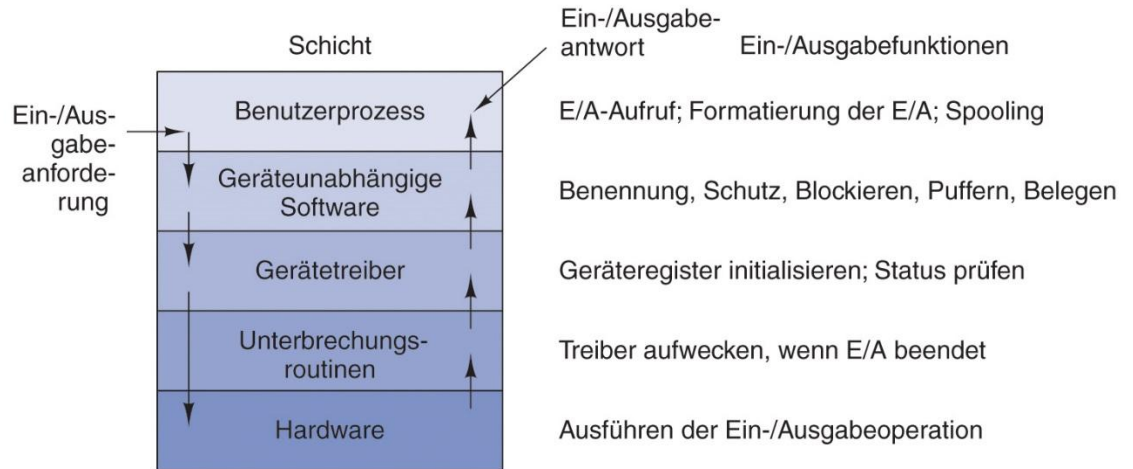
Dateiverwaltung (2)

- Dateiarten (Beispiele)
 - Normale Dateien (Textdateien, Bilddateien, ausführbare Programme etc.)
 - Verzeichnisse (bei manchen Betriebssystemen)
 - Verweise (Links) auf andere Dateien
- Organisation von Dateien
 - Üblicherweise mit einem Verzeichnisbaum
 - Verzeichnisse speichern Metainformationen (Dateinamen, Erstellungsdatum, Zugriffsrechte etc.)



Geräteverwaltung und Treiber

- Betriebssystem verwaltet E/A-Geräte
- Betriebssystem bietet eine einfache Schnittstelle für die Ein-/Ausgabe (*Input/Output, I/O*) an
- Treiberschnittstelle
 - Um die unterschiedlichen neuen E/A-Geräte zu unterstützen
 - Steuerprogramme für E/A-Geräte heißen Treiber (Treiberprogramme)
 - Treiber werden vom Hersteller des E/A-Gerätes angeboten und werden oft extra installiert
 - Zugriff auf Geräte

