

# Ein-/Ausgabe und Disk Scheduling

Peter Puschner

Institut für Technische Informatik

[peter@vmars.tuwien.ac.at](mailto:peter@vmars.tuwien.ac.at)

# Was interessiert uns?

- Charakteristika von I/O-Geräten
- Anforderungen an das I/O-System
- Realisierung von I/O
- BS-Mechanismen
  - Pufferung
  - Disk Scheduling
  - Disk Cache

# Ein-/Ausgabe

“perhaps the messiest aspect of OS design ...”

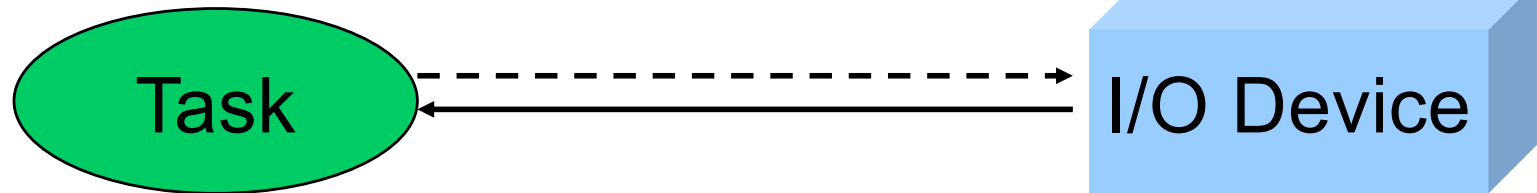
- Vielzahl an Gerätearten und Anwendungen
- 3 Arten von externen I/O-Geräten:
  - Mensch-Interface (z.B.: Anzeige, Tastatur)
  - Maschinen-Interface (elektron. Geräte, z.B.: Datenlaufwerke, Sensoren, Aktuatoren)
  - Kommunikation (für Datenaustausch mit anderen, weiter entfernten Geräten, z.B.: Netzwerk)

# Merkmale von I/O-Geräten

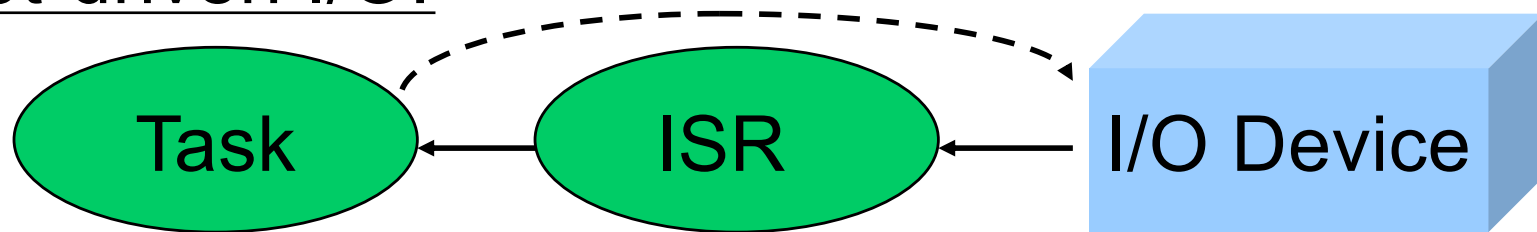
- Datenrate
- Anwendung (z.B.: Dateiverwaltungs-Layer: Files vs. Sekundärspeicher für VMM)
- Ansteuerungs-Komplexität (z.B. Pollen vs. Interrupt)
- Transfer-Einheiten (z.B. Zeichen vs. Blöcke)
- Daten-Repräsentation (z.B. Byte Order, Parität)
- Fehlerbehandlung (Arten, Auftreten)

# I/O-Funktionen: Mechanismen

## Programmed I/O:



## Interrupt-driven I/O:



## DMA:



# I/O-Funktionen

- **Programmed I/O**: Prozess schickt ein I/O-Kommando und wartet mittels “Busy-Waiting”.
- **Interrupt-driven I/O**: Prozess schickt ein I/O-Kommando und BS/SW arbeitet weiter. Wird durch I/O-Modul unterbrochen, wenn I/O fertig.
- **Direct Memory Access (DMA)**: Prozess schickt Kommando an I/O-Modul. Das I/O-Modul kopiert autonom Daten zwischen Speicher und I/O-Modul und unterbricht CPU-Aktivität nach Fertigstellung der I/O-Operation.

