

Kapitel 2

Echtzeitbetriebssysteme

Implementierungsaspekte der Taskverwaltung mit Taskliste mit Taskkontrollblock und Taskkontext

Taskkontext muss bei Unterbrechen/Blockieren gerettet werden.

Taskkontext für schwergewichtige Tasks:		
Taskverwaltung	Speicherverwaltung	Ein-/Ausgabeverwaltung
Programmzähler Statusregister Steuerregister Stapelzeiger Registerblock Taskidentifikation Taskpriorität Taskparameter Elterntask Kindertasks	Startadresse, Größe und Zustand von: Taskcode Taskdaten Taskstapel dynamisch belegtem Taskspeicher (Taskhalde)	Erteilte Ein-/Ausgabebefehle Zugriffsrechte auf Geräte Angeforderte Geräte Belegte Geräte Zeiger auf Gerätepuffer Zustand belegter Geräte Zeiger auf offene Dateien Zugriffsrechte auf Dateien Zeiger auf Dateipuffer Zustand offener Dateien

Taskkontext für leichtgewichtige Threads:

- Programmzähler
- Statusregister
- Steuerregister
- Stapelzeiger
- Registerblock

Die Speicherverwaltung teilt den verfügbaren Speicher mittels einer **Speicherabbildungsfunktion** M zu:

M : logische Adresse \rightarrow physikalische Adresse

Logische Adresse: Speicheradresse innerhalb einer Task

Physikalische Adresse: wirkliche Arbeitsspeicheradresse

Modelle der Speicherverwaltung:

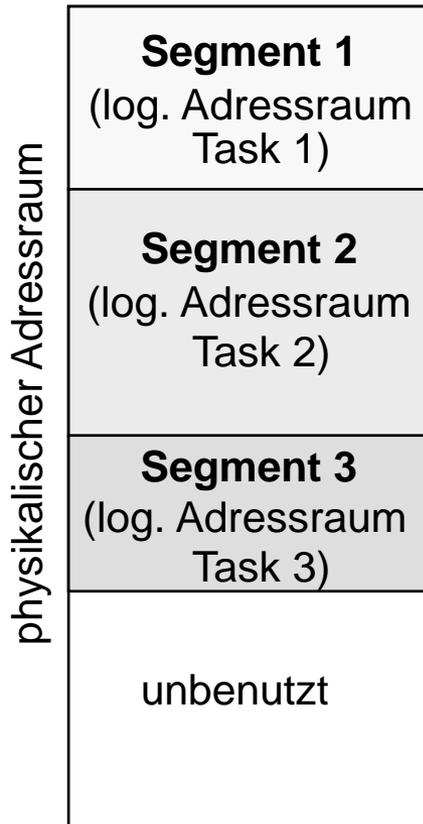
Speicherzuteilung:

- Statische Speicherzuteilung
- Dynamische Speicherzuteilung
- Nichtverdrängende Speicherzuteilung
- Verdrängende Speicherzuteilung

Adressbildung und Adressierung:

- Reelle Adressierung
- Virtuelle Adressierung
- Lineare Adressbildung
- Streuende Adressbildung

Lineare Adressbildung durch Segmente, statische Zuteilung, reelle Adressierung, keine Verdrängung



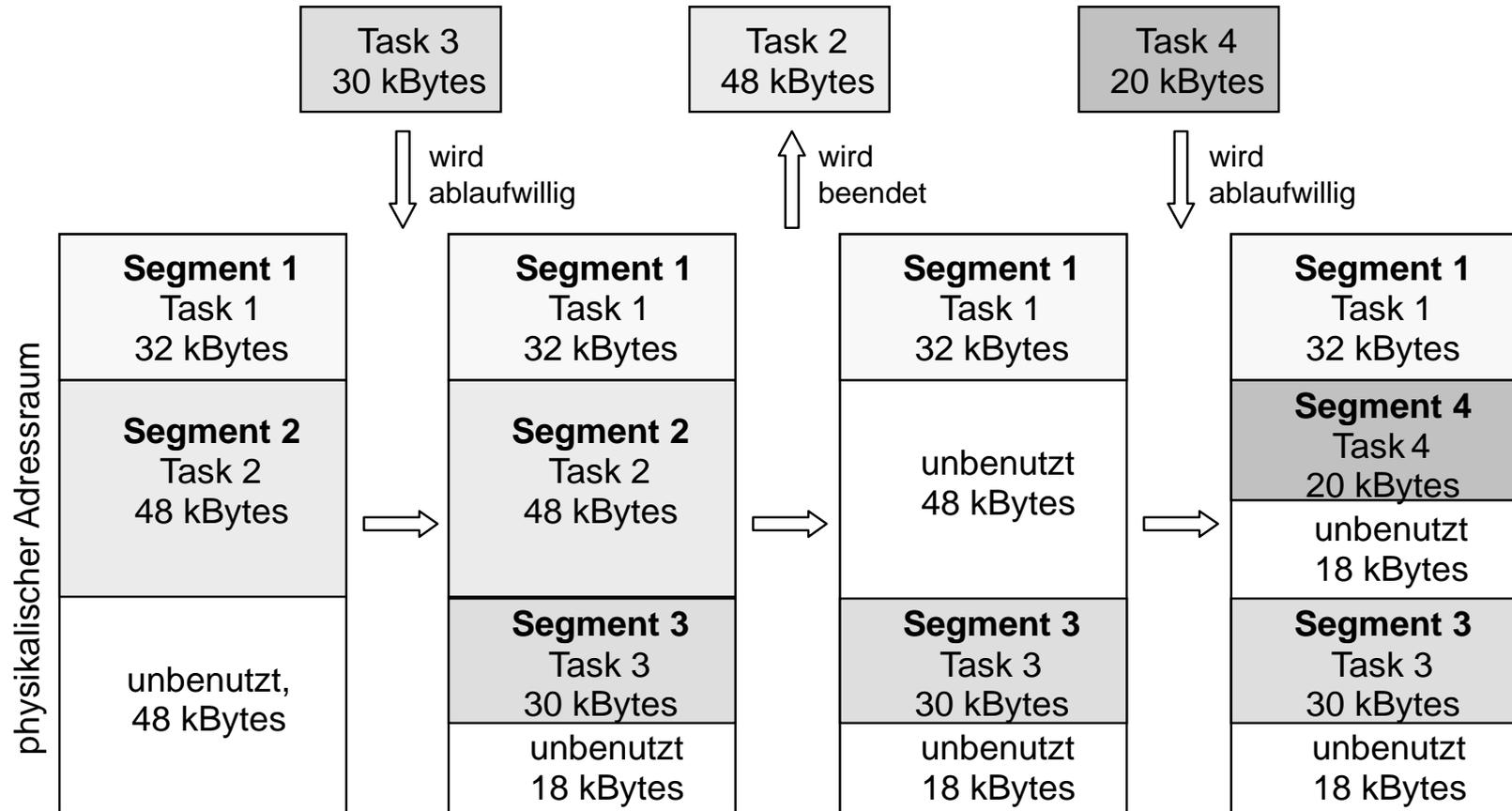
Vorteile:

- Tasks im Speicher verschiebbar
- Tasks gegeneinander geschützt
- Zeitverhalten sehr gut vorhersagbar

Nachteile:

- Speicherschema starr, nicht online änderbar
- Für jede Task maximaler Speicher
- Bei variierender Anzahl von Tasks muss für alle Tasks Speicher reserviert werden, kein Teilen von Speicher zwischen Tasks

Lineare Adressbildung durch Segmente, dynamische Zuteilung, reelle Adressierung, keine Verdrängung

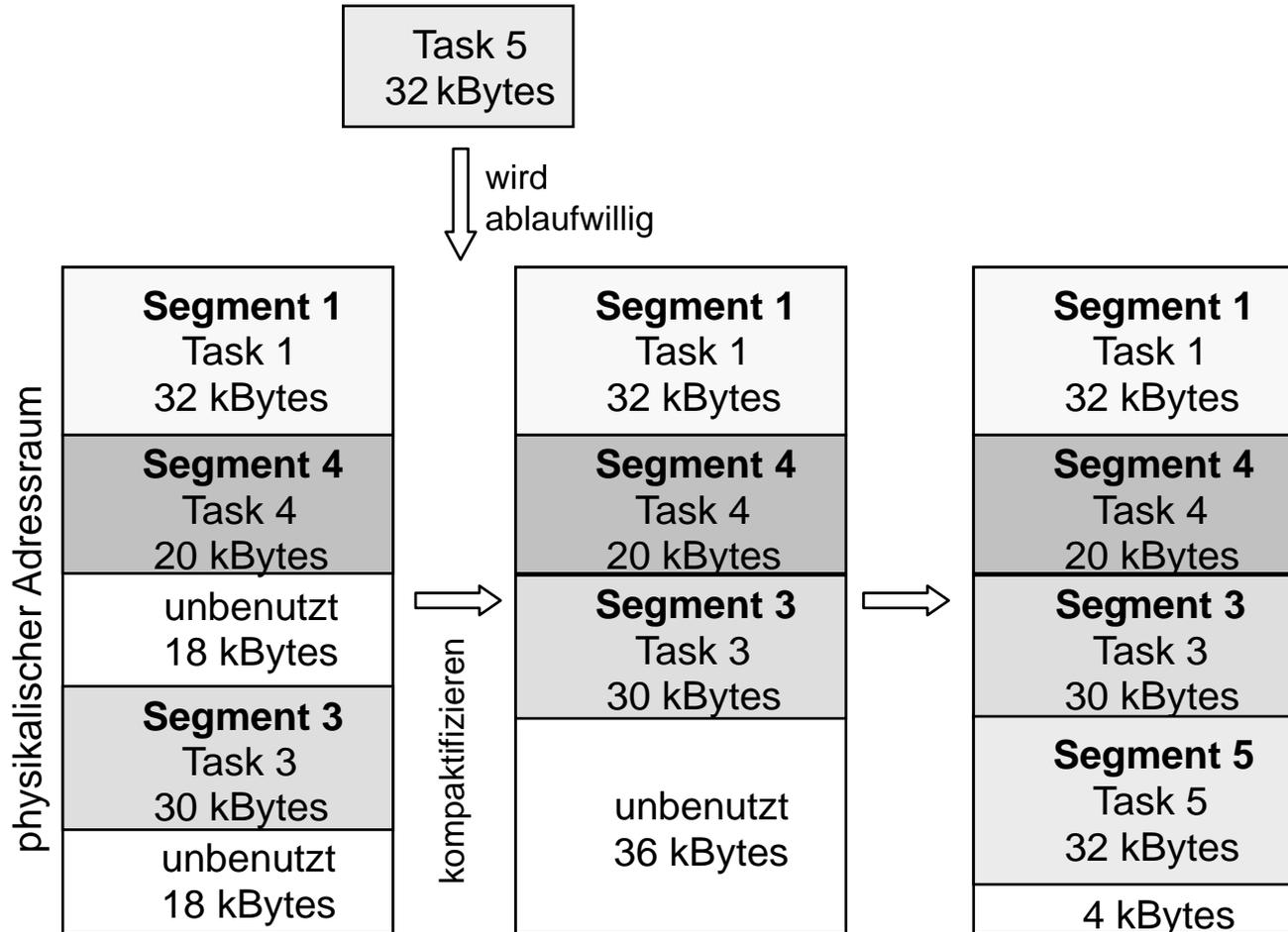


Tasks können sich phys. Speicher teilen.

