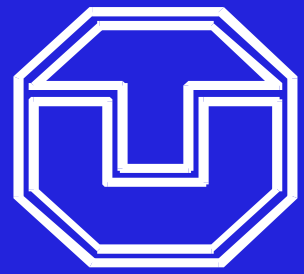
Einführung und Bausteine

Betriebssysteme und Sicherheit

Hermann Härtig
TU Dresden



Betriebssysteme: Vorlesung

Web-Seite

http://os.inf.tu-dresden.de/Studium/Bs

- **Mailingliste !!!**

Folien kurz vor Vorlesungstermin auf der Web-Seite

Weitere Lehrveranstaltungen (ab 4. Sem.)

Sommersemester:

- Proseminar
- Distributed Operating Systems
- Microkernel Construction
- (nicht mehr: Quantitative Methoden der BS-Konstruktion)

Wintersemester:

- Real-Time Systems
- Microkernel-Based Operating Systems
- Praktika (Komplex und Pflicht)

➔ **WS+SS:** EZAG (Forschungsseminar)
➔ **WS+SS:** Hauptseminar
➔ **WS+SS:** Reading Group

- **Ca 2. Woche vor WS:** Advanced Systems Programming (Blockveranstaltung)

Literatur

Modern Operating Systems
Andrew S. Tanenbaum
Prentice Hall

Distributed Systems, Concepts and Design, 4rd edition
Coulouris, Dollimore, Kindberg
Addison Wesley

Publikationen

Betriebssystem - Begriffsbestimmung

- Summe derjenigen Programme, die als residenter Teil einer EDV-Anlage für den Betrieb der Anlage und für die Ausführung der Anwenderprogramme erforderlich ist.
  (Lexikon der Informatik, 1991)
- das wichtigste Systemprogramm, es kontrolliert die Ressourcen des Rechners und ist die Basis für die Entwicklung der Anwendungsprogramme
  (TANENBAUM)
- Das Betriebssystem wird gebildet durch die 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 Programmen steuern und überwachen. *(DIN 44300)*

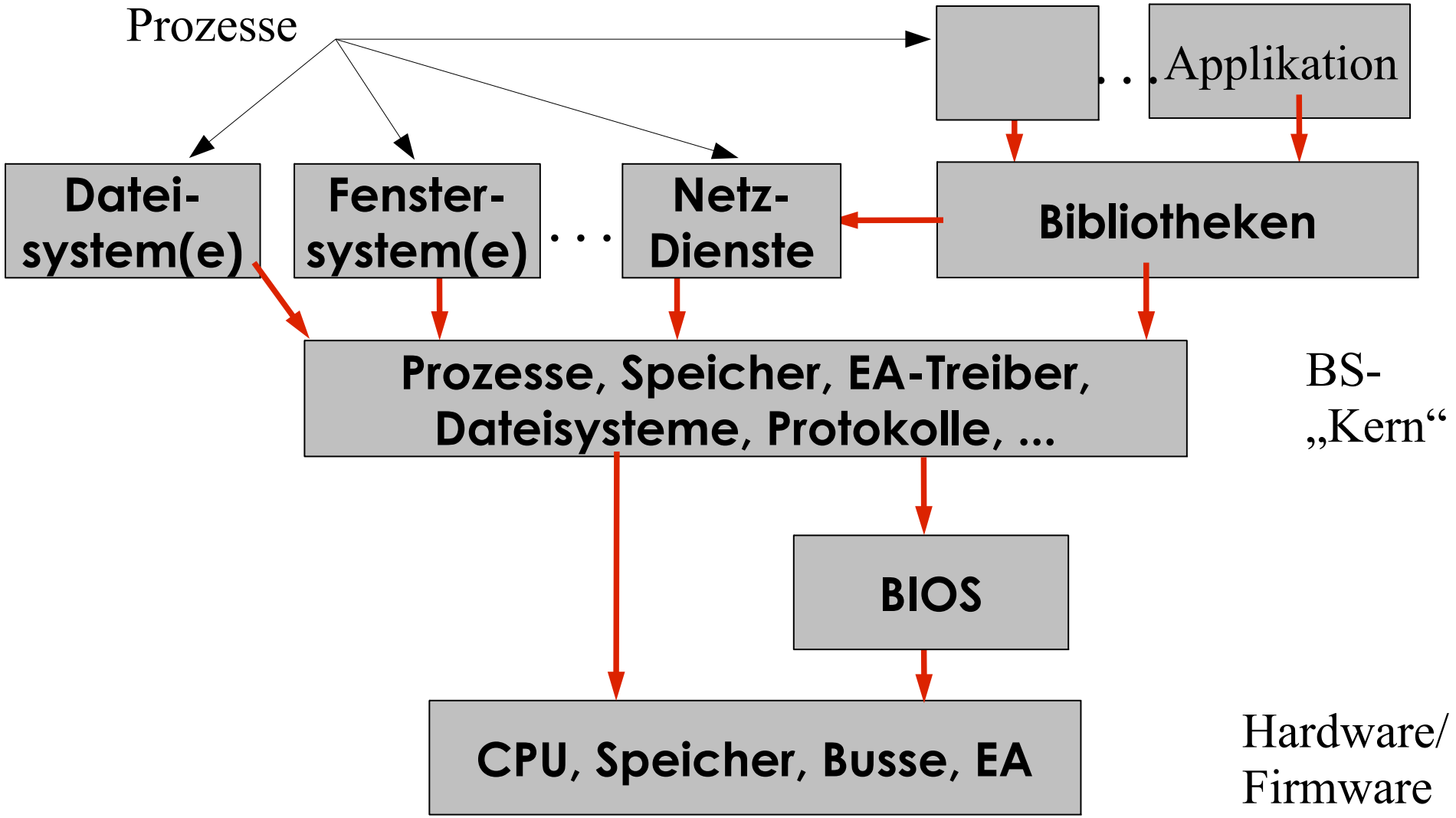
Betriebssystem - Begriffsbestimmung

- Betriebssystem ist ein Programm, das als Zwischenmedium zwischen Nutzer und Hardware tritt.
  Zweck: Umgebung zur Programmausführung schaffen
  Ziel: bequeme Nutzung, effizientes Betreiben
  (SILBERSCHATZ)