# Direct Memory Access (DMA)

- Kann Aktionen der CPU nachbilden, indem DMA Controller die Kontrolle von Bus übernimmt.
- Zwei Arten, wann DMA-Transfer gestartet:
  - blockweiser Transfer (ev. parallel mit CPU-Aktivität)
  - DMA erzwingt einzelne Übertragungszyklen am Bus (“cycle stealing”)
- DMA ist effizient: Kein Kontextwechsel der CPU notwendig
- Flexibel: DMA-Modul kann mehrere I/O-Module versorgen (mehrere Ports oder eigener I/O-Bus).

# I/O Channel

- I/O Channel ist eine Erweiterung zu DMA
- Durch Ausführung von I/O-Befehlen komplette Kontrolle über I/O-Operationen
- I/O-Instruktionen im Hauptspeicher, Abarbeitung durch I/O Prozessor, Interrupt signalisiert Ende
- Zwei typische Arten von I/O Channels:
  - **Selector Channel** kann mehrere Geräte verwalten, wobei immer nur eines aktiv ist.
  - **Multiplexer Channel** kann mehrere Geräte simultan bedienen.



# Kriterien für OS-Design

- Kompromiss zwischen widersprechenden Zielen:
  - **Effizienz**: wichtig, da I/O oft Bottleneck des Systems. Mehr Prozesse bringt ev. Swapping (zusätzl. I/O).  
⇒ spricht für *gerätespezifische Lösungen*
  - **Flexibilität**: für Einfachheit, geringere Fehlermöglichkeiten von I/O-Operationen, Ersetzbarkeit, Austauschbarkeit von Geräten  
⇒ *einheitliche Schnittstellen* erforderlich (zB: Zugriffsfunktionen wie open, close, lock, unlock,...).

# Logische Struktur von I/O

Struktur von I/O-Funktionen für Peripheriegeräte  
(Schichtung gem. Komplexität, Zeitkonstanten, Abstraktion)

Ziel: Verbindung von

*einheitlichem Interface – gerätespezifischer Ansteuerung*

Schichtungsebenen:

- **Logical I/O**: logische Bedienung des Gerätes  
(z.B.: open, close, read,...)
- **Device I/O**: Übersetzung von Operationen in Sequenz von I/O- und Kontroll-Kommandos; Pufferung
- **Scheduling and control**: Queuing, Verwaltung von I/O-Operationen, Interrupt-Handling, Lesen des I/O-Status.

# I/O für Geräte mit File System

- I/O für File System hat die Struktur wie Lokale Peripheriegeräte.
- *Logical I/O* besteht aus drei Komponenten:
  - **Directory Management**:  
Übersetzung: symbol. Dateinamen – Dateireferenzen;  
Directory-Benutzeroperationen (zB: add, delete, ...)
  - **File System**: behandelt logische Struktur von Dateien,  
Dateioperationen (zB: open, close, read, write, ...)
  - **Physical Organisation**: Übersetzung von logischen  
Referenzen in Spuren und Sektoren.

# Blocking vs. Non-Blocking I/O

- Blocking I/O

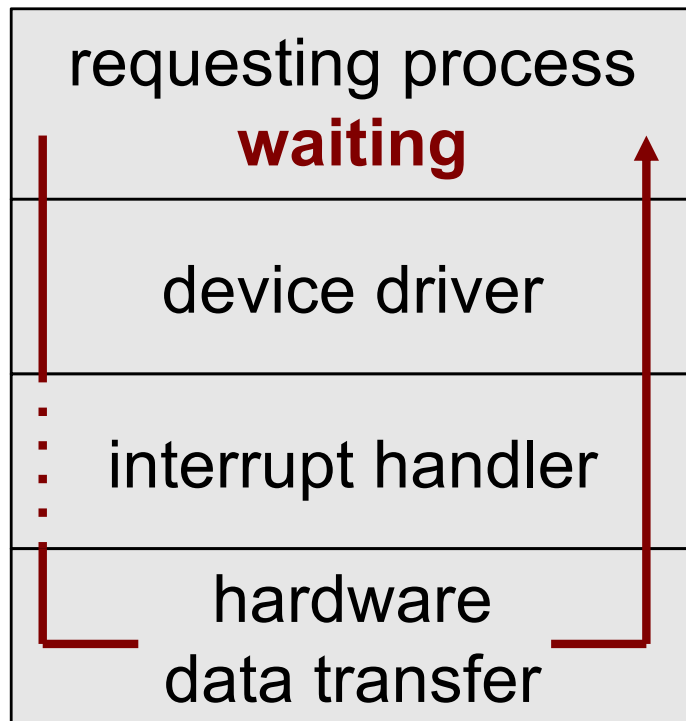
Prozess wird von Zustand *Running* in die *Blocked* Queue gestellt und blockiert.

