



**Hochschule für Technik
und Wirtschaft Berlin**

University of Applied Sciences

**Innovativ und vielfältig: die Hochschule
für Technik und Wirtschaft Berlin**

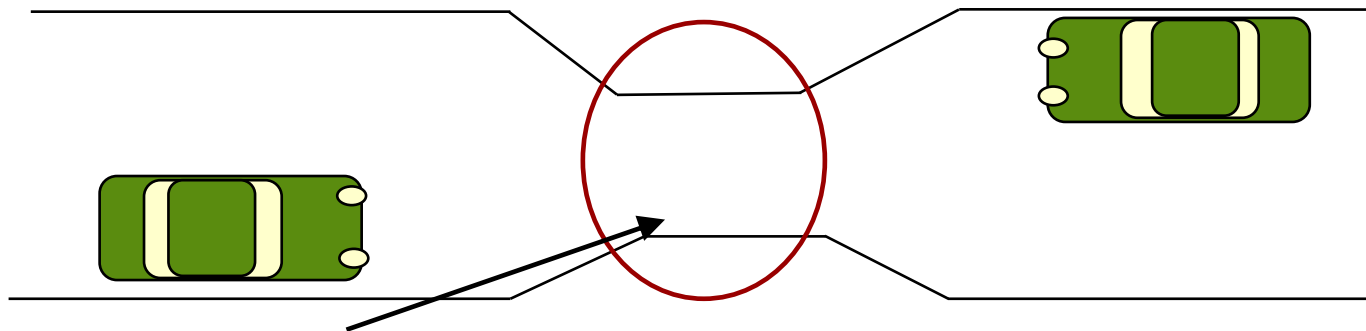
Fachbereich 2
Informatik
Vorkurs **Informatik**

Lektion 5

BETRIEBSSYSTEME

Betriebsmittelvergabe

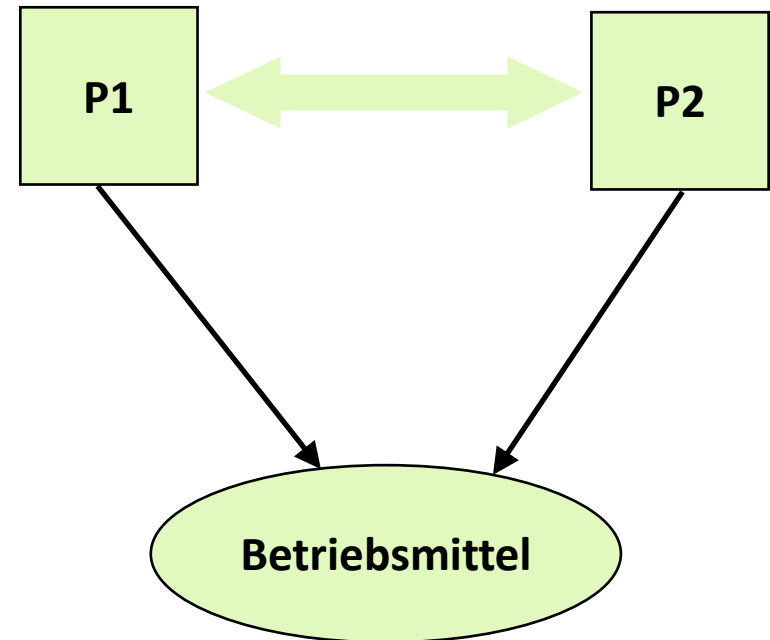
- Betriebsmittelbelegung
 - Betriebsmittel werden vom BS entzogen und zugeteilt.
 - Problem: Zwei Prozesse wollen auf das gleiche Betriebsmittel zugreifen und dieses ist nur exklusiv nutzbar
- Betriebsmittelproblem des Alltags



- **Kritischer Abschnitt** wird der Bereich genannt in dem sich höchstens ein Prozess befinden darf.
- Lösung: Verständigung der Teilnehmer, wechselseitiger Ausschluss (Ampel)

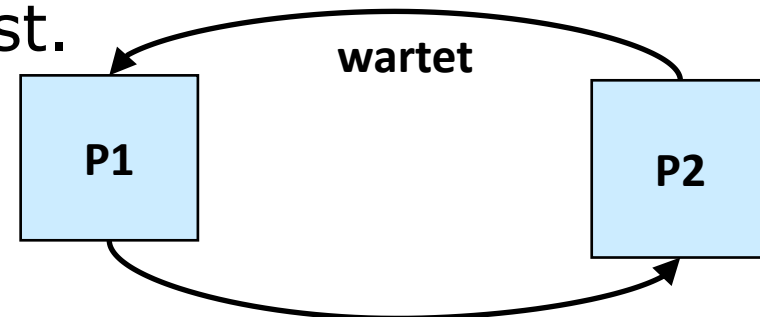
Betriebsmittelverwaltung

- Kommunikation zwischen den beteiligten Prozessen P1 und P2 über Protokolle
- Kollisionsvermeidung durch Nutzung von Sperren zum gegenseitigen Ausschluss beim Passieren des kritischen Abschnitts.



Wann treten Verklemmungen auf?

- **Wechselseitiger Ausschluss** – das Betriebsmittel kann zu einem Zeitpunkt nur von einem Prozess genutzt werden.
- **Hold-and-Wait** – Ein Prozess, der im Besitz von Betriebsmitteln ist, kann weitere anfordern.
- **Ununterbrechbarkeit** – Das Betriebsmittel kann dem Prozess nicht entzogen werden.
- **Zyklisches Warten** – Es gibt eine zyklische Kette von Prozessen, bei der jeder auf ein Betriebsmittel wartet, das von anderen Prozessen in der Kette belegt ist.