So lange keine Verdrängung zur Laufzeit ist die Speicherzugriffszeit konstant.

Speicherbereinigung wegen löchrigem Speicher.

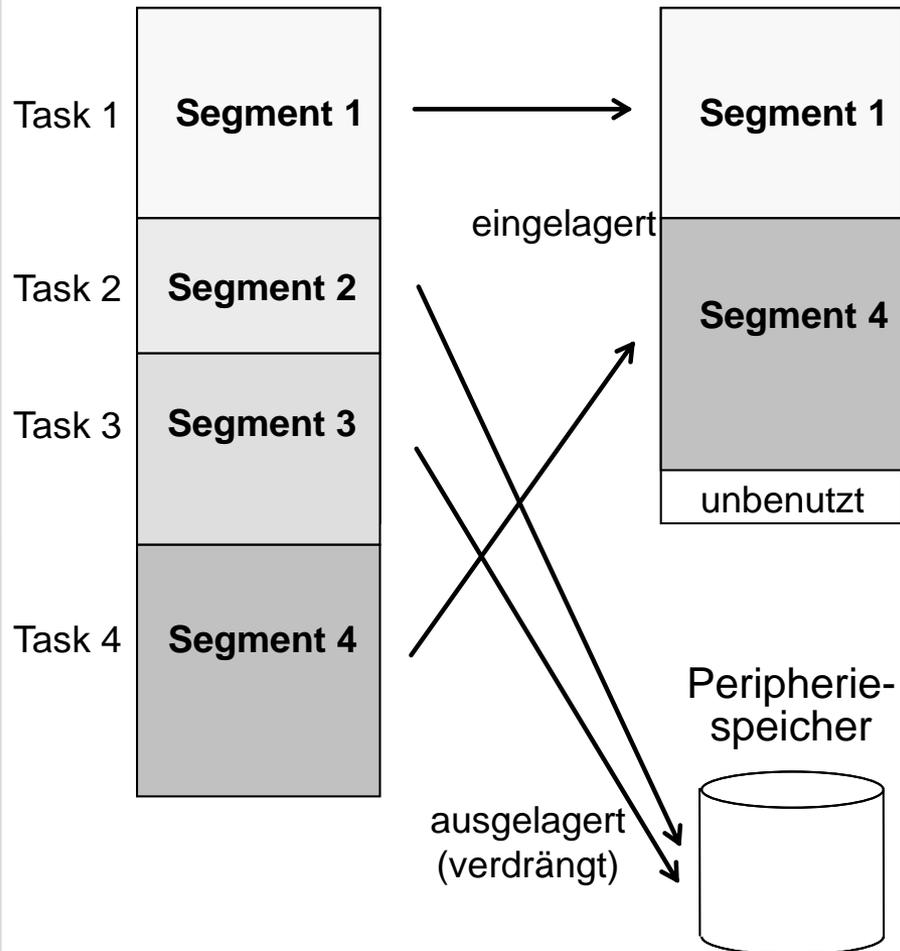
Lineare Adressbildung mit Speicherbereinigung



Wegen Segmentverschiebungen für Echtzeit problematisch.

Lineare Adressbildung mit virtueller Adressierung und Verdrängung

logischer Adressraum physikalischer Adressraum



Verdrängungsstrategien (auslagern):

- FIFO (*First In First Out*), am längsten im Sp.
- LRU (*Least Recently Used*), a. I. nicht mehr ben.
- LFU (*Least Frequently Used*), am seltensten ben.
- LRD (*Least Reference Density*), ger. Zugriffsdichte

Zuteilungsstrategien (einlagern, wohin):

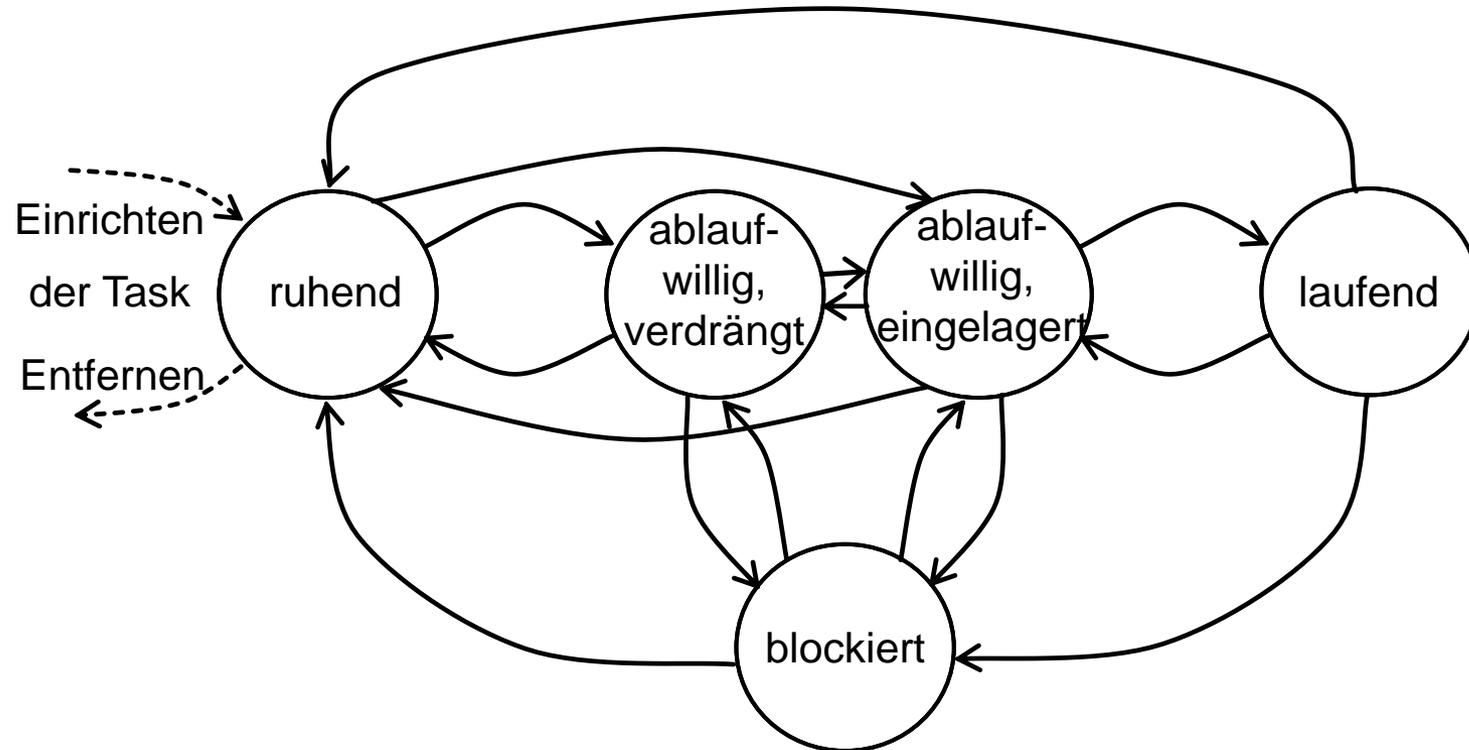
- *First Fit* (erstes)
- *Best Fit* (kleinstes)
- *Worst Fit* (größtes)

Nachschubstrategien (wann einlagern):

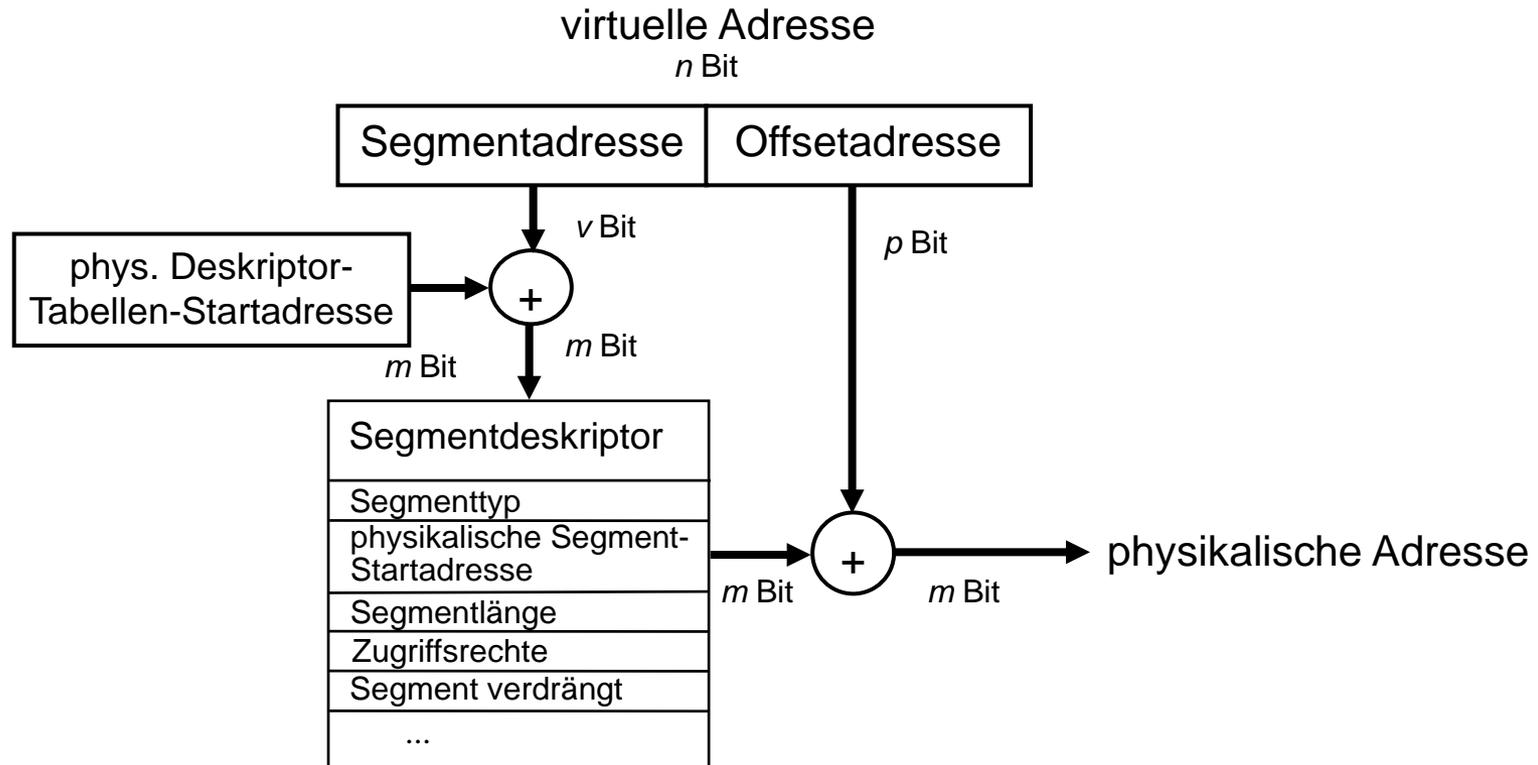
- Verlangend (*Demand Fetch*)
- Vorausschauend (*Anticipatory Fetch*)