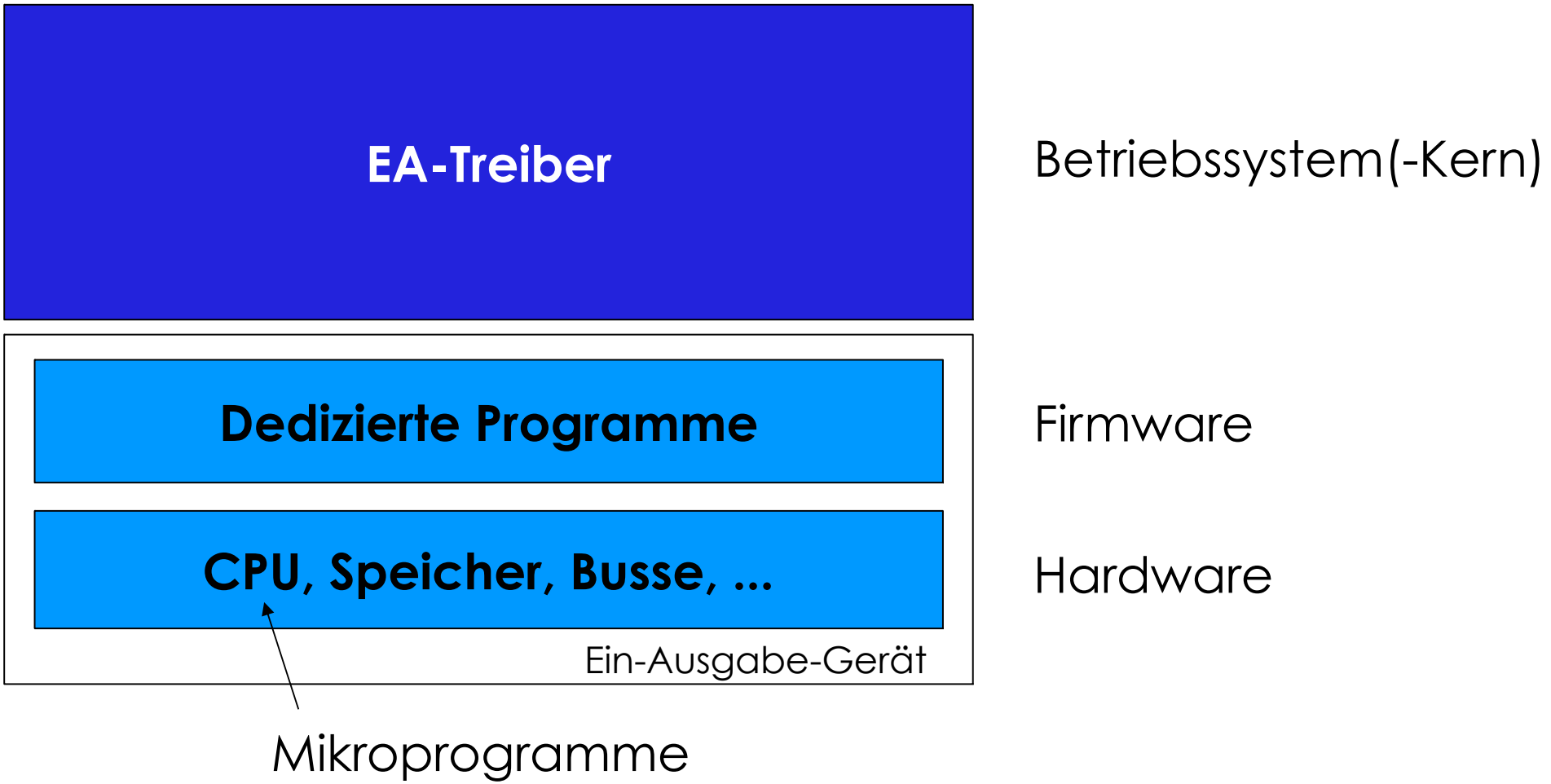
- Komponente eines Rechensystems zur Steuerung von Prozessen und zur Verwaltung von Betriebsmitteln mit dem Ziel, eine einfache Nutzung und einen effizienten Betrieb zu ermöglichen.