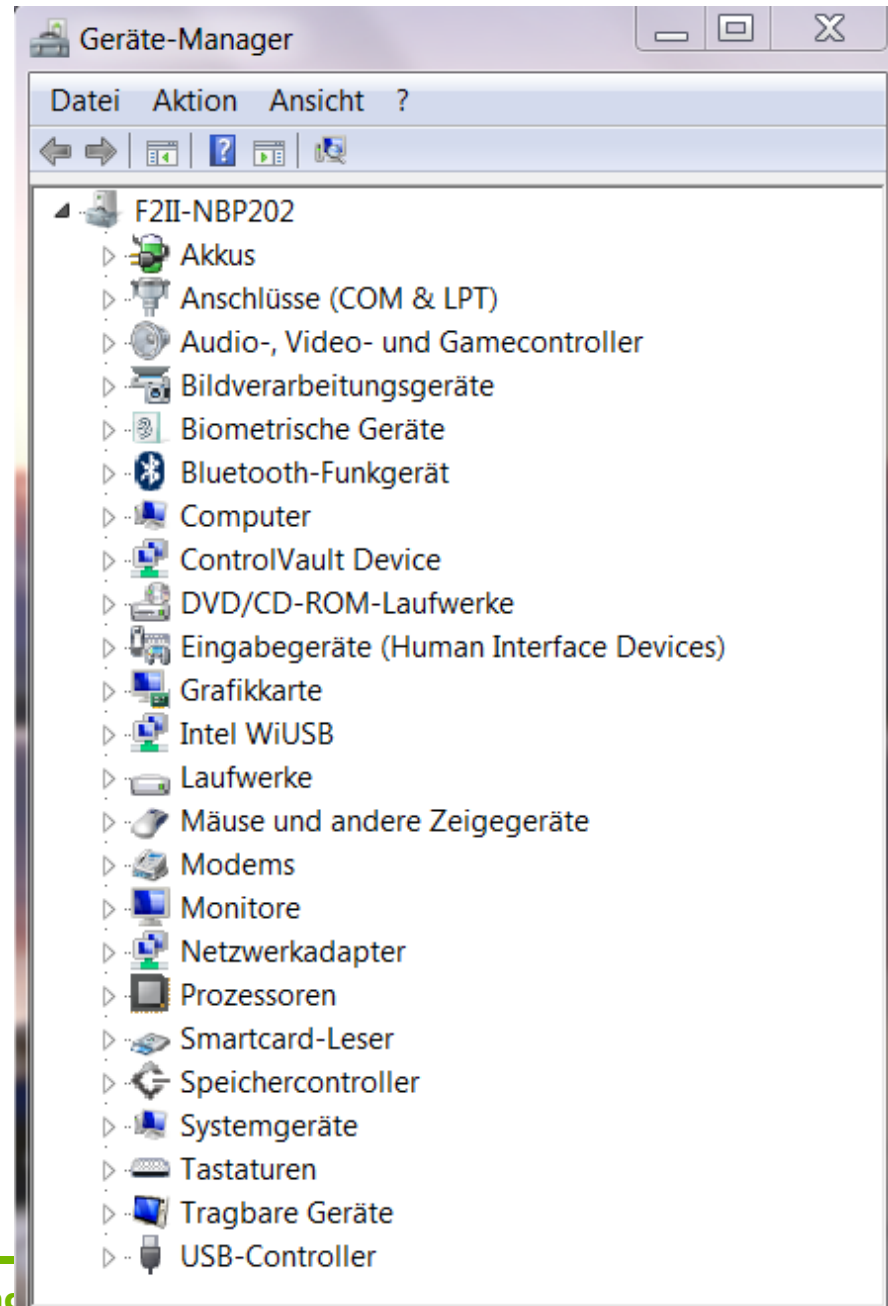


Geräteverwaltung

- Als Gerät wird auf BS-Ebene alles bezeichnet mit dem über Ein/Ausgabe - Kanäle kommuniziert wird.
- Geräte sind Betriebsmittel
- Geräte werden über Gerätetreiber angesprochen und gesteuert
- Bereitstellung einheitlicher Schnittstellen für Gerätetreiber
- Abstraktion von Hardwaredetails
- Geräte unter Linux sind über Namen im Dateisystem ansprechbar



Was sind Geräte?



Aufgaben der Geräte-Verwaltung

- Initiierung von E/A- Vorgängen (Gerätetreiber)
 - Definition der Geräteeigenschaften gegenüber dem BS
 - Initialisierung des Gerätes beim Systemstart
 - Aktivierung des Gerätes
 - Meldung von Gerätefehlern
 - Reaktion auf Interrupts des Gerätes
 - Umsetzung der E/A-Aufträge in spezifische Geräte-Informationen
 - Übergabe von Daten und Status-Informationen an den Nutzer
- Überwachung von E/A-Vorgängen
- Terminierung von E/A-Vorgängen
- Effiziente Zuteilung der E/A-Geräte an Prozesse
- Vermeidung von Zugriffskonflikten

KOMMUNIKATION PROZESSE

Programme und Prozesse

Prozessdefinition:

Wird ein Programm aufgerufen (Benutzerprogramm, Kommando), wird der Programmcode in den HS geladen und abgearbeitet.

Ein Prozess (engl. Task) ist ein in Ausführung befindliches Programm

In einem Multitasking-BS entscheidet das BS periodisch, einen laufenden Prozess zu unterbrechen und einen anderen Prozess zu starten (wenn die dem Prozess zustehende CPU-Zeit verbraucht ist).

-> **Prozessscheduling**

Wenn ein Prozess suspendiert wurde, muss er später in genau diesem Zustand wieder gestartet werden.

Prozesse und Scheduling

Instruktionen werden sequentiell ausgeführt

- Sprünge verändern nur die Reihenfolge

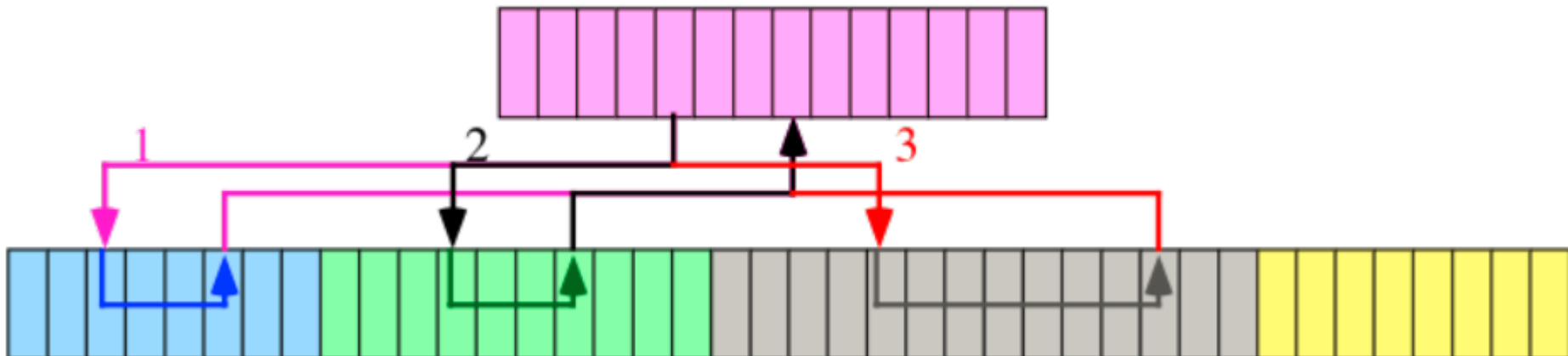


Programm = ausführbares Objekt = Prozeß

- mehrere Prozesse liegen im Speicher



Benutzerprozesse und ein besonderer Steuerprozeß

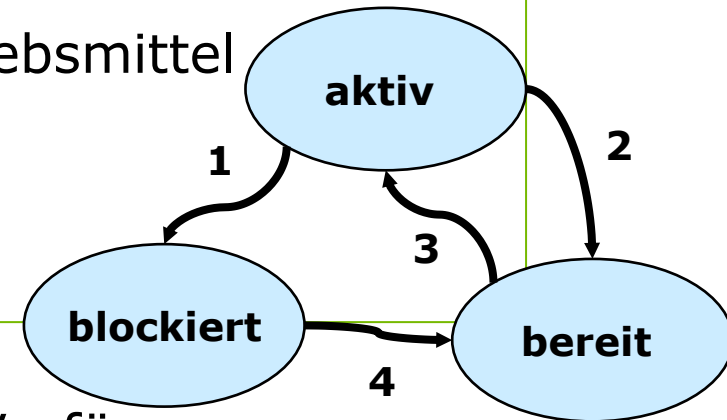


Prozess- und Prozessorverwaltung

- Ein Betriebssystem erlaubt „*Multitasking*“, d.h. mehrere Prozesse befinden sich gleichzeitig im System und werden von einem Prozessor (CPU) bedient. Ein Prozess besteht aus:
 - dem Speicherabbild (core image), d.h. den Programmanweisungen und Daten, die im Hauptspeicher liegen,
 - der Registerbelegung und
 - Verwaltungsinformationen, z.B. der bisher verbrauchten CPU-Zeit
- Das BS hat die Aufgabe alle n Prozesse auf einen oder mehr Prozessoren abzubilden
- Bei Verwaltung mehrerer Prozesse entscheidet der **Prozess-Scheduler** des Betriebssystems über die Ausführungsreihenfolge und die Zuteilung des Prozessors
- Jeder Prozess besitzt eine Liste mit Betriebsmittelanforderungen sowie eine Liste der bereits zugeteilten Betriebsmittel

Prozess- und Prozessorverwaltung

- **Zustände** eines Prozesses
 - **bereit** (ready): wartet auf die Zuteilung des Prozessors
 - **aktiv** (running): besitzt den Prozessor und läuft ab
 - **blockiert** (blocked, asleep): Prozessor wurde entzogen, wartet auf Betriebsmittel
- Vier **Übergänge** sind zwischen diesen Zuständen möglich:



1. Prozess blockiert, notwendiges Betriebsmittel steht nicht zur Verfügung
2. Scheduler entzieht den Prozessor
3. Scheduler teilt Prozessor zu
4. Benötigtes Betriebsmittel steht zur Verfügung

MS Windows Task Manager

- Informationen über
- die aktuell ausgeführten Programme
 - die z.Z. laufenden Prozesse
 - verschiedene Leistungsmerkmale des Systems (z.B. Speicher- und Prozessorstatus)