Echtzeitbetriebssysteme: **Verriegeln** von einzelnen Tasks, d.h. keine Verdrängung

Taskzustände bei verdrängender Speicherverwaltung

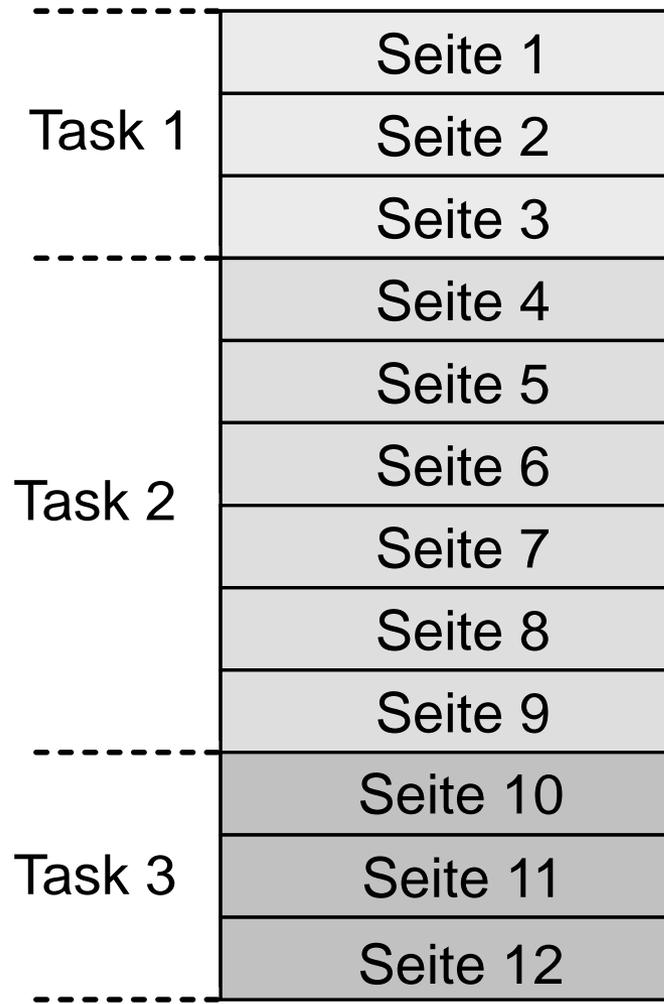


Realisierung der linearen Adressbildung mit Segmenten

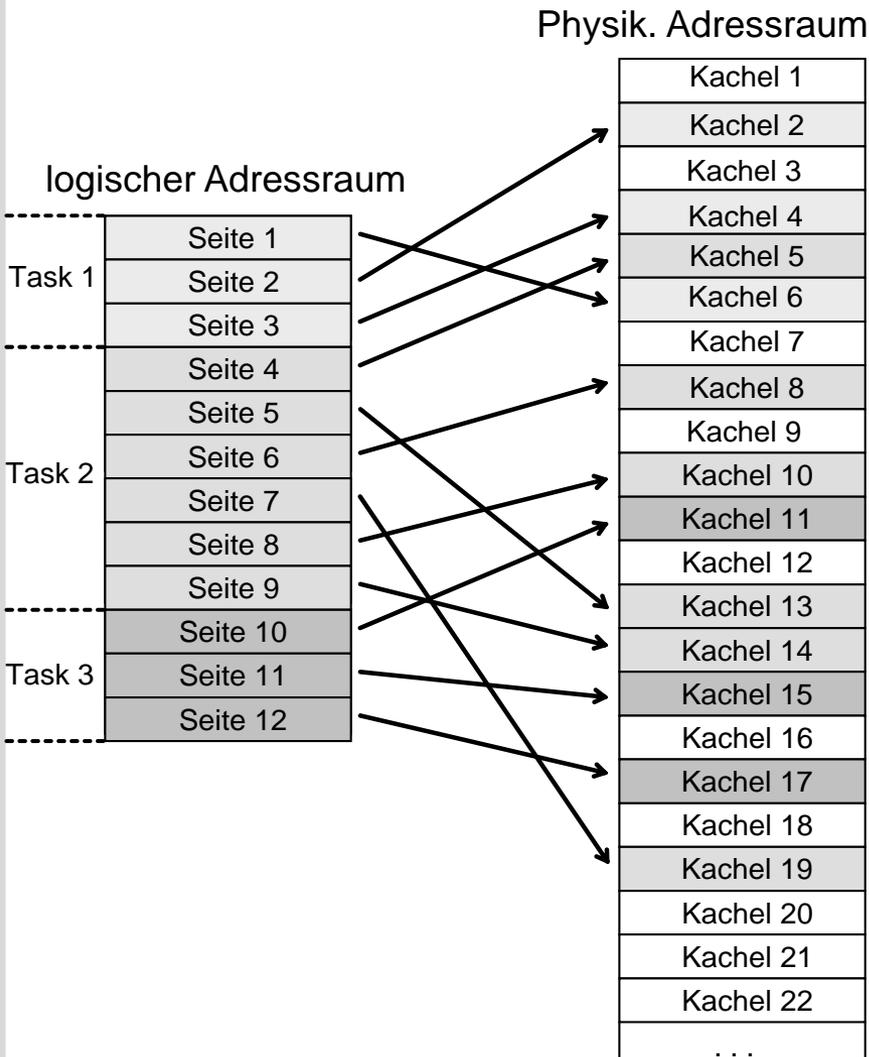


in der Segmentdeskriptor-Tabelle im Arbeitsspeicher

Streuende Adressbildung: Aufteilung der Tasks auf Seiten



Streuende Adressbildung: Zuordnung von Seiten zu Kacheln



Die Zuordnung von Seiten zu Kacheln erfolgt in der Regel ohne Beachtung der sequentiellen Reihenfolge der Seiten

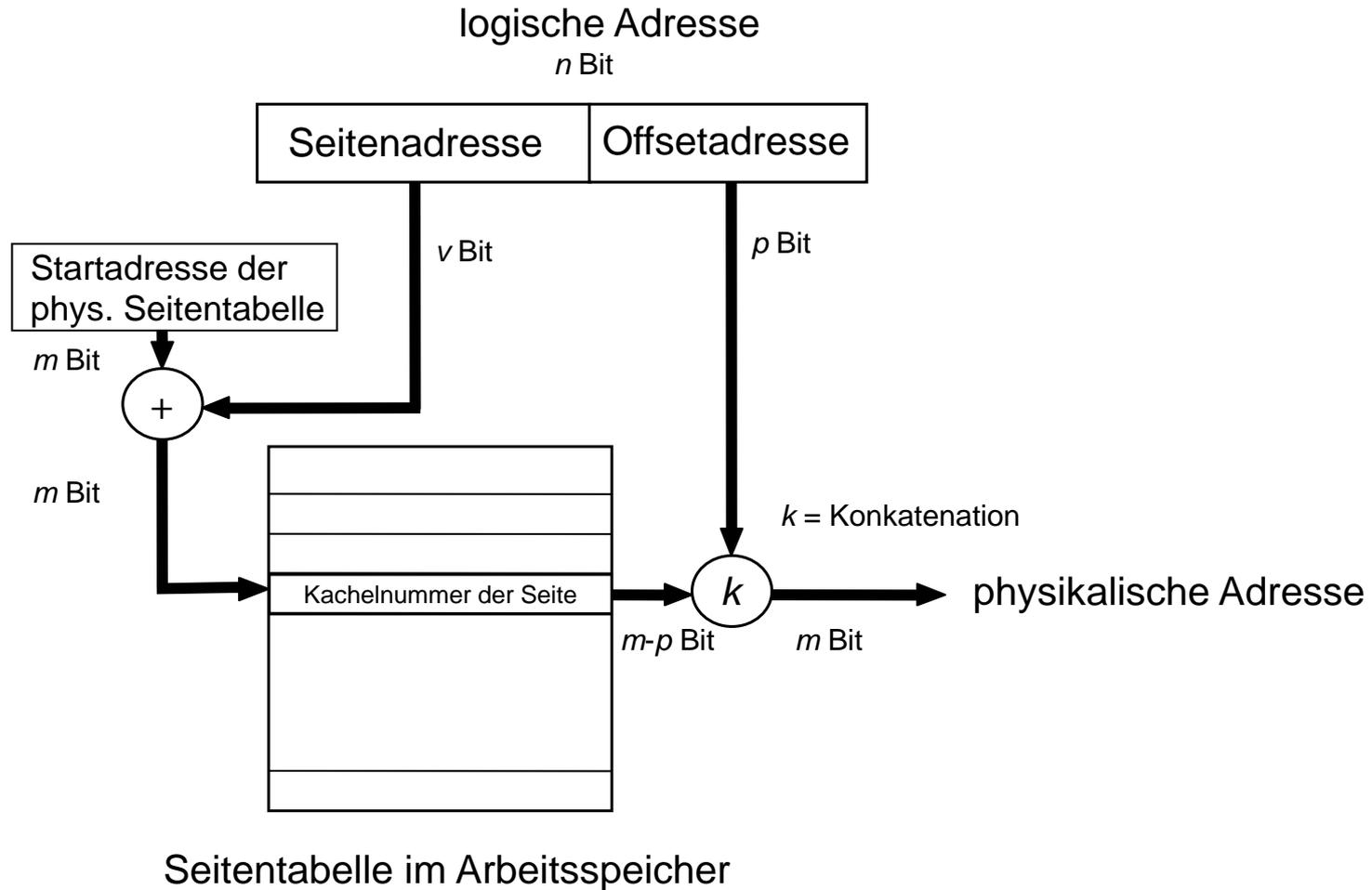
Vorteile:

- einfache Zuweisung, keine Zuteilungsstrategie
- dynamische Taskgrößenänd. einfacher und zeitlich besser vorhersagbar
- keine Speicherbereinigung
- Virtuelle Adressierung benötigt nur Verdrängungs- und Nachschubsstrategie