- Ein Bild sagt mehr als … (nächste Folie)

System-Struktur



System-Struktur: Ein-/Ausgabe



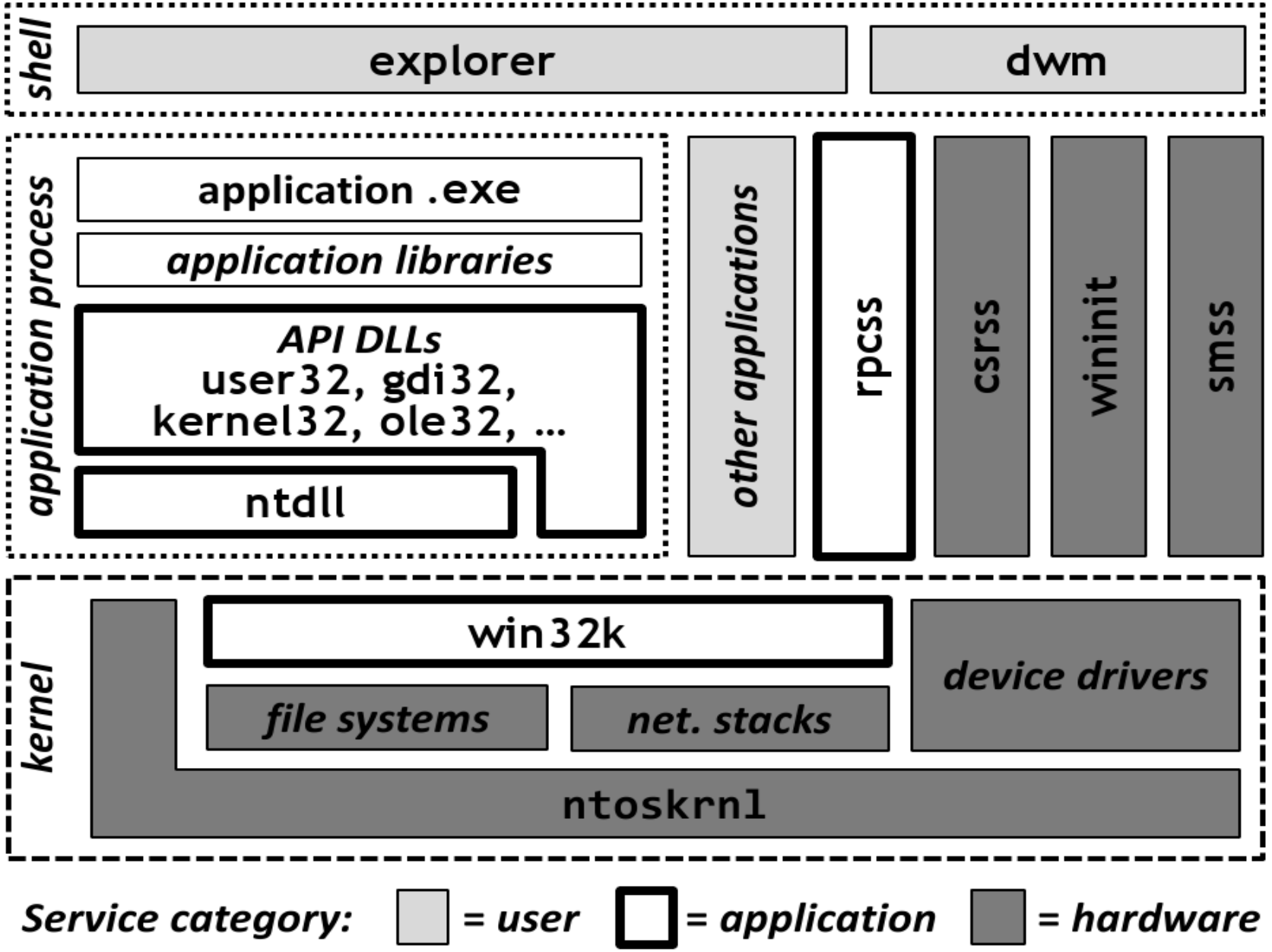


Figure 1. Windows 7 OS Architecture.

(Source: Rethinking the Library OS from the Top Down, ASPLOS '11)

Aufgaben eines Betriebssystems

- Betriebsmittelverwaltung
- Persistente Datenhaltung (z. B. Dateien)
- Kommunikation/Vernetzung
- Ein-/Ausgabe
- Schutz
- Umgang mit Fehlfunktionen (HW-Ausfälle, SW-Fehler, ...)
- Virtualisierung
- Messen, Abrechnen („Accounting“)
- Benutzerfreundliche Schnittstelle zur Maschine

Klassifikationskriterien für Betriebssysteme

- Aufgabenbereich/Betriebsart
  - Stapelverarbeitung („Batch“): HPC
  - Interaktiver Betrieb (Notebook, Desktop, Tablet, Handy)
  - Betrieb als Server
  - Echtzeit-Aufgaben
- Architektur (mikrokern – monolithisch, ...)
- Einsatzbreite
- Nutzeranzahl und Nebenläufigkeit
- Verteilung
- Prozessor-Anzahl

Bewertungskriterien für Betriebssysteme

- Effizienz: CPU, Hauptspeicher, Platte, Energie
- Funktionsumfang
- Einfachheit
- Wartbarkeit/Erweiterbarkeit
- Schutzmöglichkeiten gegen Angriffe/Fehler
- Echtzeitfähigkeit
- Kompatibilität

Lernziele dieser Vorlesung

- Grundlagen der Systemarchitektur
- Vermittlung einer integrierten Sicht auf Systeme (Hardware, kryptographische Verfahren, Mathematik, maschinennahes Programmieren, ...)
- Betriebssysteme
- Umgang mit parallelen Prozessen
- Umgang mit Betriebsmitteln
- Abwägungen (Trade Off)

Wegweiser durch die Vorlesung

- Einleitung und grundlegende Bausteine
  Unix als Fallbeispiel

- Prozesse/Threads/Kommunikation
  Speicher
  Dateien
  Ein/Ausgabe
- Sicherheit / Kryptoverfahren
  Fehlertoleranz
  Echtzeit
  Verklemmungen
  Quantitative Methoden
  Verteilung

- Forschungsthemen (L4, ...)