- Non-Blocking I/O

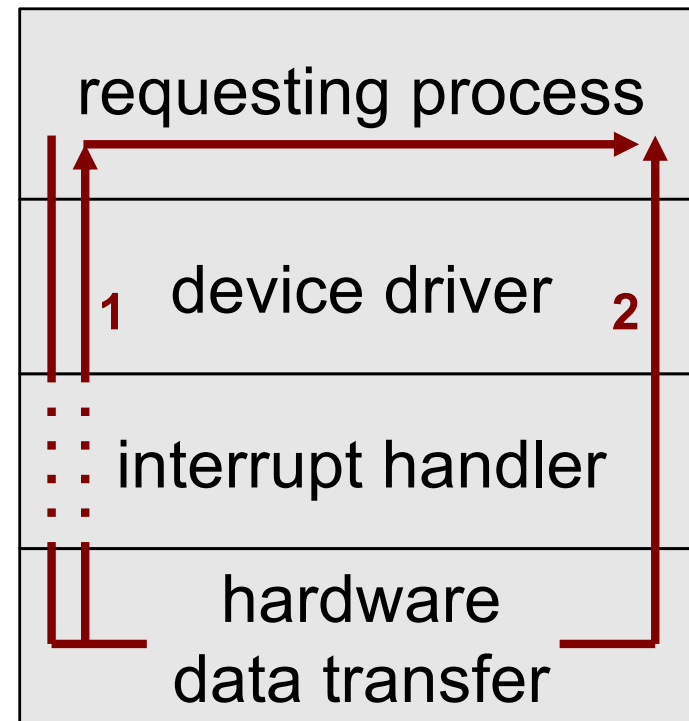
I/O Operation wird sofort fertiggestellt,  
*kein Blockieren*,

Return Status liefert Feedback über Operation  
(z.B. Anzahl der übertragenen Bytes).