Abbildname	Benu...	CPU	Arbeitsspei...	Beschreibung
firefox.exe *32	abu...	00	384.700 K	Firefox
OUTLOOK.EXE *32	abu...	00	74.468 K	Microsoft Outlook
Acrobat.exe *32	abu...	00	71.744 K	Adobe Acrobat
explorer.exe	abu...	00	59.004 K	Windows-Explorer
POWERPNT.EXE *32	abu...	00	54.944 K	Microsoft PowerPoint
iTunes.exe *32	abu...	00	52.484 K	iTunes
dwm.exe	abu...	01	45.144 K	Desktopfenster-Manager
PrivacyIconClient.exe	abu...	00	11.836 K	Intel(R) Management and Security Status
BTStackServer.exe	abu...	00	11.556 K	Bluetooth Stack COM Server
AcroRd32.exe *32	abu...	00	9.832 K	Adobe Reader
IAStorIcon.exe *32	abu...	00	9.272 K	IAStorIcon
nvxdsync.exe		00	8.932 K	
iFrmewrk.exe	abu...	00	8.108 K	Intel(R) PROSet/Wireless Framework
DbrmTrayicon.exe	abu...	00	7.948 K	DBRM_Toaster
AppleMobileDeviceHelper.ex...	abu...	00	7.604 K	YSLoader.exe
AcroRd32.exe *32	abu...	00	6.984 K	Adobe Reader
YsLoader.exe	abu...	00	6.580 K	YSLoader.exe

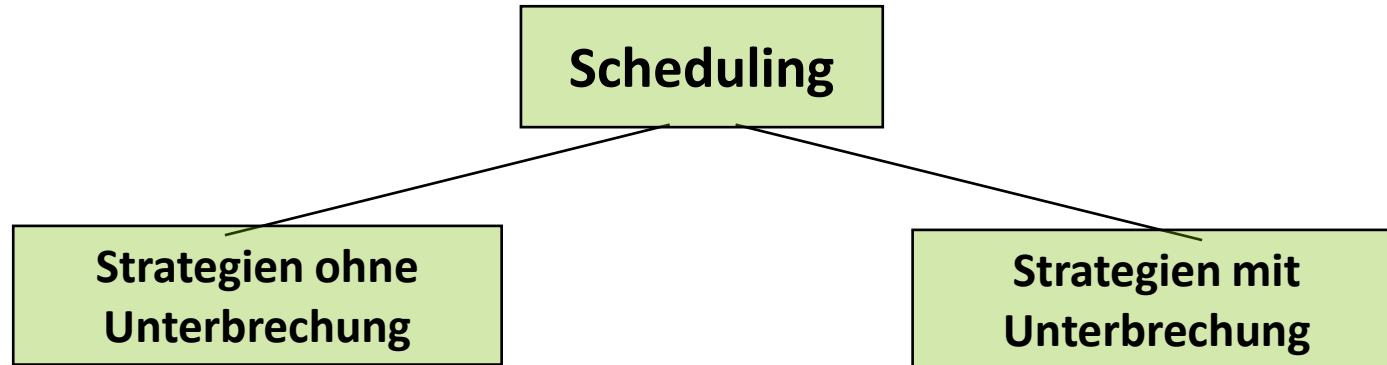
Prozesse aller Benutzer anzeigen Prozess beenden

Prozesse: 133 CPU-Auslastung: 1% Physikalischer Speicher: 22%

Prozess vs. Programm

- Programm ist Handlungsvorschrift
- Prozess ist die Handlung selbst
- In jedem Prozess läuft ein Programm
- Nicht jedes Programm läuft gerade (manche liegen nur auf der Platte und sind inaktiv)

Prozess- und Prozessorverwaltung



Dem laufenden Prozess kann der Prozessor nicht entzogen werden.

Vorteil: Geringer Aufwand

Nachteil: Für den Dialogbetrieb und Echtzeitsysteme nicht geeignet.

Dem laufenden Prozess kann der Prozessor entzogen werden. Warten mehrere Prozesse auf den Prozessor, wird die Prozessorzeit auf diese aufgeteilt (Zeitscheibenverfahren).

Vorteil: gerechte Betriebsmittelzuteilung

Nachteil: höherer Aufwand (Overhead)

Strategien ohne Unterbrechung

- **First Come First Serve (FCFS/FIFO):**
 - Planung nach Ankunftszeit
 - Prozesse werden in der Reihenfolge ihres Eintreffens abgearbeitet
 - Antwortzeit hängt von der Prozessorauslastung ab, d.h. von der Anzahl der Prozesse
- **Shortest Job Next (SJN):**
 - Planung nach Laufzeit
 - Prozesse mit (geschätzt) kurzer Laufzeit werden bevorzugt
 - Wartezeit für kurze Prozesse sinkt
 - Antwortzeit hängt von der Länge des Prozesses und der Prozessorauslastung ab

Strategien mit Unterbrechung

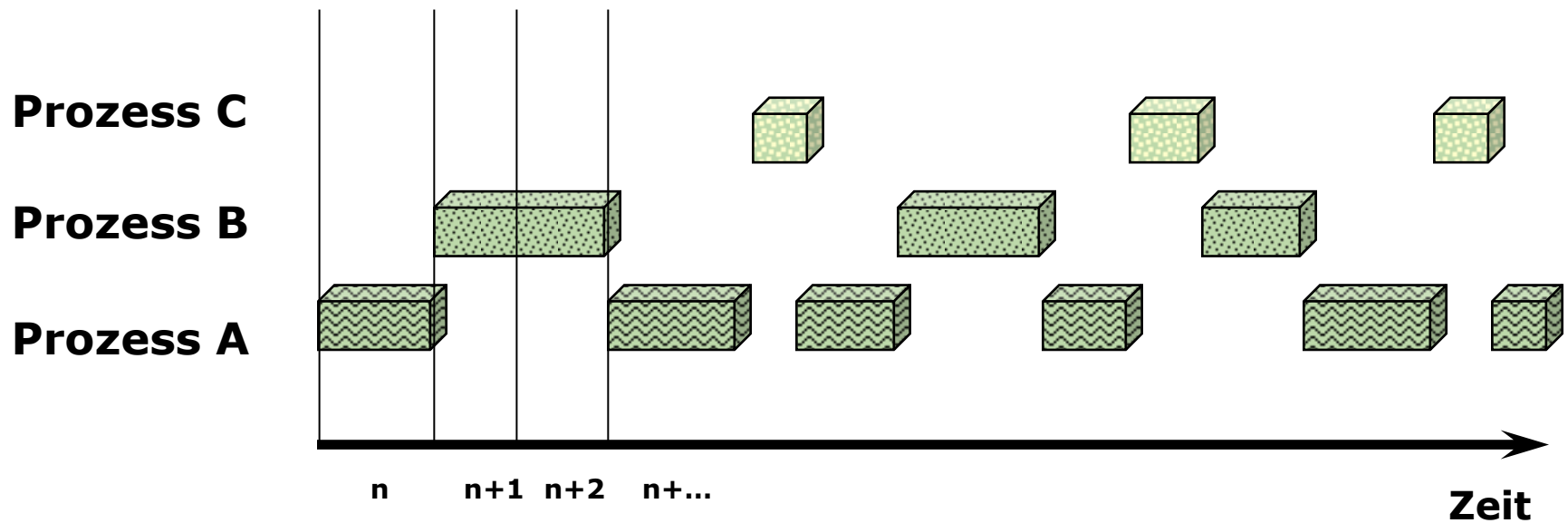
- **Round Robin**

- Weitverbreitete Strategie
- Einfache Implementierung
- Jeder Prozess erhält den Prozessor für eine festgelegte Zeitspanne und wird danach wieder in eine Warteliste einsortiert
- Warteschlange kann nach FCFS sortiert sein
- Problem: sinnvolle Zeitspanne wählen