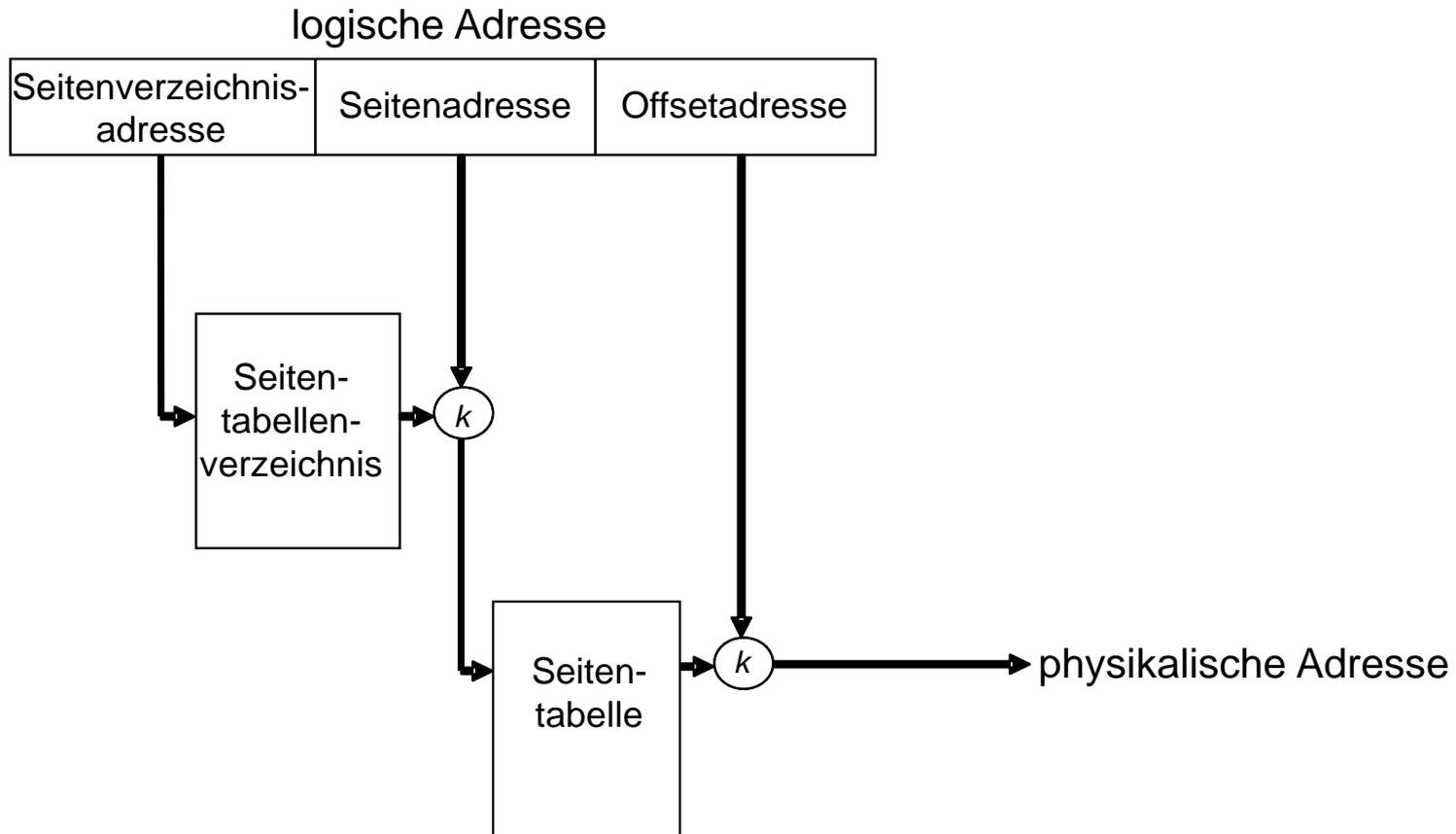
Nachteile:

- letzte Seite nur teilweise voll
- höherer Seitenverwaltungsaufwand
- durch kleine Seiten mehr Datentransfers zwischen Haupt- und Peripheriespeicher

Realisierung der streuenden Adressbildung mit Seiten



Hierarchischer Aufbau der Seitentabellen



Kombination von linearer und streuender Adressbildung

logische Adresse



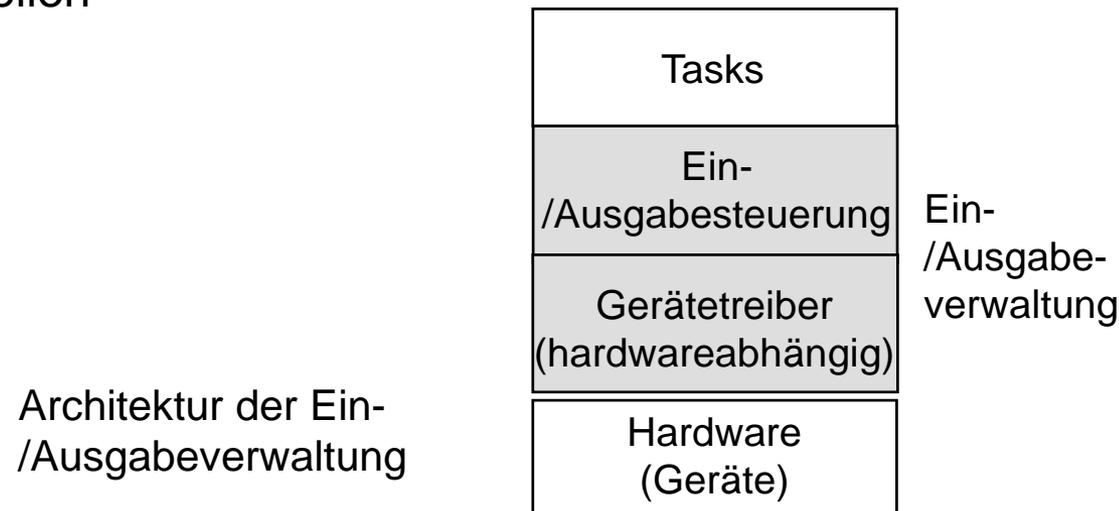
Kombination der Vorteile

Aufgaben der IO-Verwaltung:

- Zuteilen und Freigeben von Geräten
- Benutzen von Geräten

Die angeschlossenen Geräte unterscheiden sich hierbei stark in Geschwindigkeit und Datenformat

=> das Betriebssystem muss dem Anwenderprogramm die Schwierigkeiten der Kommunikation abnehmen und einfache, transparente Schnittstellen zur Verfügung stellen

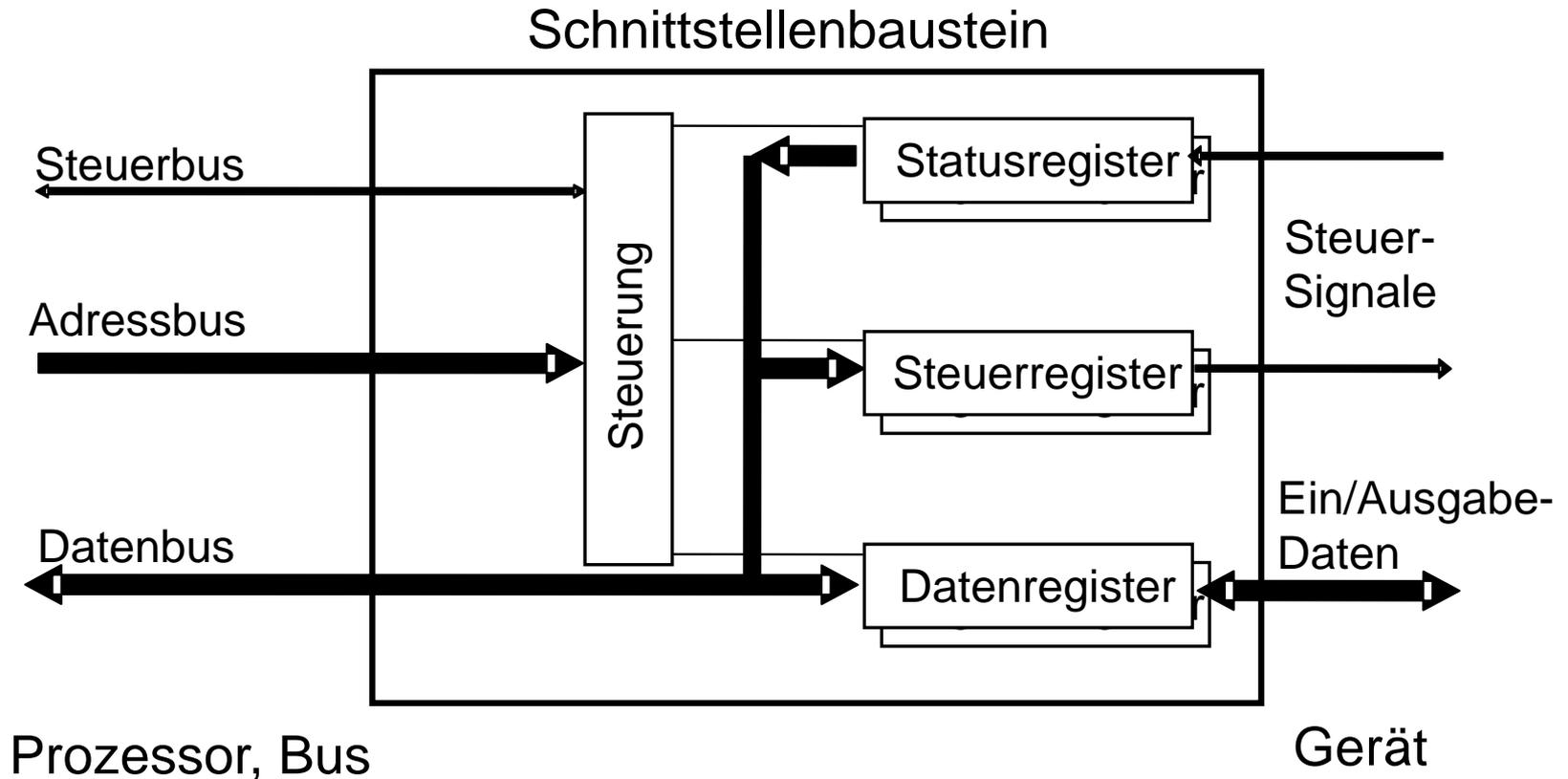


Die **wichtigsten Teilfunktionen** der Ein/Ausgabe-Steuerungsschicht sind:

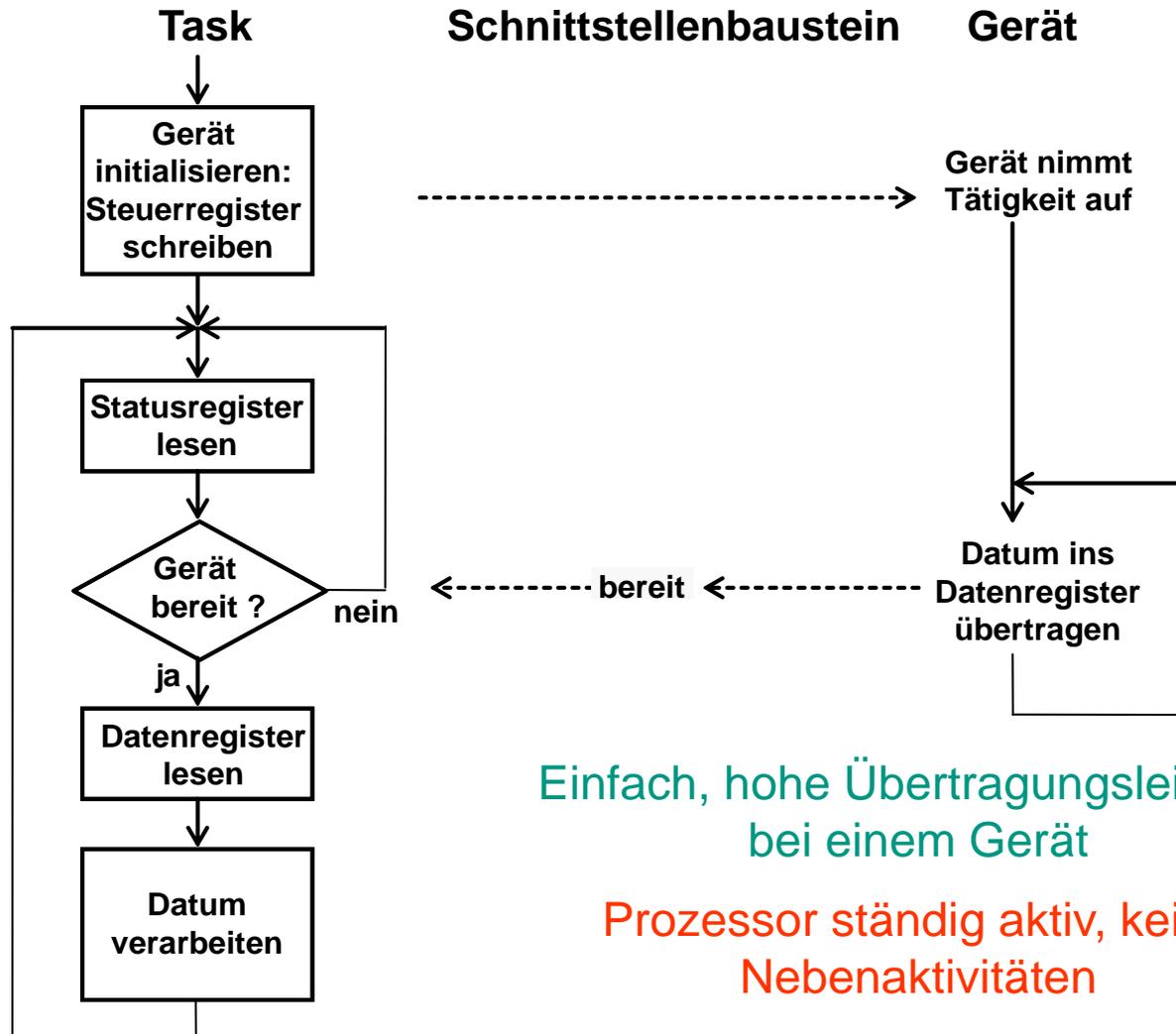
- Symbolische Namensgebung
- Annahme von Ein-/Ausgabeanforderungen
- Vergabe und Zuteilung von Geräten
- Synchronisation
- Schutz der Geräte
- Kommunikation mit den Gerätetreibern
- Pufferung
- Einheitliches Datenformat

Synchronisationsmechanismen

Kommunikation zwischen Tasks und Peripherie findet häufig über SSB statt



Synchronisation durch Polling

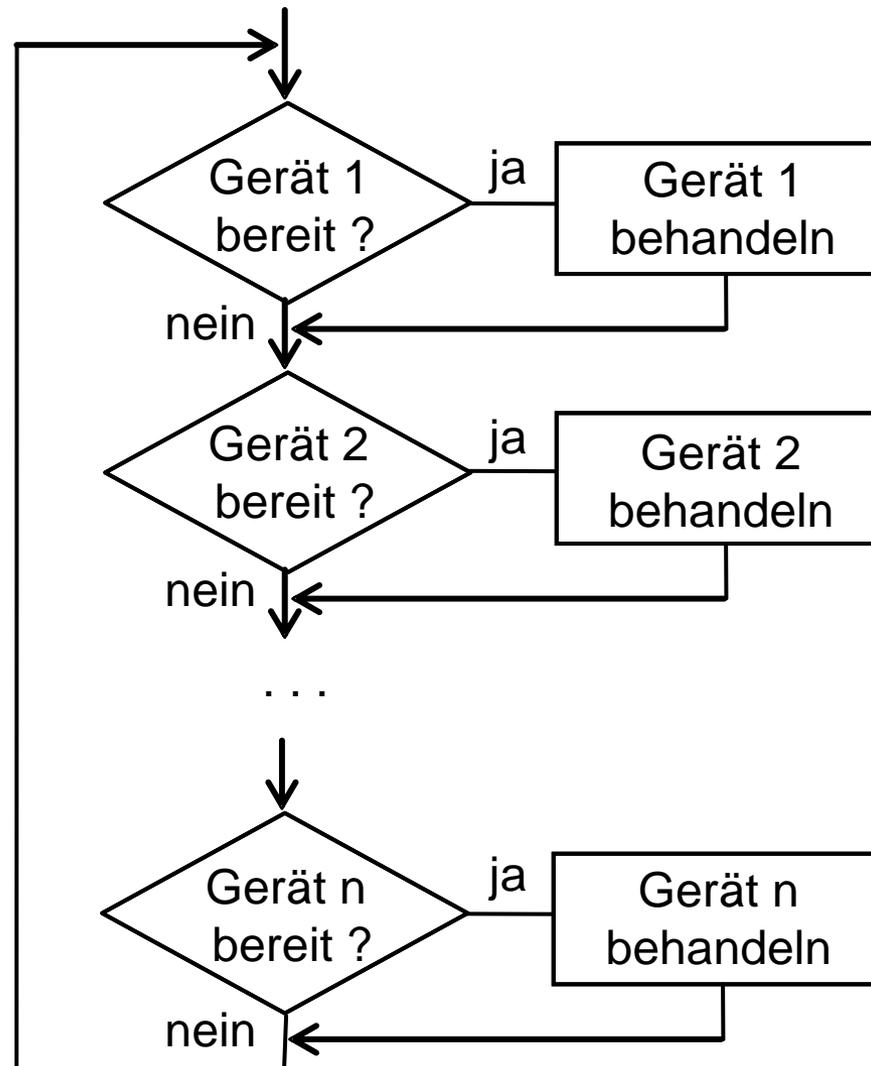


Einfach, hohe Übertragungsleistung
bei einem Gerät

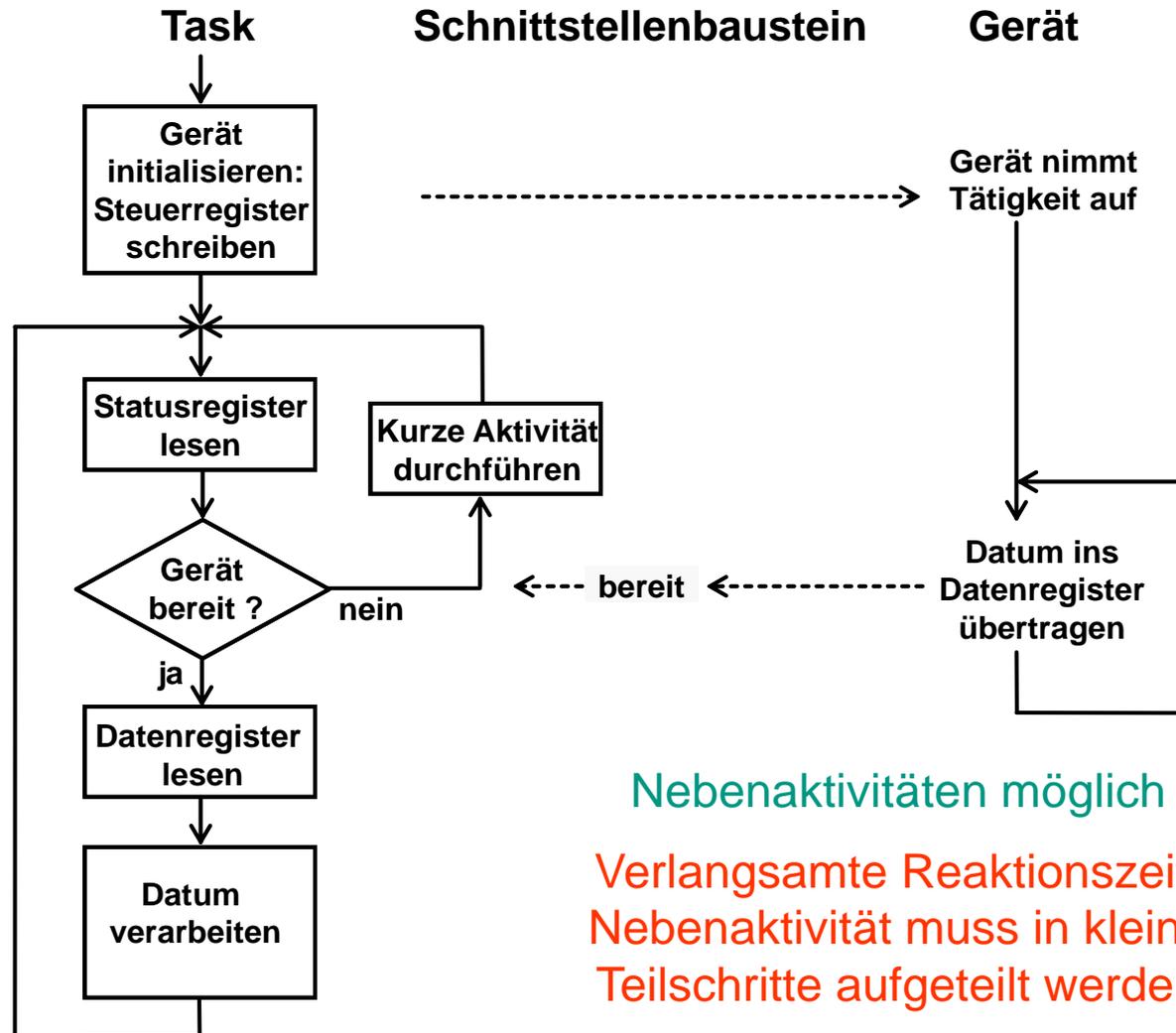
Prozessor ständig aktiv, keine
Nebenaktivitäten