# Synchrone vs. Asynchrone I/O

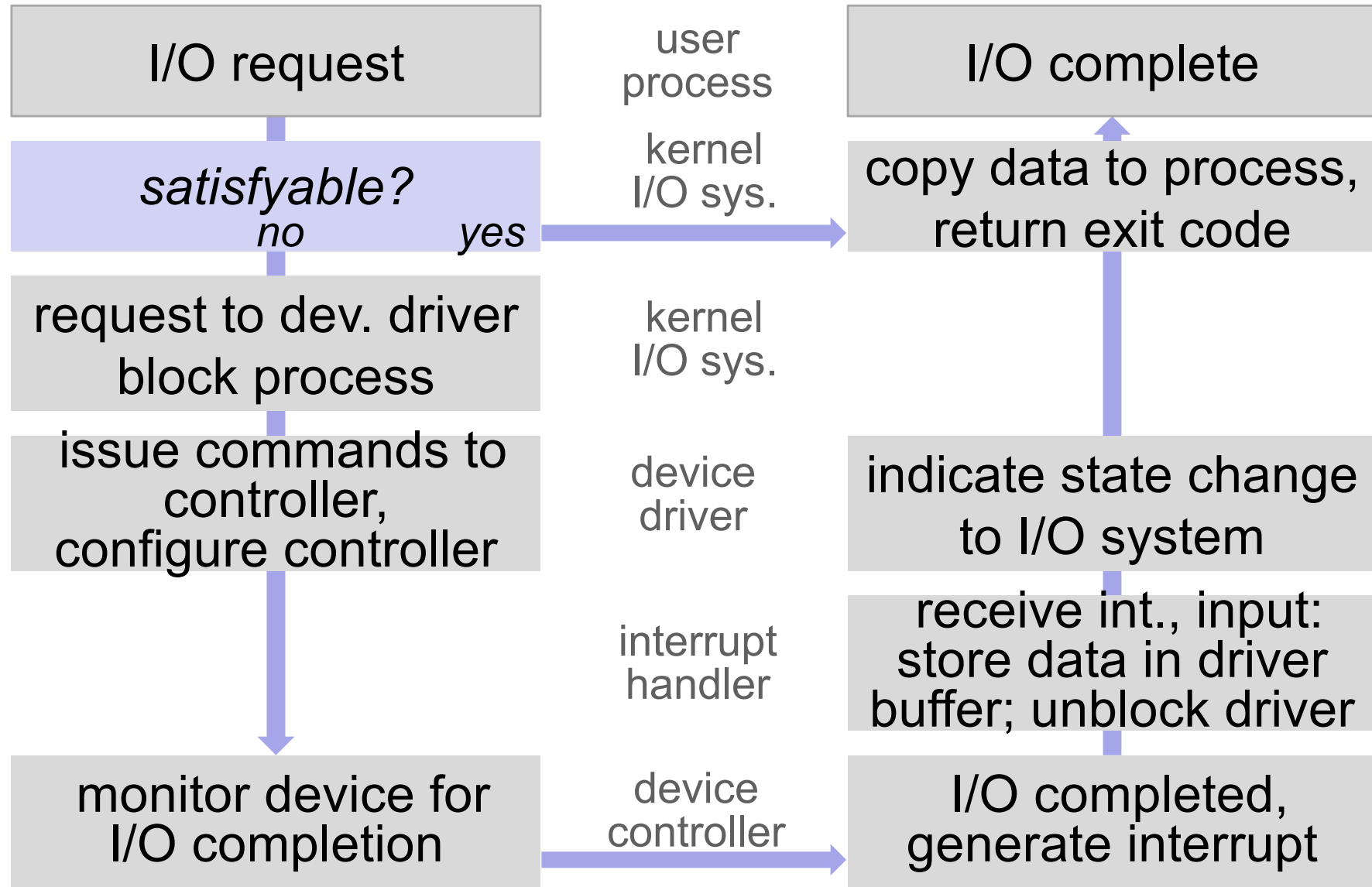


Synchronous I/O  
(blocking or non-blocking)