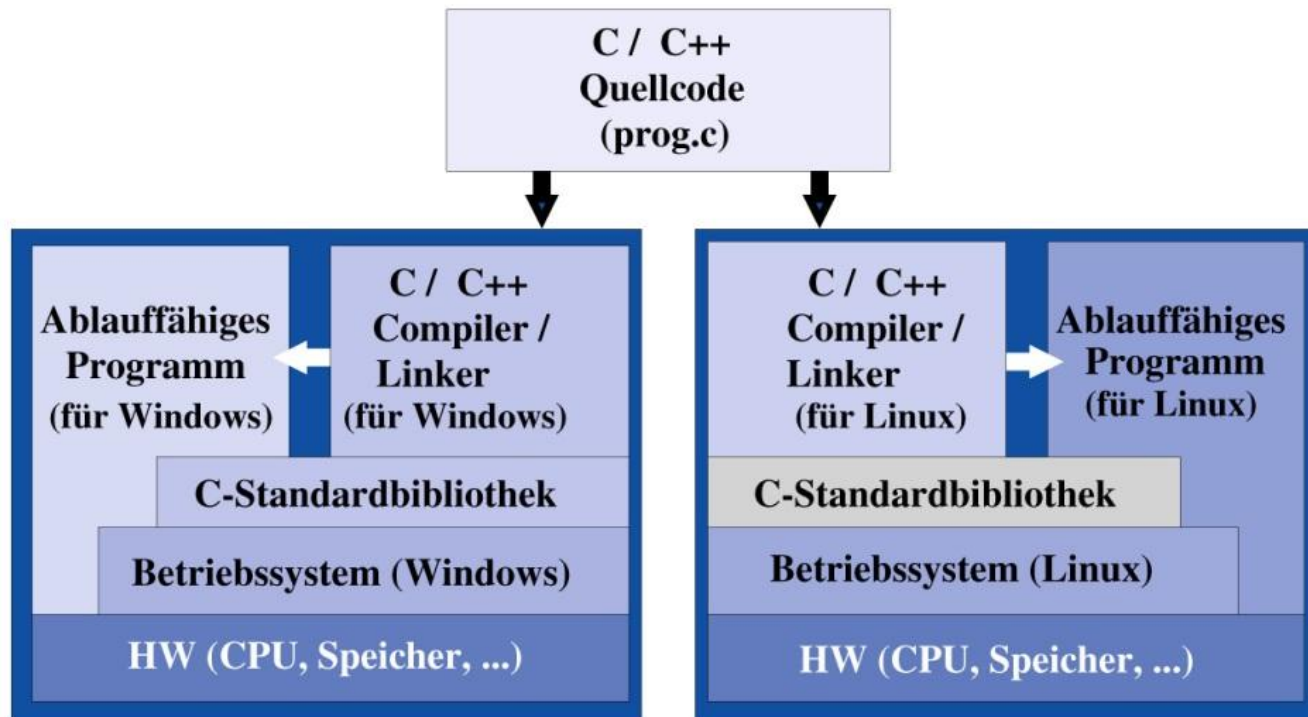


Benutzerschnittstelle (GUI)

- Grafische Benutzeroberfläche
 - Gängigste Methode
 - Einfachere Bedienung
 - Ähnliche Abläufe bzw. Interaktionsmöglichkeiten auf verschiedenen Betriebssystemen
- GUI-Einbindung benötigt
 - Einbindung von Multitasking
 - Leistungsfähiges Speichermanagement
 - Einheitliche Grafikschnittstelle
 - Eingabeverarbeitung (Ereignisse, asynchron)

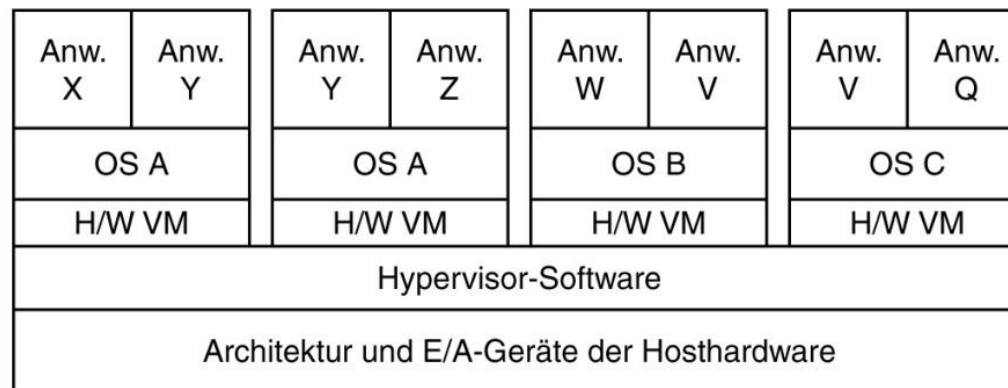
Programmierschnittstelle

- Komplexität hinter Befehlen versteckt (Systemaufrufe)
- Beispiel C-Programme



Virtualisierung

- Virtualisierung
 - Ermöglicht mehrere virtuelle Systeme mit eigenem Betriebssystem auf einem physischen Rechner zu betreiben
 - Aktuelle Prozessoren unterstützen diese Virtualisierung mit Hardware
- Hypervisor
 - Abstrahierende Schicht zwischen tatsächlich vorhandener Hardware und Betriebssystemen
 - Verwaltet Instanzen von virtuellen Rechnern



- **Grundliteratur**

- H. Herold, B. Lurz, J. Wohlrab, **Grundlagen der Informatik**, 2. Auflage, Pearson, 2012 – Kapitel 9
- A. S. Tanenbaum, **Moderne Betriebssysteme**, 3. Auflage, Pearson Studium, 2009 – Kapitel 1 – 4
- A. S. Tanenbaum, T. Austin, **Rechnerarchitektur**, 6. Auflage, Pearson Studium, 2014 – Kapitel 6