Polling von mehreren Geräten innerhalb einer Task

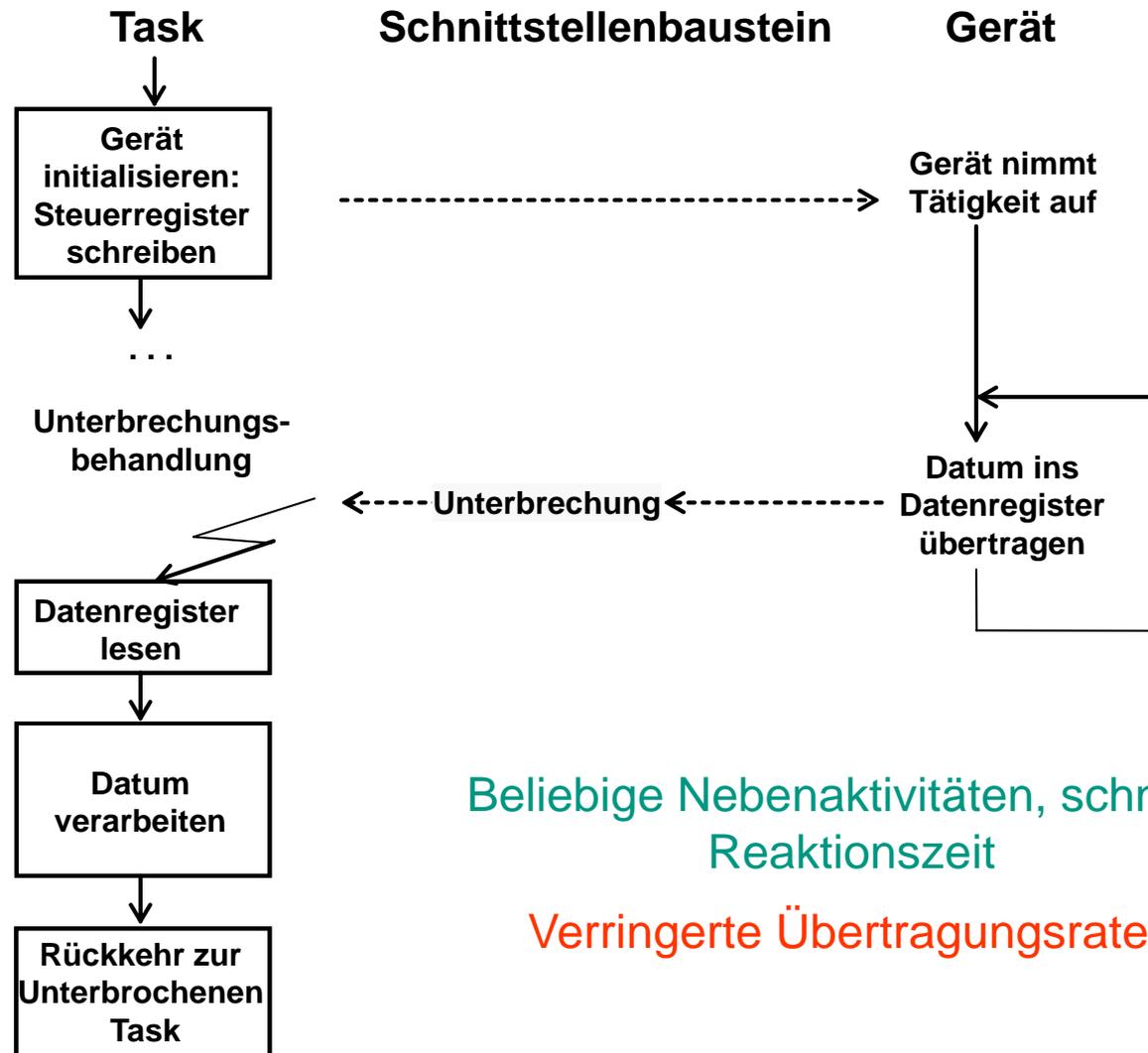
Verlangsamte
Reaktionszeit bei
mehreren Geräten



Synchronisation durch Busy Waiting



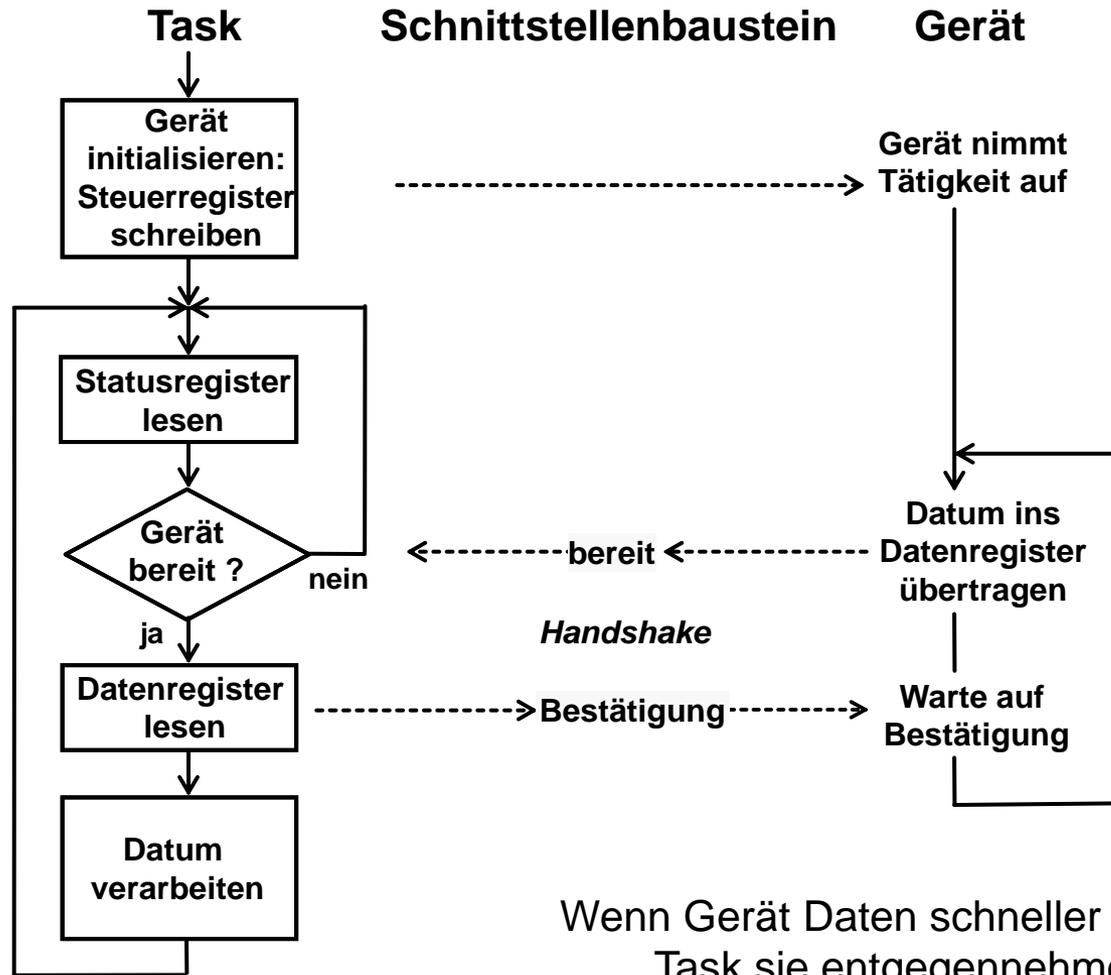
Synchronisation durch Unterbrechung



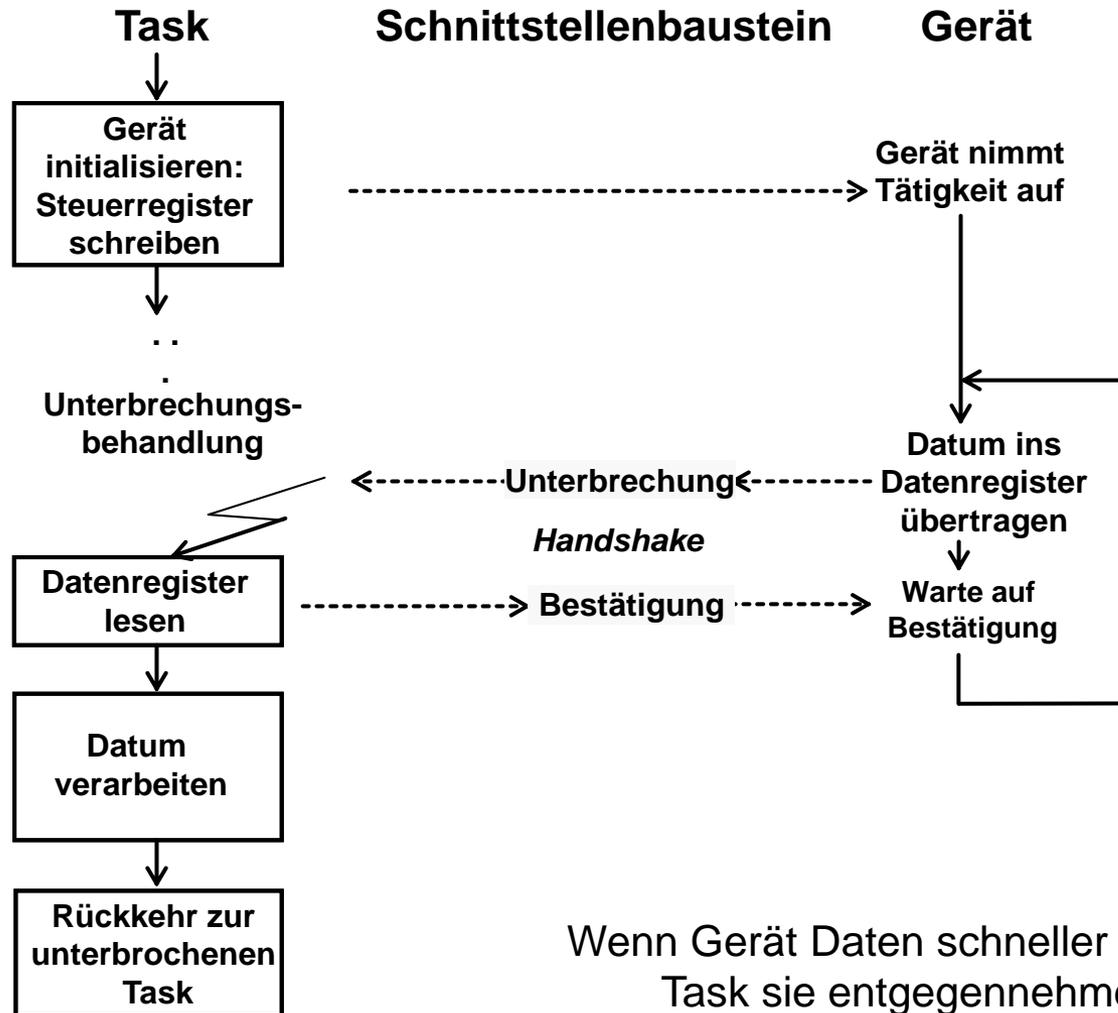
Beliebige Nebenaktivitäten, schnelle Reaktionszeit