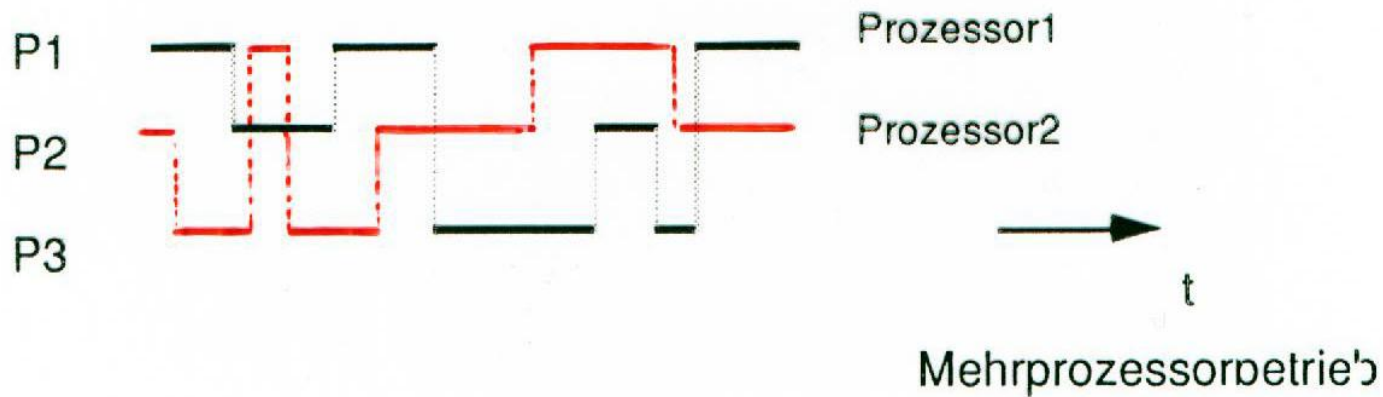
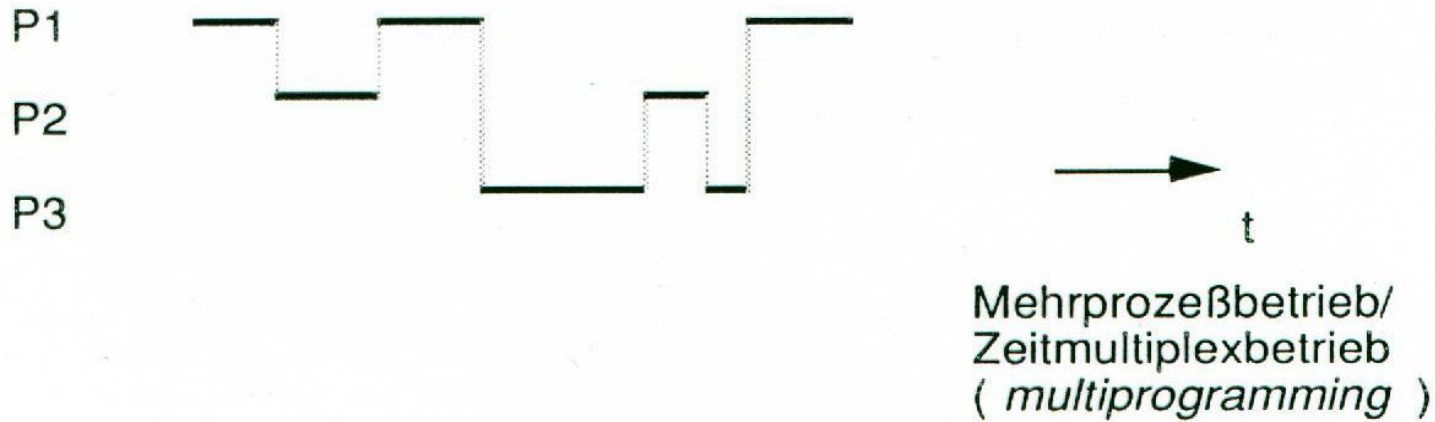
- **Planung nach Prioritäten**

- Jeder Prozess erhält eine bestimmte Priorität
- Prozesse mit hoher Priorität werden bevorzugt
- Problem bei Scheduling mit festgesetzten Prioritäten
Prozesse können *ausgehungert* werden
- Lösung: Aging - Priorität steigt mit der Wartezeit