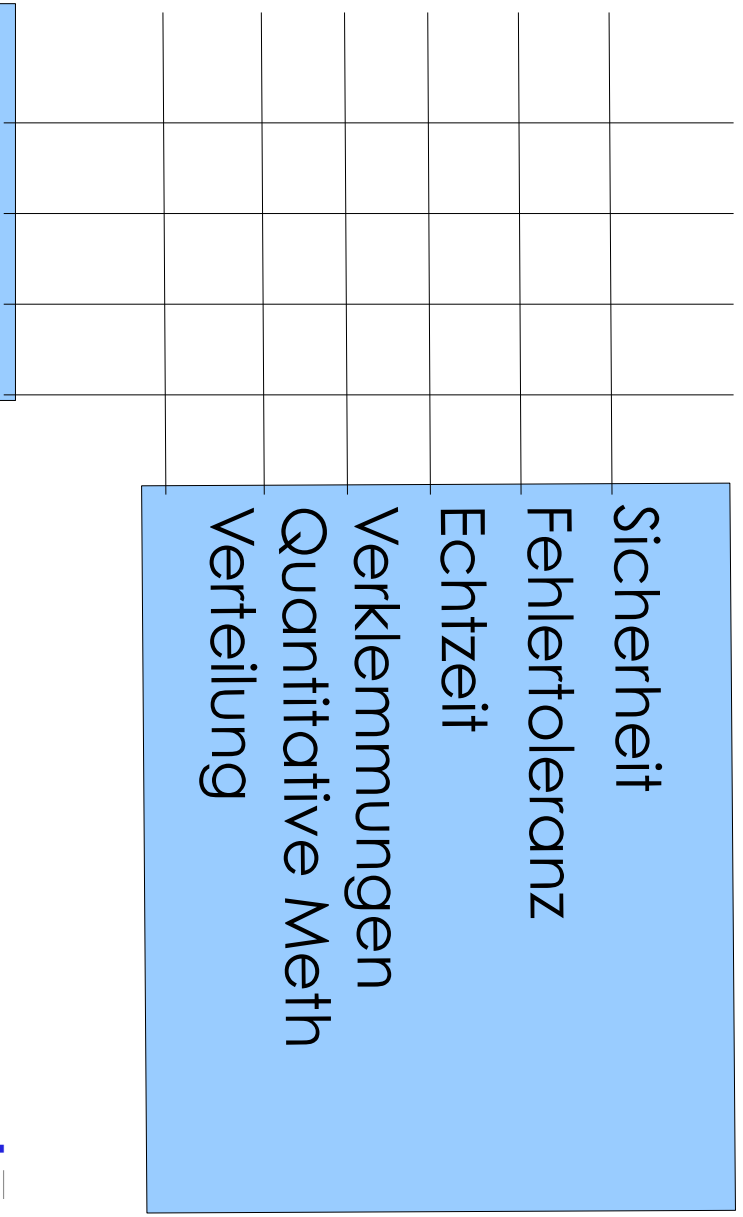
Wegweiser durch die Vorlesung

Prozesse/Threads/Kommunikation

Speicher

Dateien

Ein/Ausgabe



Die wichtigsten Bausteine von BS

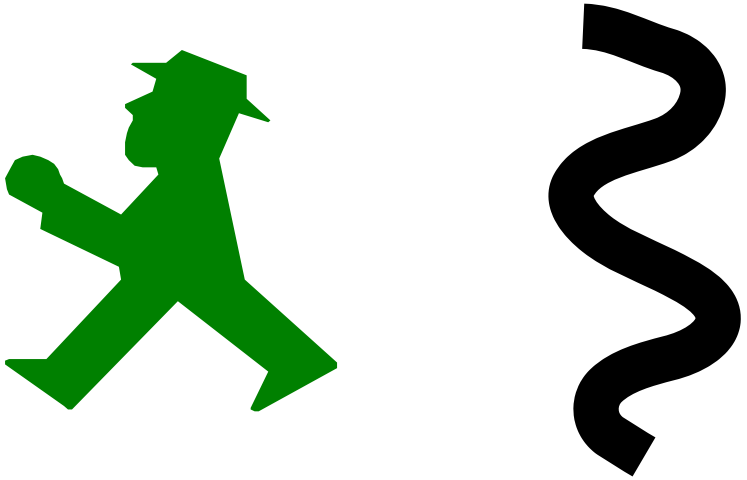
- Threads
- Adressräume
- Prozesse und Kommunikation
- Dateien
- Betriebssystem-“Kern“
- E/A-Treiber

Definition: Thread

Eine selbständige

- ein sequentielles Programm ausführende
- zu anderen Threads parallel arbeitende
- von einem Betriebssystem zur Verfügung gestellte