Verringerte Übertragungsraten

Synchronisation durch Polling mit Handshake

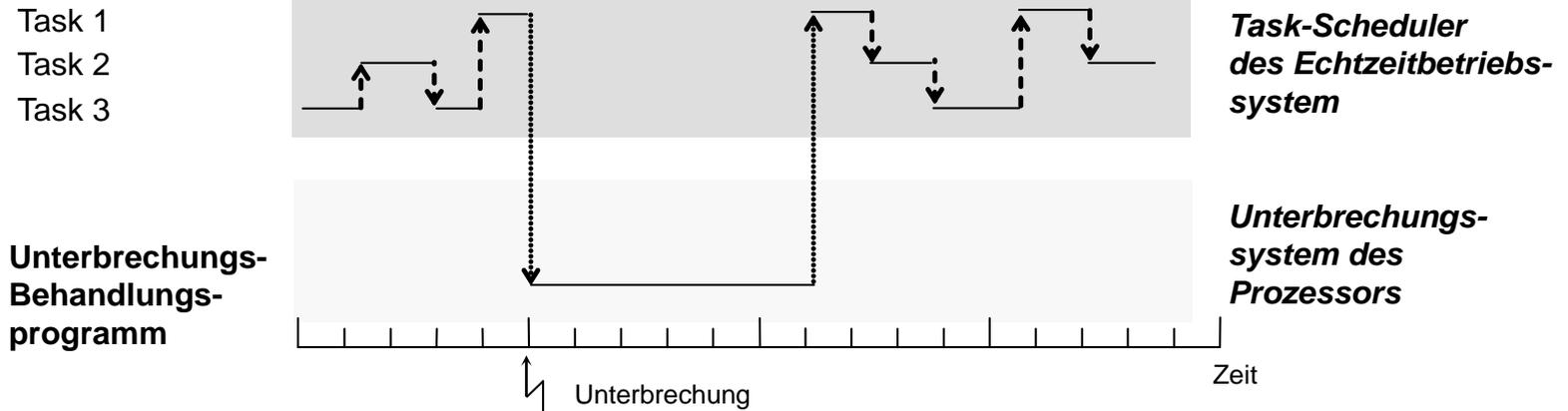


Synchronisation durch Unterbrechung mit Handshake

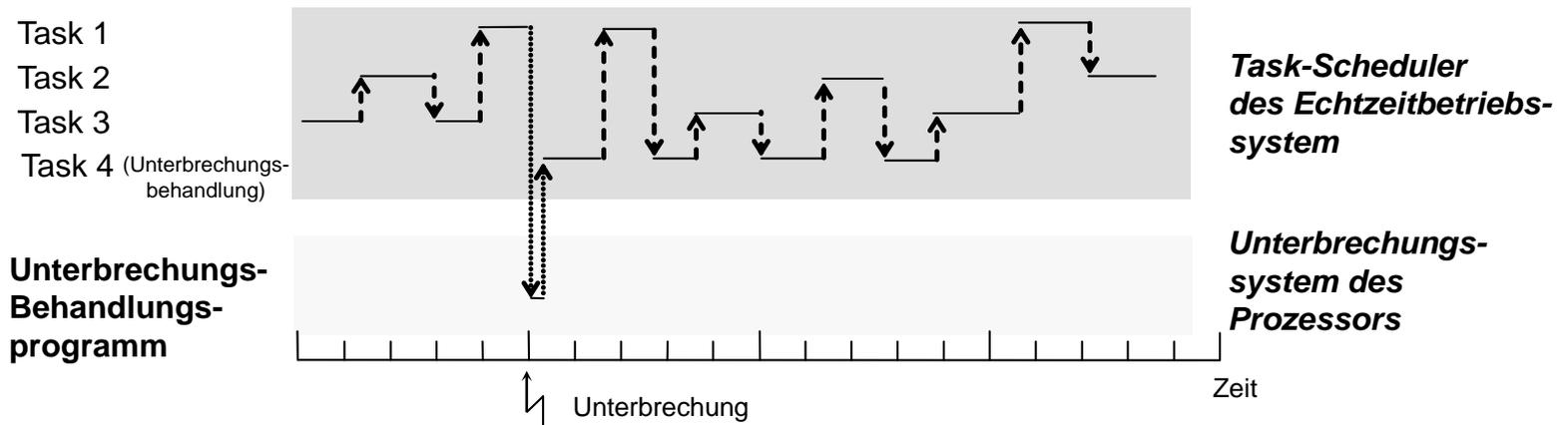


Wenn Gerät Daten schneller liefert als die Task sie entgegennehmen kann

Unterbrechungsbehandlung in Echtzeitbetriebssystemen



a) Unterbrechung der Taskverarbeitung



b) Integration der Unterbrechung in die Taskverarbeitung

Klassifizierung von Echtzeitbetriebssystemen

	Threads	Tasks	Semaphore	Komm. über DPRAM, FIFO Nachrichten	Scheduler	Speicherverwaltung Physikal. Adr.	Speicherverwaltung Virtuelle Adr.	Speicherverwaltung Einfache E/A Treiber	Ein/Ausgabe Verw. Einfache E/A Treiber	Komfortable E/A Treiber, Netzwerke	Filesystem
Minimales Echtzeitbetriebssystem (MEBS)	X		X	X	X	X			X		
Controller System (CS)	X		X	X	X	X			X		X
Dediziertes System (DS)	X		X	X	X		X	X			X
Betriebssystemaufsatz (BA)	X	X	X	X	X	X	X			X	X
Allgemeines Echtzeitbetriebssystem (EBS)	X	X	X	X	X	X	X			X	X

1. Entwicklungs- und Zielumgebung

- Programmiersprache
- Zielsystem (ZE)- Crossentwicklungsumgebung (CE)
 - Ladezeiten bei Crossentwicklung
 - Debug-Möglichkeiten (CE): Quellcode-Debugging (C, C++, Java), Zeitverfälschung bei CE-Debugging
 - EBS und EU vom selben Hersteller

2. Modularität und Kerngröße

- Konfigurierbarkeit auf Anwendung
- Minimaler Mikrokern für eingebettete Systeme

3. Anpassbarkeit an verschiedene Zielumgebungen (Prozessoren) für Automatisierung

- Ist EBS an Mikrocontroller/Signalprozessor mit spezieller Peripherie anpassbar?
- ROM-Fähigkeit für Code mit RAM für Daten
- Feldbusse verfügbar

4. Allgemeine Eigenschaften

- Bedieneroberfläche (Grafik, Kommandos, keine)
- Bibliotheken (Mathem., Graph., Textverarbeitung, ...)
- Werkzeug für GUI-Echtzeitanwendungen
- Werkzeuge zur Versionsverwaltung, Datenhaltung oder zur Programmentwicklung

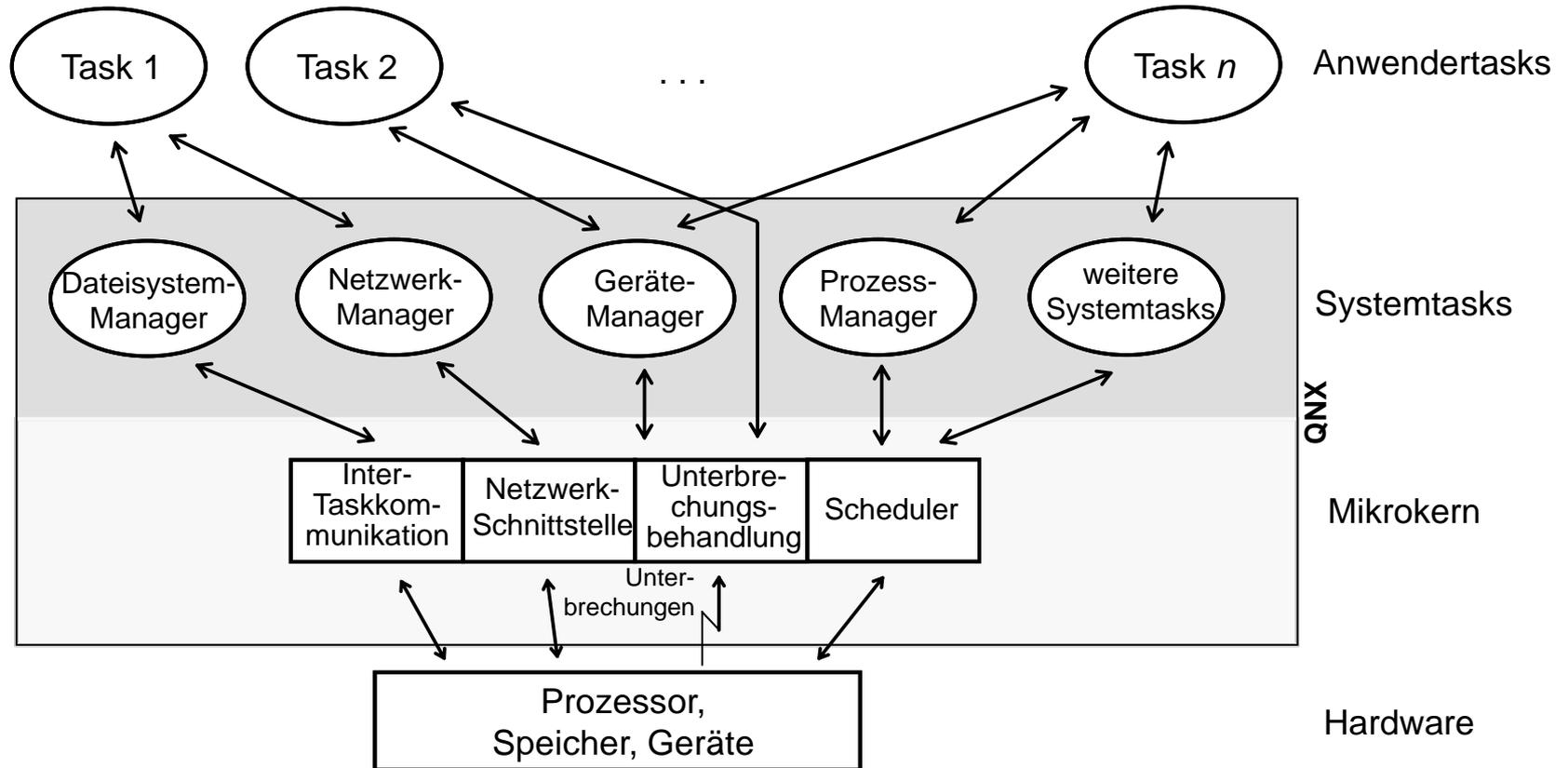
5. Leistungsdaten

- maximale Taskanzahl (Tasks, Threads)
- Welche Scheduling-Strategien
- Anzahl unterschiedlicher Prioritätsebenen (64-256)
- Taskwechselzeiten (Threads schneller als Tasks)
- Latenzzeiten bei Unterbrechungen
- Behandlung von Unterbrechungen (direkt – indirekt)
- EBS-Klasse für harte, feste, weiche Echtzeitanwendungen

Industrielle Echtzeitbetriebssysteme

Produkt	QNX	POSIX.4	RT-Linux	VxWorks	OS-9	CMX	Windows Emb.
Hersteller	QNX Software Systems Ltd.	Posix	GPL (GNU Public License)	Wind River	Microware	CMX Systems	Microsoft
Typ	DS, EBS	Standard für EBS, BA	BA	EBS	EBS	MEBS	EBS
Zielsystem	IA32	IA32, IA64, PowerPC	IA32, PowerPC, ARM	IA32, PowerPC, 680XX, div. Mikrocontroller	IA32, PowerPC, 680XX	Diverse Mikrocontroller	IA32, PowerPC, MIPS, ARM
Sprachen	C, C++	C, C++, Java, Ada	C, C++, Java	C, C++, Java	C, C++, Java	C	C, C++, Java
Dateisystem	Unix, Windows	Unix	Unix, Windows	Unix, Windows	Windows	-	Windows
GUI	X-Win	X-Win	X-Win	X-Win	X-Win	-	Windows
Netzwerk	TCP/IP	TCP/IP, UDP	TCP/IP, UDP	TCP/IP	TCP/IP	-	TCP/IP
Feldbus	-	-	-	CAN, ProfiBus	CAN, ProfiBus, Interbus S	-	-
Scheduling	FPP, FPN, Timeslice, FIFO	FPP, FPN, Timeslice, Benutzerdef.	FPP, FPN, Timeslice	FPP, FPN, Timeslice	FPP, FPN, Timeslice	FPP, FPN, Timeslice	FPP, FPN
Speicher- verwaltung	Stat., Dyn. reel-, virt. Adr.	-	Stat., Dyn. reel-, virt. Adr.	Stat., Dyn. reel-,virt. Adr.	Statisch reele Adr.	Statisch reele Adr.	Stat., Dyn. reel-,virt. Adr.