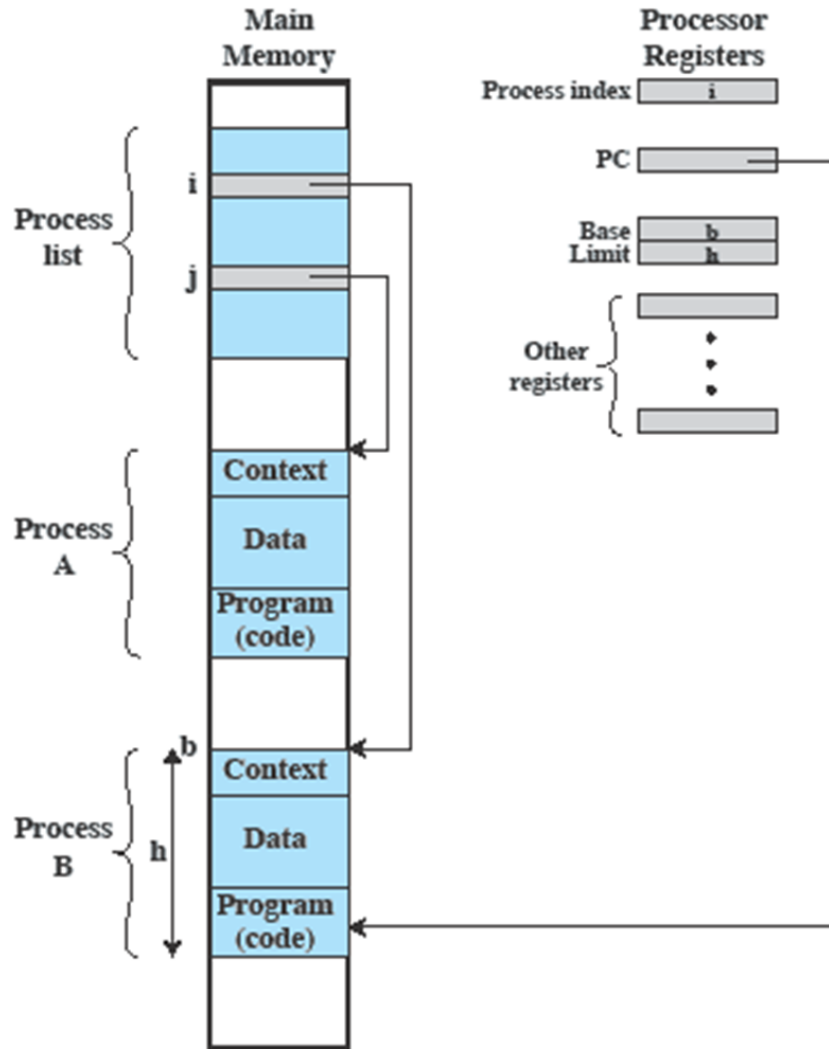
Zeitscheibenverfahren



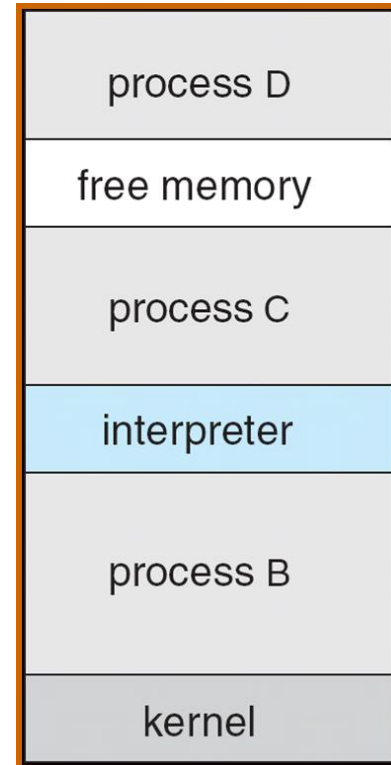
Prozessumschaltung



Memory Management for Processes

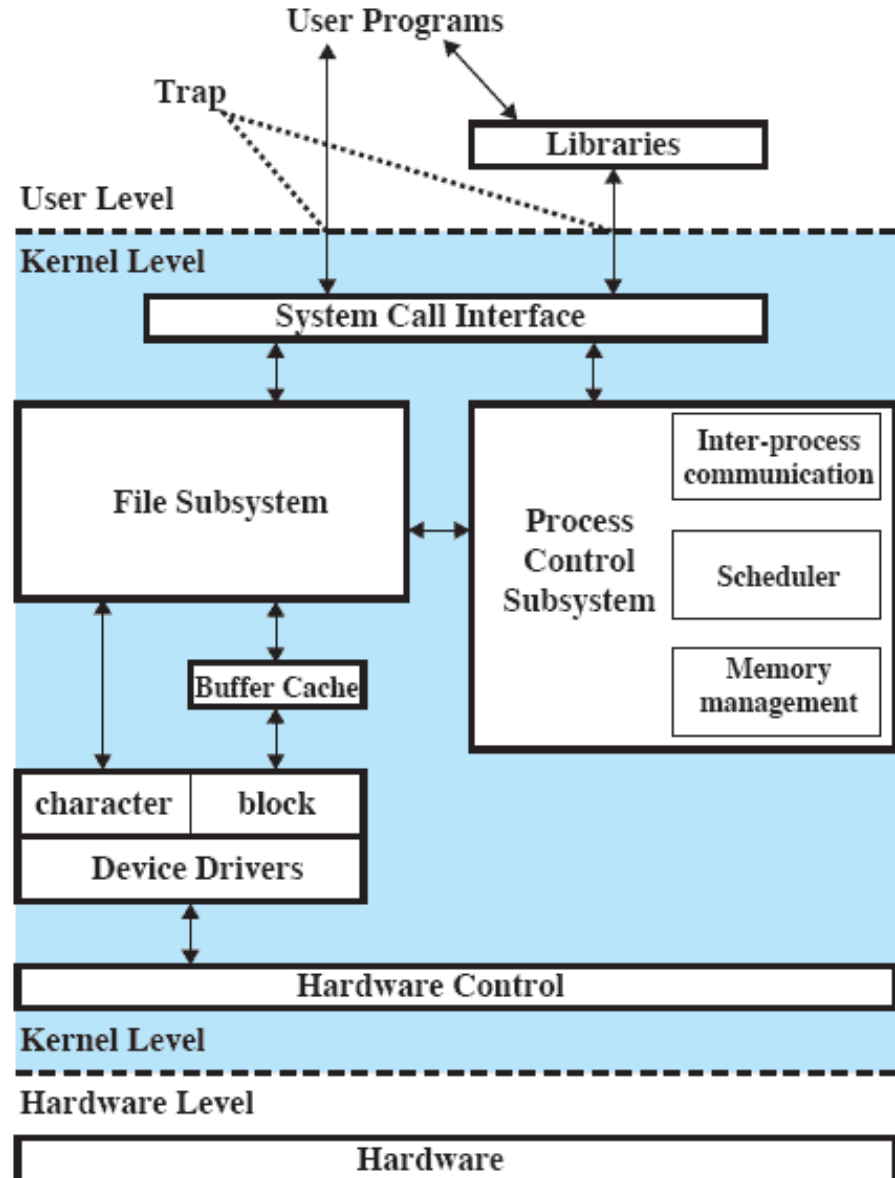


- Process isolation
- Automatic allocation and management
- Support of modular programming
- Protection and access control
- Long-term storage



UNIX kernel

- Hardware
- Abstraction
- Layering



UNIX	Win32	Description
fork	CreateProcess	Create a new process
waitpid	WaitForSingleObject	Can wait for a process to exit
execve	(none)	CreateProcess = fork + execve
exit	ExitProcess	Terminate execution
open	CreateFile	Create a file or open an existing file
close	CloseHandle	Close a file
read	ReadFile	Read data from a file
write	WriteFile	Write data to a file
lseek	SetFilePointer	Move the file pointer
stat	GetFileAttributesEx	Get various file attributes
mkdir	CreateDirectory	Create a new directory
rmdir	RemoveDirectory	Remove an empty directory
link	(none)	Win32 does not support links
unlink	DeleteFile	Destroy an existing file
mount	(none)	Win32 does not support mount
umount	(none)	Win32 does not support mount
chdir	SetCurrentDirectory	Change the current working directory
chmod	(none)	Win32 does not support security (although NT does)
kill	(none)	Win32 does not support signals
time	GetLocalTime	Get the current time

Betriebssystemnahe Programmierung

```
1: #include <stdlib.h>
2: #include <stdio.h>

3: int main() {
4:     int *number = malloc( sizeof(int) );
5:     scanf("%d", number);
6:     printf("Die Zahl ist: %d\n", *number);
7:     free(number);
8: }
```

Kommunikation zwischen Prozessen

- temporäre Dateien
- Pipes
- Signale (Unterbrechungssystem)
- Shared Memory (gemeinsam genutzter Speicher)
- Semaphore
- Messages
- Sockets

IPC = Inter
Process
Communication



Kommunikation mittels Shared Memory

- Zeitkritische Abläufe (race conditions)
- Kritische Bereiche
- Wechselseitiger Ausschluss
- Schlafen und Aufwecken
- Semaphore
- Monitore

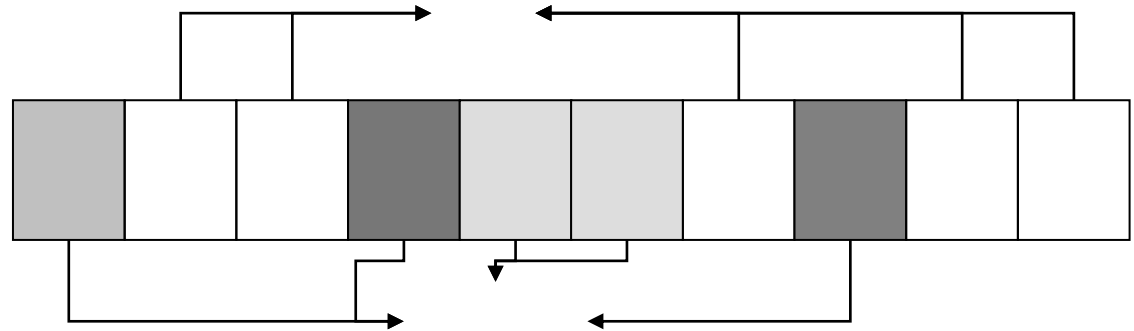
Zeitkritische Abläufe: Probleme bei gemeinsam genutztem Speicher (HS oder Dateien)

Mehrere Prozesse lesen/schreiben gemeinsam genutzte Daten, Ergebnisse hängen von zeitlicher Reihenfolge der Lese-/Schreiboperationen ab.

Beispiel: Drucker-Warteschlange

SPEICHER

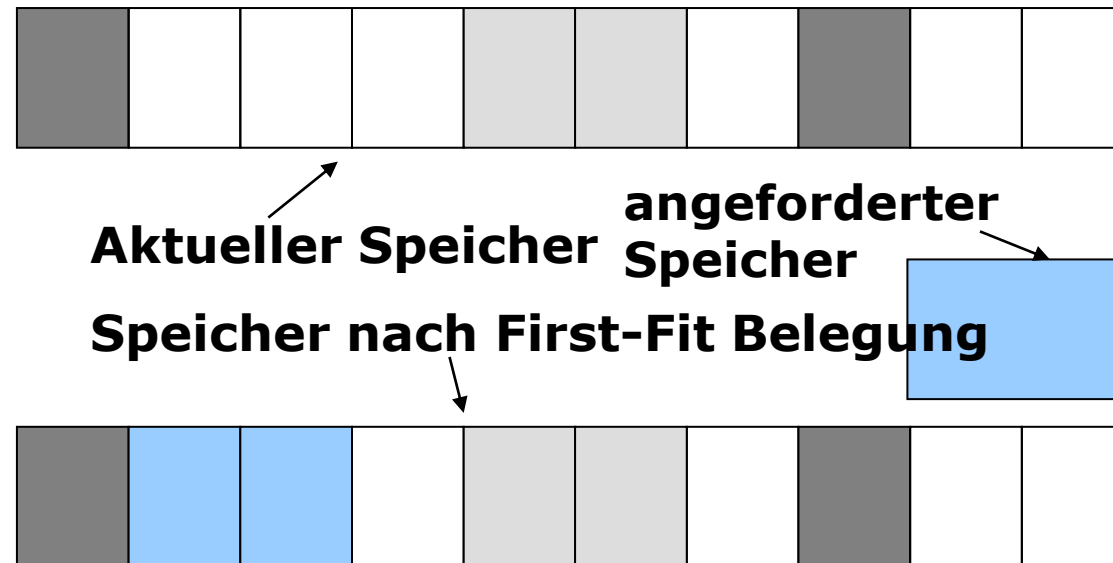
Arbeitsspeicherverwaltung



- Problem: Belegungsverfahren
 - Auswahl der Stücke
 - Effizienz der Algorithmen
 - Speicherausnutzung
- Speicherverwaltungsverfahren unterscheiden sich hinsichtlich
 - der Größe der Stücke
 - des Verschnitts
 - der Auswahlstrategie (bei freien Stücken)