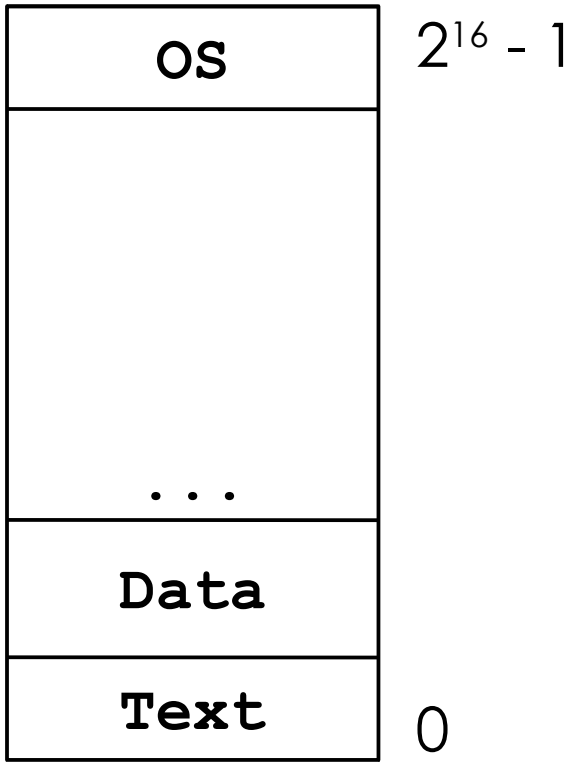
Aktivität.



Definition: Adressraum

- Menge aller Daten, Instruktionen. .., auf die „direkt“, d. h. mittels **load/store**/... HW-Instruktionen und über den PC zugegriffen werden kann
- z.B.: primitive CPU, d.h. Zugriff nur über ein Adressregister A (16 Bit)

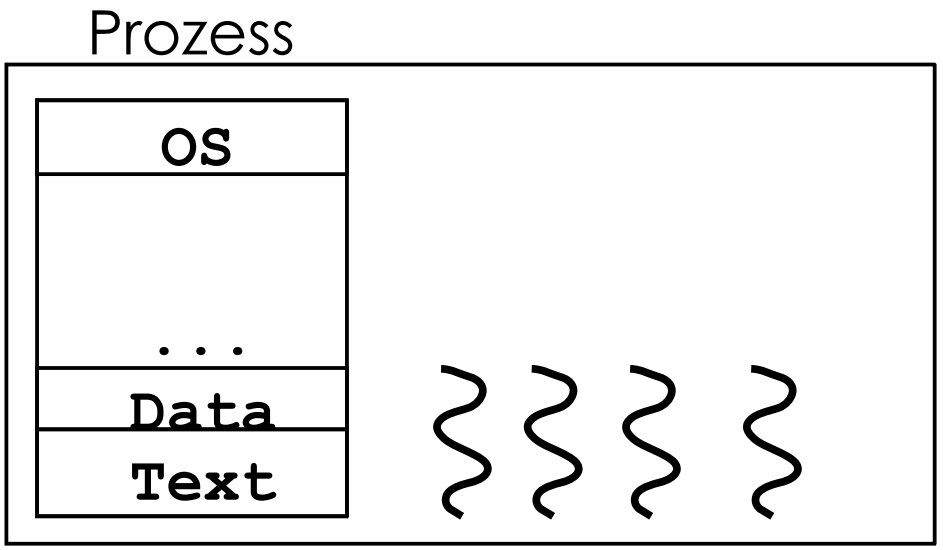
```
ld R, A
//lade den Wert, welcher
//im Speicher an der
//Adresse steht, die im
//Register A angegeben ist,
//ins Register R

st R, A
//analog schreibend
```



Definition: Prozess

Eine Einheit aus

- einem Adressraum und
- mindestens einem Thread



Weitere Eigenschaften

- Einheit, der Betriebssmittel zugeordnet werden
- Repräsentant von Benutzern innerhalb des Betriebssystems

Begriffsvielfalt in Lehrbüchern, aber auch in ...

... Betriebssystemen und Programmiersprachen

- MACHs Task ist ein „multithreaded“ Prozess
- ADAs Task ist ein Thread

Weiterhin:

- „Thread“ manchmal nur im Kontext von „multithreaded“ Prozessen
- „User Level Threads“ vs. Threads, die von einem Betriebssystem zur Verfügung gestellt oder unterstützt werden
- Hardware-Architektur Kontext: mehrere eigenständig arbeitende Instruktionsströme (Threads) nutzen CPU „Hyperthreading“

➔ **Wichtig**:
in jedem Fall zunächst Verwendung der Begriffe klären !!!

Gegenüberstellung: Prozeduren und Threads

Prozeduren P,Q

```
P;Q;
```

- Q beginnt, wenn P beendet
- P und Q werden von einem Thread ausgeführt

Threads P, Q

```
P||Q
```

- keinerlei Aussage zur Ausführungsreihenfolge der Instruktionen von P und Q
- P,Q – wenn einmal gestartet – sind selbständig

Beispiele für den Einsatz von Prozessen

- Mehrere Benutzer gleichzeitig auf einem Rechner
  - jeder Benutzer wird durch mehrere Prozesse repräsentiert
- Explizite Parallelarbeit eines Benutzers, z. B.
  - Übersetzung im Hintergrund
  - paralleles “make”
- Bereitstellung von Diensten durch Hintergrundprozesse
- Auslagerung einer schützenswerten Unteraufgabe in eigenen Prozess