Asynchronous I/O

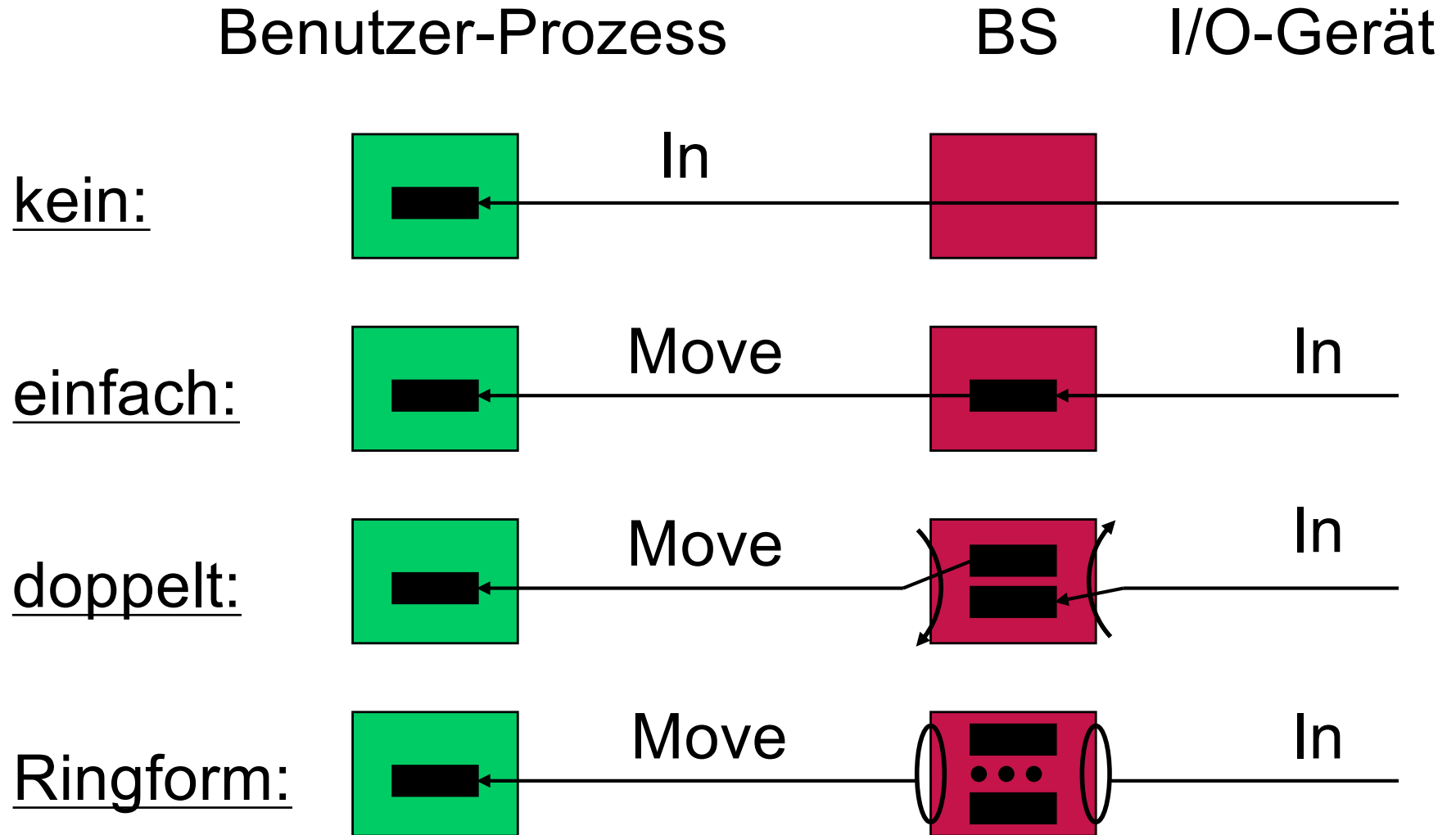
# Synchronous I/O Request



# Puffern von I/O-Anfragen

- Puffer: Zwischenspeicher für Daten bei I/O-Transfer
- **Zusammenfassung** von low-level I/O-Operationen  
Verringerung des Overheads
  - **Entkopplung** Prozess I/O – BS I/O
  - BS: Entkopplung I/O – Swapping
  - **Maskieren** von Geschwindigkeitsunterschieden  
bei Lastspitzen (ev. mehrere Puffer)  
Achtung: Langzeit- I/O-Performance durch Device  
Performance limitiert
  - **Kosten:** Speicher, Puffermanagement

# Puffern von I/O-Anfragen (2)





# Disk I/O – Scheduling

- Seek Time  $T_S$ : benötigte Zeit, um Disk-Arm zur gewünschten Spur zu bewegen (Mittelwert).
  - Rotational Delay  $T_{RD}$ : Zeitverzögerung, bis Anfang des gesuchten Sektors gefunden (im Mittel: halbe Umdrehungszeit der Disk)
  - Transfer Time  $T_{TF}$ : benötigte Zeit zum Übertragen der Daten.
- ⇒ Average Access Time  $T_A$ : benötigte Zeit für Datenzugriff (Mittelwert)

# Charakteristische I/O-Zeiten

b...Anzahl der zu übertragenden Bytes

N...Anzahl der Bytes pro Spur

r...Umdrehungsgeschwindigkeit [U/sec]

$$T_{RD} = \frac{1}{2 \cdot r}$$
$$T_{TF} = \frac{b}{r \cdot N}$$
$$T_A = T_S + T_{RD} + T_{TF}$$

Wichtig: Organisation der Daten auf der Platte!!!

# Strategien des Disk Scheduling

Ziel: Ordnung von Disk-I/O Requests zur Minimierung der Seek Time ( $T_s$ )

- Priority
- First-In, First-Out (FIFO)
- Last-In, First-Out (LIFO)
- Shortest Service Time First (SSTF)
- SCAN (Elevator Alg.)
- C-Scan
- N-step-SCAN und FSCAN