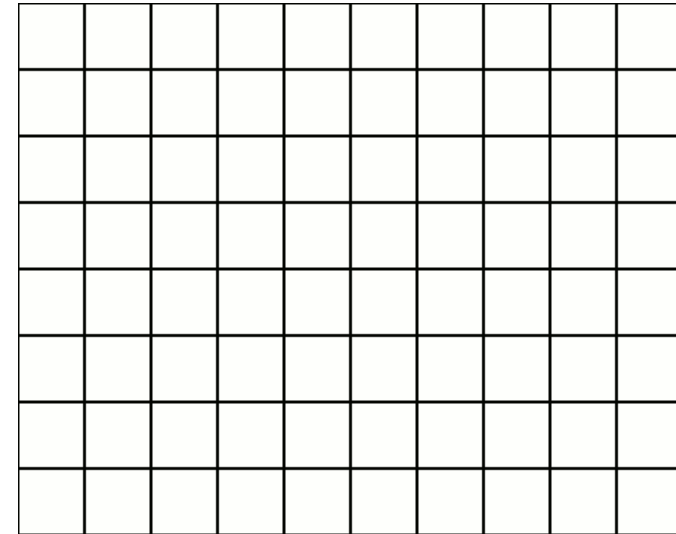
Auswahlstrategien bei der Belegung > First-Fit - Erstes passendes Stück

- Nach Adressen sortierte Liste der freien Stücke wird durchlaufen
- Das erste passend große freie Stück wird genommen
- Großer Verschnitt
- Konzentration der belegten Stücke auf den Speicheranfang



Verschnitt (Fragmentierung)

- **Fragmentierung** (Fragment = Bruchstück): verstreute Speicherung von logisch zusammengehörigen Datenblöcken des Dateisystems auf einem Datenträger
- Durch das wiederholte Belegen und Freigeben des Speichers können Stücke entstehen, die einzeln zu klein sind um einen neuen Prozess/Datei aufzunehmen. Dies passiert dann, wenn ein neu zu allozierender Bereich größer ist als das größte freie Fragment
- Bei Speichermedien mit relativ hohen Zugriffszeiten (z.B. Festplatten) kann Fragmentierung zu einer spürbaren Verlangsamung der Lese- und Schreibvorgänge führen, da diese ihre volle Geschwindigkeit technisch bedingt nur bei sequentiellm Zugriff erreichen.



Auswahlstrategien bei der Belegung

- **Next-Fit** – Nächstes passendes Stück
 - Liste wird zyklisch durchlaufen
 - Suche wird an der Stelle der letzten Belegung fortgesetzt
 - Geringerer Suchaufwand im Vergleich zu First-Fit
- **Best-Fit** – Am besten passendes Stück
 - Kleinstes hinreichend große Stück wird genommen
 - Es empfiehlt sich eine Sortierung nach Größe um den Suchaufwand zu minimieren
 - Teilweise bessere Speicherausnutzung, kann aber zu sehr kleinen Stücken führen, die kaum verwendet werden können
- **Nearest-Fit** – Nächstliegendes passendes Stück
 - Wunschadresse wird übergeben
 - First-Fit Suche ab der Wunschadresse

Hauptspeicherverwaltung

Swapping :

Ist nicht genügend HS zum Einlagern eines ausgewählten Prozesses vorhanden, werden andere Prozesse ausgelagert.

Paging :

Virtueller Speicher: Der Speicher wird in Seiten (pages) eingeteilt, es werden nur noch benötigte Teile von Prozessen ein-/ausgelagert. Der Seitenwechsel wird als Paging bezeichnet.

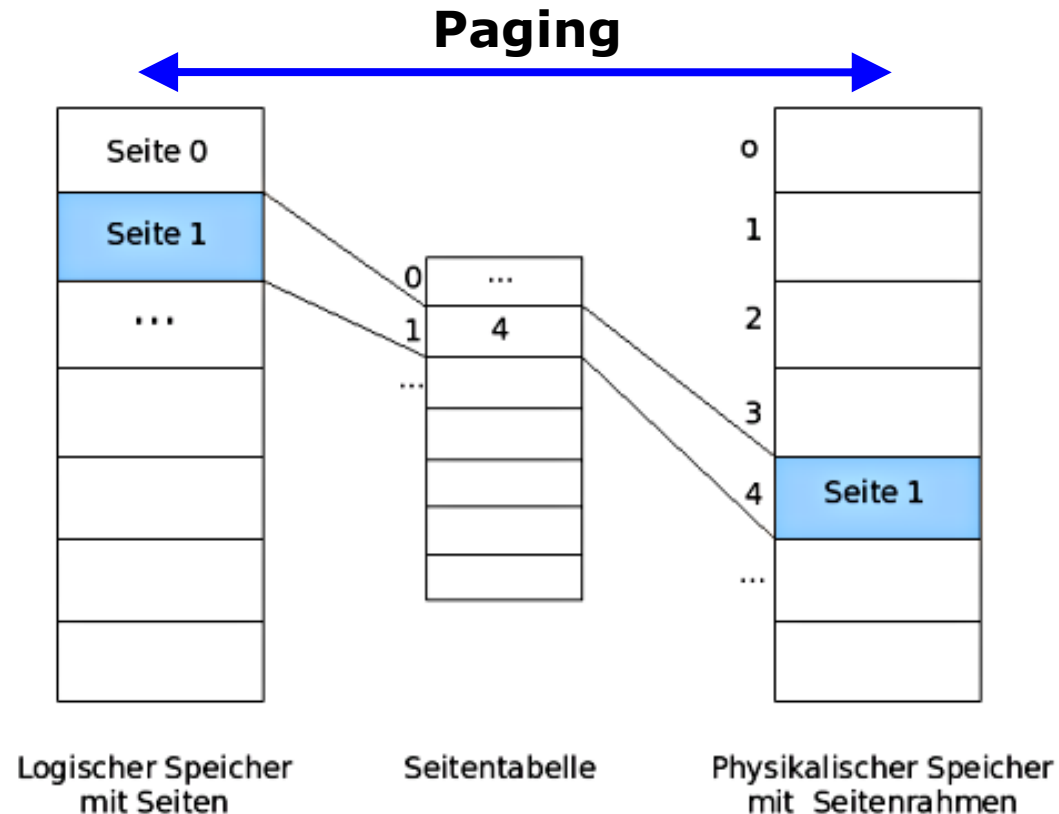
→ es können Programme ausgeführt werden, die größer sind als der HS

Virtueller Speicher - Paging

- Programme können größer sein als der zur Verfügung stehende Speicher – Lösung durch die Nutzung von virtuellem Speicher
- Die Memory Management Unit bildet virtuelle Adressen auf physikalische Adressen ab
- Das BS hält gerade benutzte Teile im Arbeitsspeicher und verwaltet den Rest auf der Festplatte
- Effektive Algorithmen werden zur Ein- und Auslagerung zwischen Arbeits- und Sekundärspeicher eingesetzt
- Der virtuelle Adressraum wird in Speicherstücke gleicher Größe, **Seiten** (Pages) genannt, aufgeteilt
- Der zugeordnete physikalische Speicher heißt **Seitenrahmen** (Page frame); ein Seitenrahmen kann genau eine Seite aufnehmen

Virtueller Speicher - Paging

- Zugriff auf eine nicht im Hauptspeicher eingelagerte Seite führt zu einem Seitenfehler (page fault)
- BS muss die physikalische Seite in den Arbeitsspeicher laden
- Verschiedene Algorithmen zur Seitenersetzung
 - NRU (Not Recently Used)
 - FIFO (First In First Out)
 - LRU (Least Recently Used)



DIENTSTE

Dienste verwalten

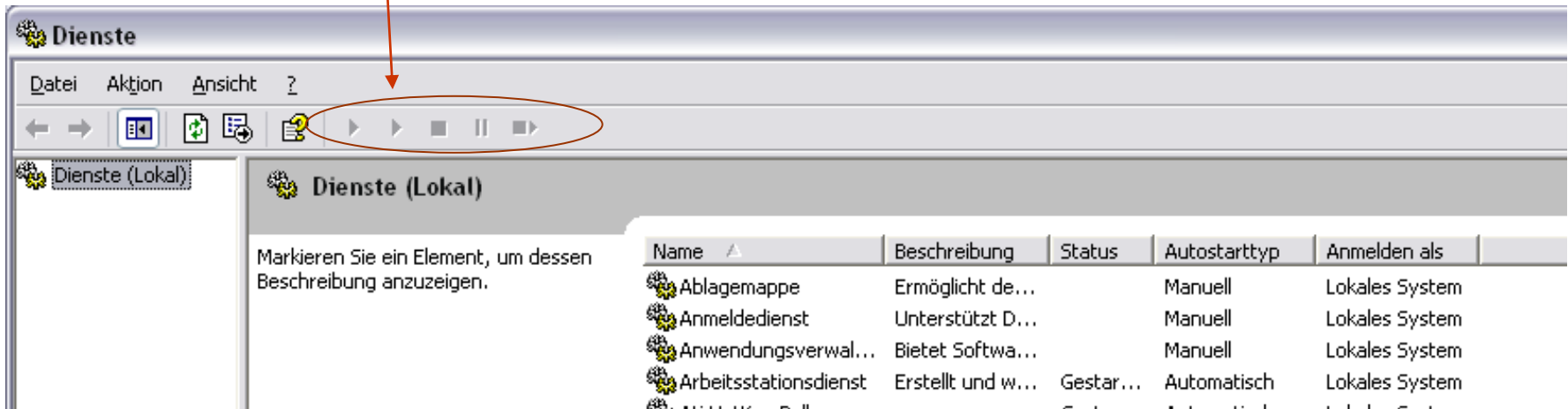
Dienste sind Komponenten des BS, die in der Regel im Hintergrund arbeiten (oberste Schicht im geschützten Kernel-Modus) .