Beispiele für den Einsatz von Threads

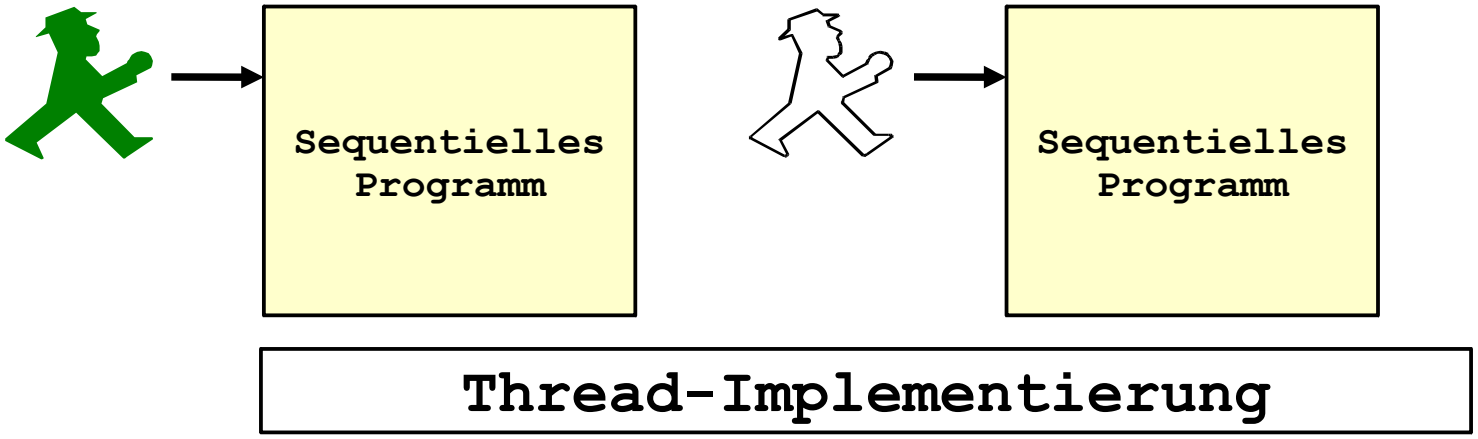
- Mehrere Benutzer (das können andere Prozesse sein, sog. *Klienten* eines Prozesses, z. B. eines Dateisystems)
  - jeder Klient wird durch einen Thread repräsentiert
- Explizite Parallelarbeit eines Benutzers, z. B.
  - parallele Bearbeitung von Matrizen
  - parallele kombinatorische Suche
- Umgang mit Asynchronität, z. B.
  - Tastatureingabe
  - Unterbrechungsbearbeitung
  - Ein-/Ausgabe
- Hilfsmittel zur Strukturierung komplexer Programme

Mehrere Threads auf einem Rechner

Implementierung

- Nur ein Thread kann zu einem Zeitpunkt die CPU besitzen
- Jeder Thread erhält hin und wieder die CPU
  ⟹ „Threadumschaltung“

Struktur:

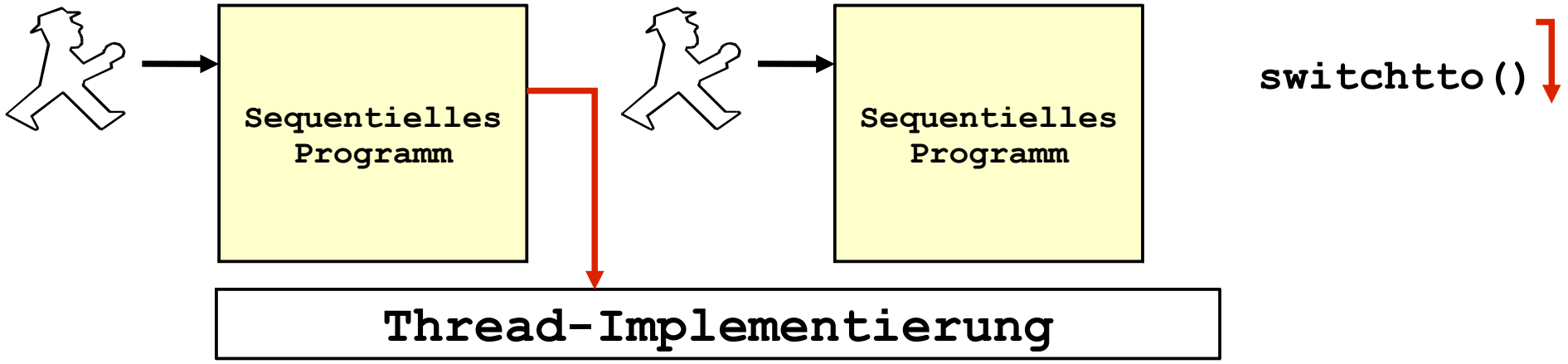


Mehrere Threads auf einem Rechner

Implementierung

- Nur ein Thread kann zu einem Zeitpunkt die CPU besitzen
- Jeder Thread erhält hin und wieder die CPU
  ⟹ „Threadumschaltung“

Struktur:

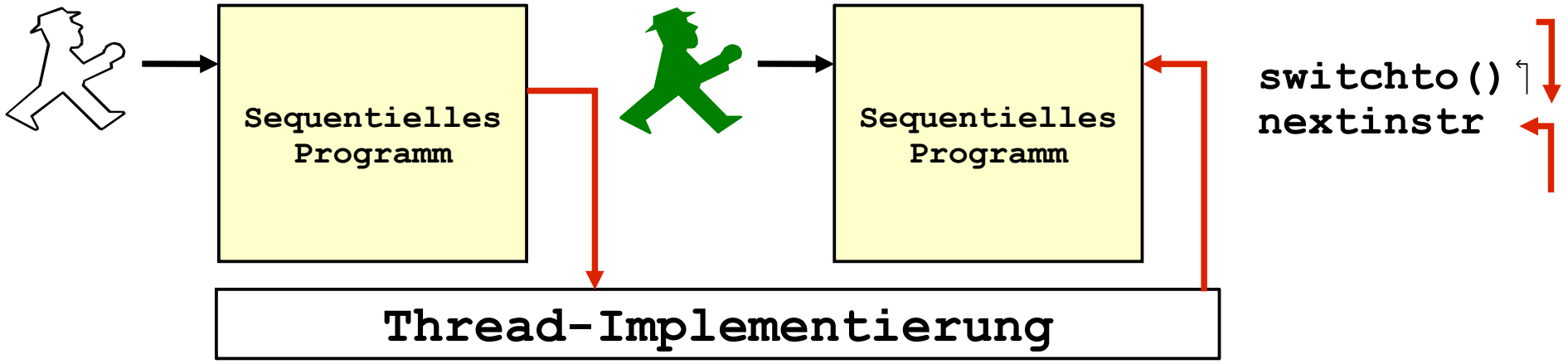


Mehrere Threads auf einem Rechner

Implementierung

- Nur ein Thread kann zu einem Zeitpunkt die CPU besitzen
- Jeder Thread erhält hin und wieder die CPU
  ⟹ „Threadumschaltung“

Struktur:

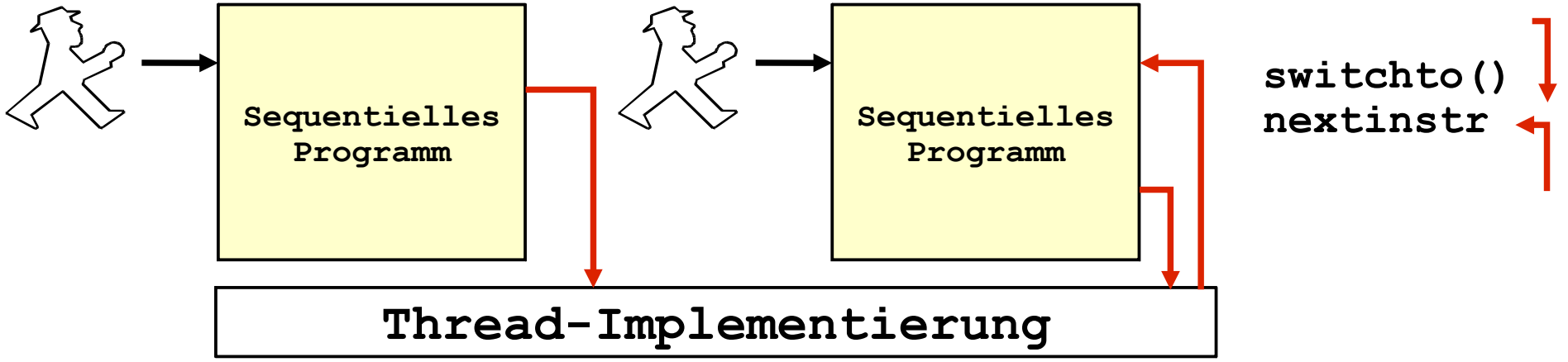


Mehrere Threads auf einem Rechner

Implementierung

- Nur ein Thread kann zu einem Zeitpunkt die CPU besitzen
- Jeder Thread erhält hin und wieder die CPU
  ⟹ „Threadumschaltung“

Struktur:

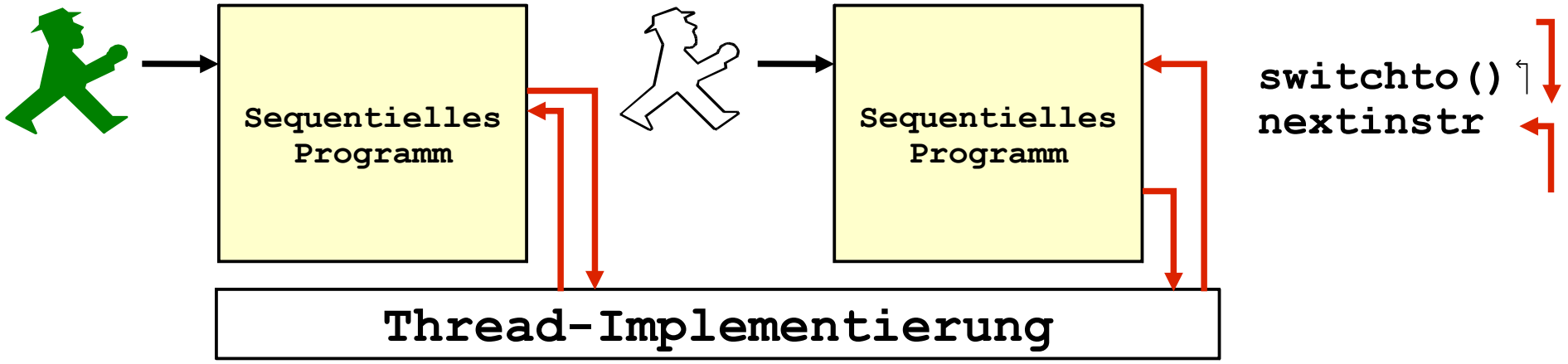


Mehrere Threads auf einem Rechner

Implementierung

- Nur ein Thread kann zu einem Zeitpunkt die CPU besitzen
- Jeder Thread erhält hin und wieder die CPU
  ⟹ „Threadumschaltung“