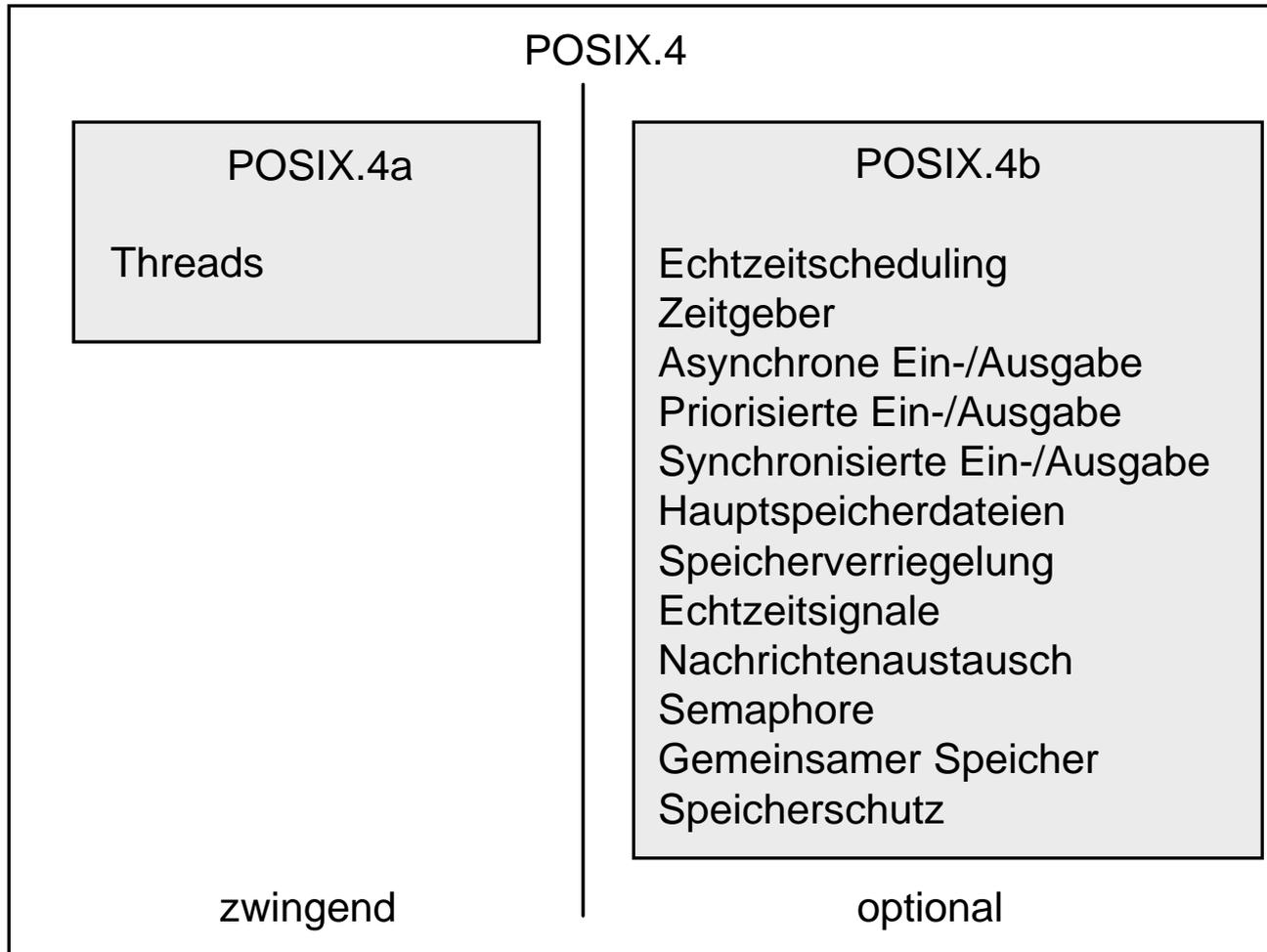
QNX-Architektur



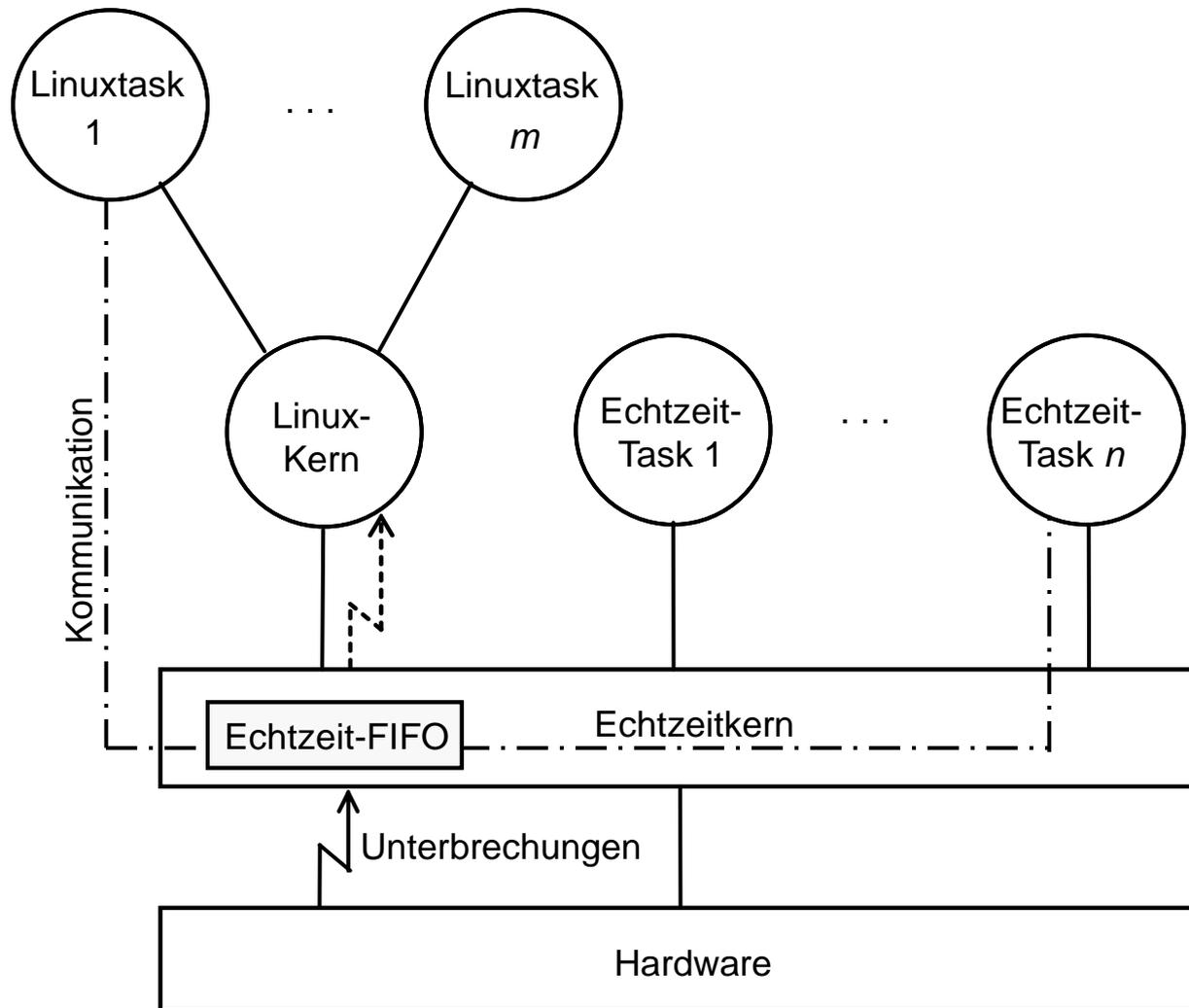
POSIX-Standards für Betriebssysteme

POSIX.1	Grundlegende Betriebssystemschnittstelle
POSIX.2	Kommando-Interpreter
POSIX.3	Test-Methoden
POSIX.4	Echtzeiterweiterungen
POSIX.5	ADA Anbindung an POSIX.1
POSIX.6	Sicherheitserweiterungen
POSIX.7	Systemverwaltung
POSIX.8	Transparenter Dateizugriff
POSIX.9	FORTRAN-77 Anbindung an POSIX.1
POSIX.10	Supercomputer Profile
POSIX.11	Transaktionsverwaltung
POSIX.12	Protokollunabhängige Kommunikation
POSIX.13	Echtzeitprofile
POSIX.14	Multiprozessorprofile
POSIX.15	Supercomputererweiterungen
POSIX.16	Sprachunabhängiges POSIX.1
POSIX.17	Verzeichnis- und Namensdienste
POSIX.18	Grundlegendes POSIX Systemprofil
POSIX.19	FORTRAN-90 Anbindung an POSIX.1
POSIX.20	ADA Anbindung an POSIX.4
POSIX.21	Verteilte Echtzeiterweiterungen

POSIX-Standards



Architektur von RTLinux



Aufteilung der Anwendung in Nicht-Echtzeit und Echtzeittasks

Anwendung

Nicht-Echtzeittasks

Benutzerschnittstelle
Grafik
Datenhaltung,
Datenaufbereitung,
....

Echtzeittasks (Module)

Aufgaben mit
harten
Echtzeitanforderungen

Industrielle Echtzeitbetriebssysteme

Produkt	QNX	POSIX.4	RT-Linux	VxWorks	OS-9	CMX	Windows Emb.
Hersteller	QNX Software Systems Ltd.	Posix	GPL (GNU Public License)	Wind River	Microware	CMX Systems	Microsoft
Typ	DS, EBS	Standard für EBS, BA	BA	EBS	EBS	MEBS	EBS
Zielsystem	IA32	IA32, IA64, PowerPC	IA32, PowerPC, ARM	IA32, PowerPC, 680XX, div. Mikrocontroller	IA32, PowerPC, 680XX	Diverse Mikrocontroller	IA32, PowerPC, MIPS, ARM
Sprachen	C, C++	C, C++, Java, Ada	C, C++, Java	C, C++, Java	C, C++, Java	C	C, C++, Java
Dateisystem	Unix, Windows	Unix	Unix, Windows	Unix, Windows	Windows	-	Windows
GUI	X-Win	X-Win	X-Win	X-Win	X-Win	-	Windows
Netzwerk	TCP/IP	TCP/IP, UDP	TCP/IP, UDP	TCP/IP	TCP/IP	-	TCP/IP
Feldbus	-	-	-	CAN, ProfiBus	CAN, ProfiBus, Interbus S	-	-
Scheduling	FPP, FPN, Timeslice, FIFO	FPP, FPN, Timeslice, Benutzerdef.	FPP, FPN, Timeslice	FPP, FPN, Timeslice	FPP, FPN, Timeslice	FPP, FPN, Timeslice	FPP, FPN
Speicher- verwaltung	Stat., Dyn. reel-, virt. Adr.	-	Stat., Dyn. reel-, virt. Adr.	Stat., Dyn. reel-,virt. Adr.	Statisch reele Adr.	Statisch reele Adr.	Stat., Dyn. reel-,virt. Adr.