Beispiele: Ereignisprotokolldienst
Anmeldedienst
Arbeitsstationsdienst
...

C:\WINDOWS\system32\services.exe
C:\WINDOWS\System32\lsass.exe
C:\WINDOWS\System32\svchost.exe

Dienste können *gestartet, gestoppt, fortgesetzt, neu gestartet, beendet* werden.

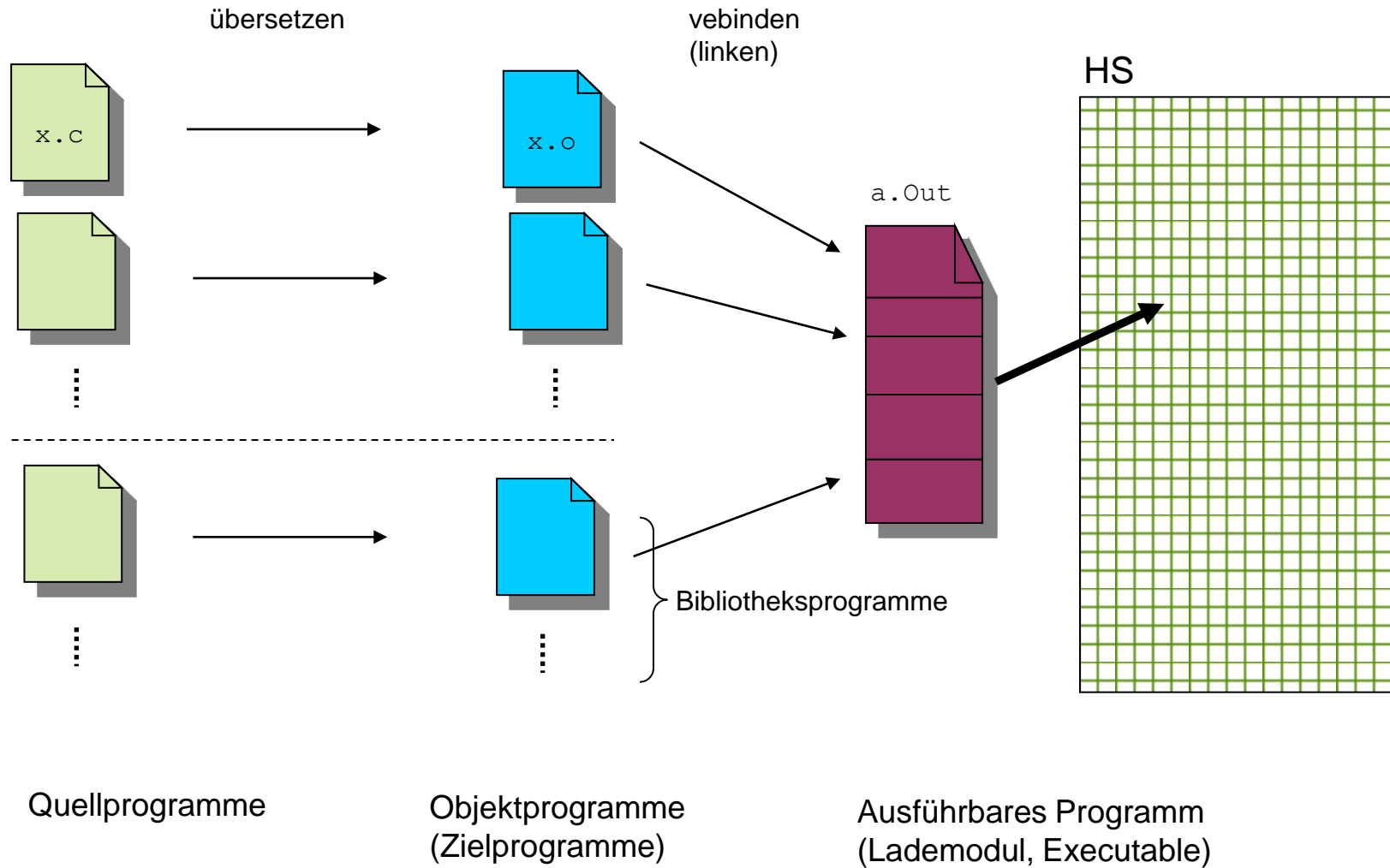
Diensteverwaltung : Start -> Einstellungen -> Systemsteuerung => Verwaltung => Dienste



Programme

- Programme in einer höheren Programmiersprache können nicht direkt auf einem Rechner ausgeführt werden:
 - Sie sind anfangs in einer Textdatei gespeichert
- Verarbeitung durch **Compiler (Übersetzer)**:
 - Ein Compiler überführt das Programm in eine Folge von Maschinenbefehlen (Maschinencode), die der Prozessor des Rechners dann ausführen kann.
- Alternative: **Interpreter**
 - zur Laufzeit wird immer nur ein kleiner Ausschnitt betrachtet, analysiert und durch Maschinenbefehle ausgeführt
- Grundsätzlich können alle Sprachen compilierend oder interpretierend implementiert werden.

Dynamisches Binden (Linken):



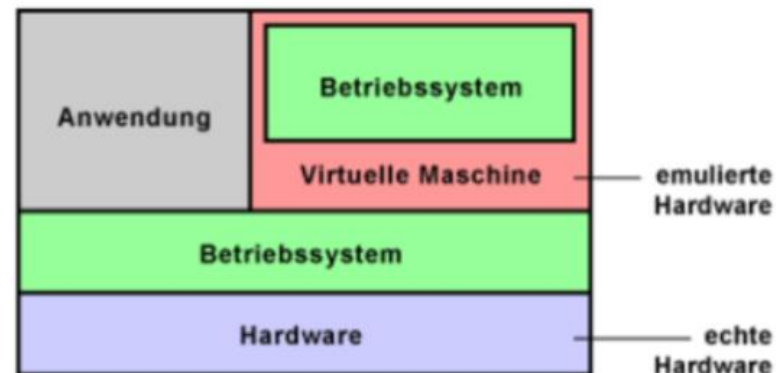
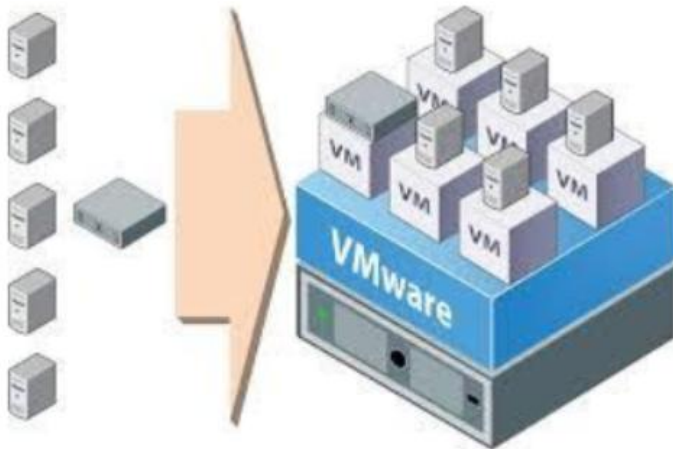
Virtualisierung



Virtualisierung von Systemen

Virtualisierung

Bereitstellung von physikalischen Hardwareressourcen für virtuelle Maschinen.
ein oder mehrere eigenständige Gastsysteme
Möglichkeit mehrere Software-Systeme auf einer Hardware laufen lassen



Es lassen sich die verschiedensten Komponenten von IT-Systemen virtualisieren: Server, Storage (Speicher), Applikationen, Desktops, komplette Netzwerke

What is virtualization?