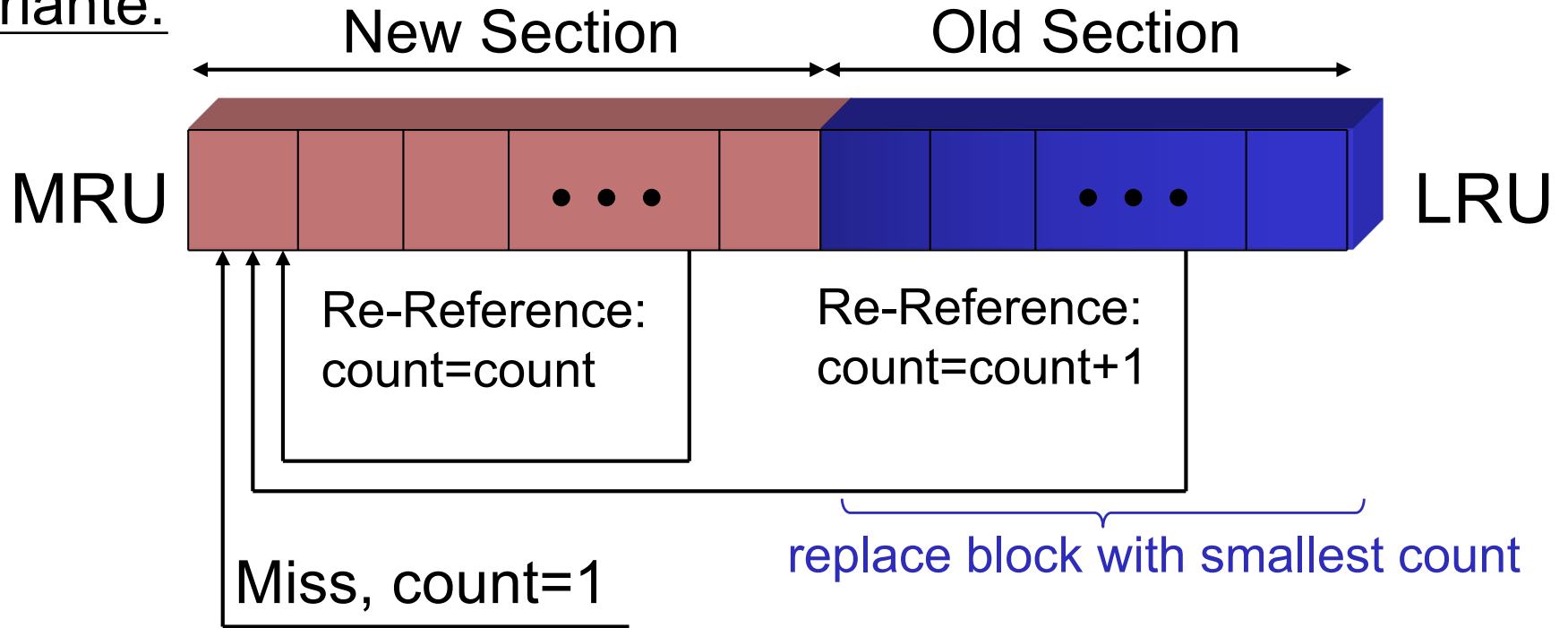
RAID (Redundant Array of Independent Disks):  
Beschleunigung und Fehlertoleranz

# Disk Cache

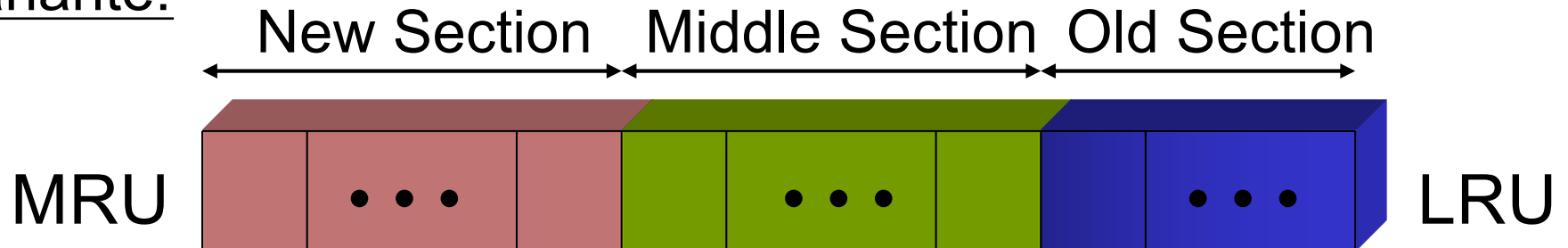
- Ähnliches Prinzip wie Memory Cache
- Puffer im Hauptspeicher für Sektoren einer Disk
- Beruht auf Lokalität der Daten
- Austauschstrategien:
  - Least Recently Used (LRU)
  - Least Frequently Used (LFU)
  - Frequency-Based Replacement  
(Verbesserung von LFU)
- Austausch im Cache bei Bedarf oder im Voraus

# Frequency-Based Replacement

## 1. Variante:



## 2. Variante:



# Zusammenfassung

- I/O ist ein Service des BS zum Zugriff auf externe Geräte oder Ports des Prozessors.
- Design von I/O ist Kompromiss zwischen Effizienz und Flexibilität
  - Schichtung der SW des I/O Systems
- Puffern erhöht Performance und Flexibilität
- Disk Scheduling und Caching zur Erhöhung der durchschnittlichen Zugriffsgeschwindigkeit