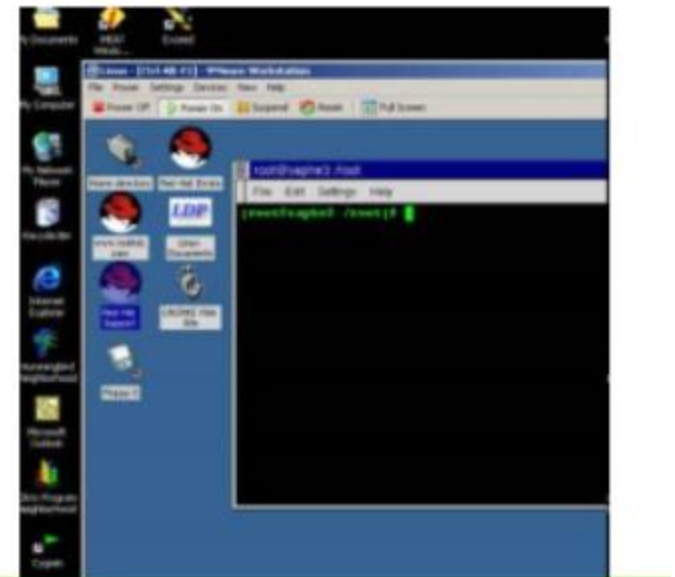
Virtualization is way to run **multiple operating systems** and **user applications** on the same hardware

- E.g., run both Windows and Linux on the same laptop

How is it different from **dual-boot**?

- Both OSes run **simultaneously**

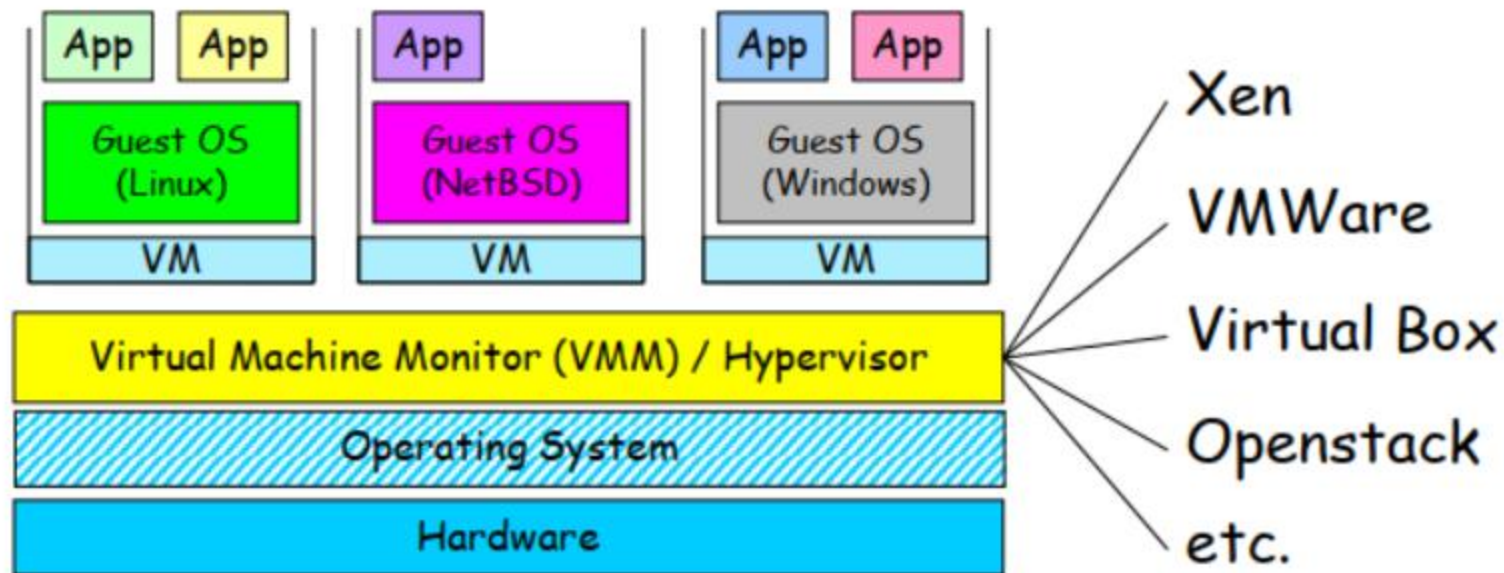
The OSes are completely **isolated** from each other



Was ist Virtualisierung

VM technology allows multiple virtual machines to run on a single physical machine.

Virtual machine (VM) is an isolated runtime environment (guest OS and applications)



Performance: Para-virtualization (e.g. Xen) is very close to raw physical performance!

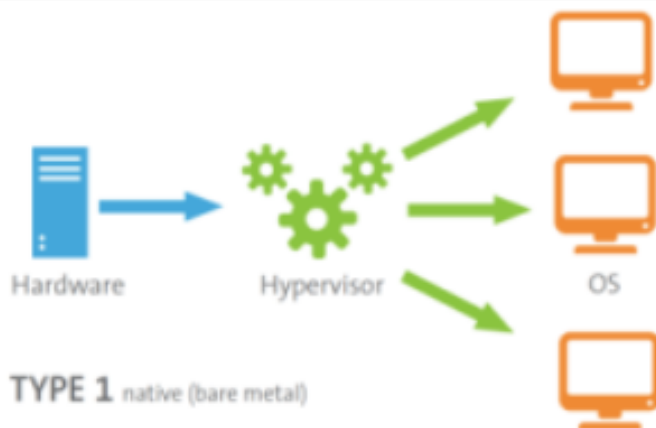
System-Virtualisierung mit Hypervisor

Ein Hypervisor Typ 1, auch „Bare Metal Virtualisierung“

- läuft als Betriebssystem direkt auf der Hardware
- Muss alle Treiber für die gesamte Hardware mitbringen.
- Arbeitet als Hardwarecontroller und Überwacher der VM Betriebssysteme
- Bildet für jede VM die CPU Eigenschaften

Ein Hypervisor Type 2 auch „Hosted Virtualisierung“

- läuft auf einem Betriebssystem
- nutzt die Gerätetreiber des Betriebssystems, um den virtuellen Maschinen CPU, Speicher und I/O zur Verfügung zu stellen.



Was ist ein Hypervisor

A **hypervisor**, a.k.a. a virtual machine manager/monitor (VMM), or virtualization manager, is a program that allows multiple operating systems to share a single hardware host.

Each guest operating system appears to have the host's processor, memory, and other resources all to itself. However, the hypervisor is actually controlling the host processor and resources, allocating what is needed to each operating system in turn and making sure that the guest operating systems (called virtual machines) cannot disrupt each other.

Ziele und Gründe der Virtualisierung

Reduzierung des Stromverbrauchs

Kostenreduzierung für Hard- und Software

Bessere Verteilung von Ressourcen

Komplette Sicherung und schnelle Wiederherstellung der Virtuellen Maschinen

Erhöhung der Ausfallsicherheit

Leichte Erweiterbarkeit

wichtige Voraussetzung für Cloud Computing

COMPUTER- WARTUNG

Hardware

- Hardware
 - Reinigung (Tastatur, Lüfter)
 - Akku – Energiemanagement
 - Temperaturüberwachung (CPU/GPU)

Software

- Backup (z.B. FreeFileSync)
- CCleaner (StartUp, Registry, Files)
- AWDcleaner
- Virenschutz (AVG Antivirus, Avira, Avast)
- SW Updates incl. Betriebssystem
- Dateiverwaltung: DropIt, SpaceSniffer, Treesize, TotalCommander
- Alte Programme deinstallieren
- Wiederherstellungspunkt setzen
- Funktionierendes Image Ziehen (Clonezilla)
- Defragmentierung

Sicherungstechniken

**Generationen-
prinzip**

Unterscheidung:

Voll-Backup

**Inkrementelles
Backup**

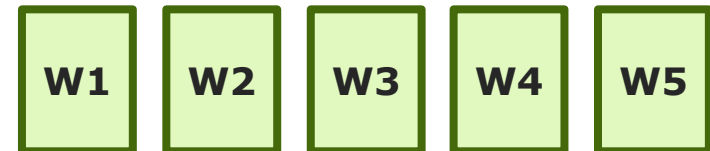
Söhne

Wöchentlich überschrieben



Väter

Monatlich überschrieben



Grossväter

Jährlich überschrieben