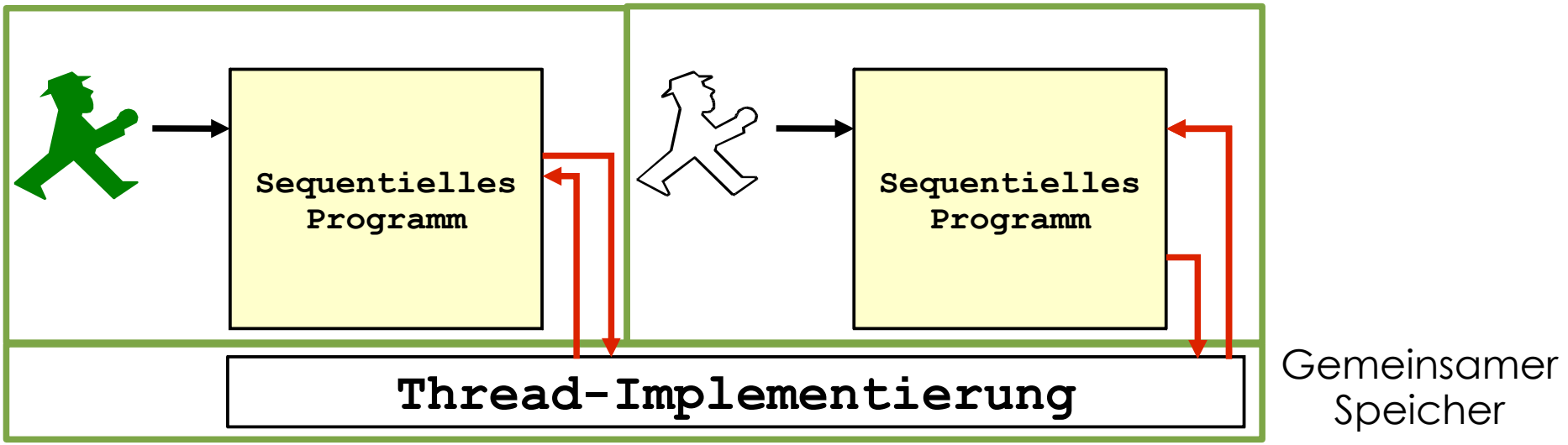
Struktur:



Rechner und Adressräume

- Einfachste Variante:
  - nur ein Adressraum pro Rechner (kleine Systeme)

- Mehrere Adressräume pro Rechner:
  - bei Umschaltung zwischen Threads mit verschiedenen Adressräumen müssen auch diese umgeschaltet werden
  - benötigt HW-Unterstützung (Memory Management Units)



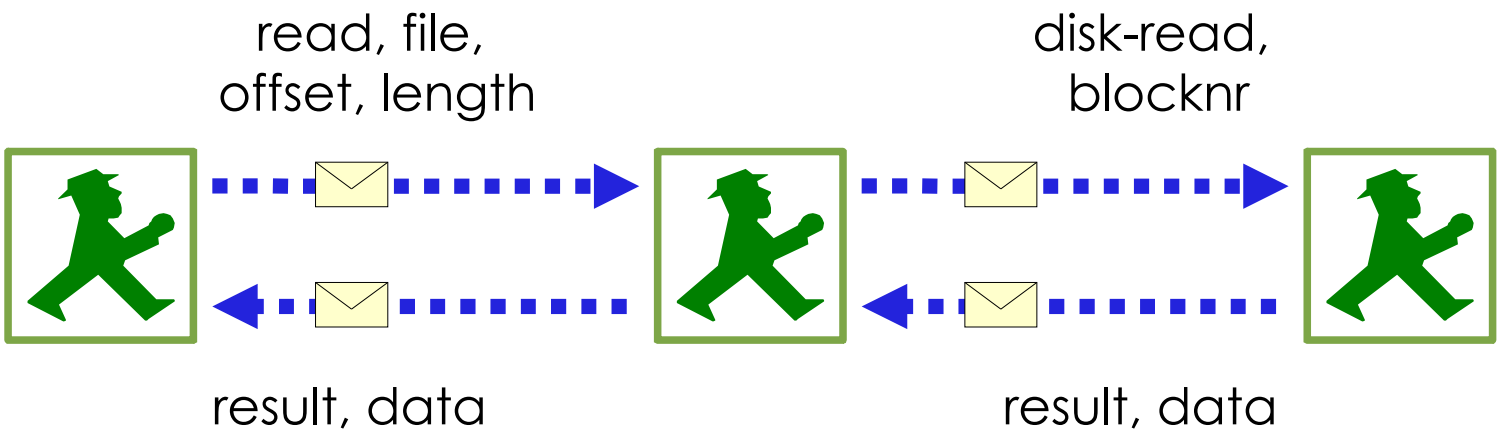
Wozu separate Adressräume ???

- größere Adressräume für Anwenderprogramme (virtueller Speicher)
- Anwenderprogramme sehen bei jeder Ausführung die gleichen Adressen (z.B. 2^8 ... 2^{16}-1)
- Schutz !!! der Daten unterschiedlicher Adressräume voreinander

Dateien

- Behälter für persistente Speicherung von Daten
- Persistentes Speichermedium (Disk, Flash)
- Operationen:
  READ, Write, …
  OPEN, CLOSE,
  SYNC (warum ?)
- Implementierung als Prozess

Prozesse und Dateien



Definition: Betriebssystemkern

Teil des Betriebssystems, der

- von allen anderen genutzt wird und
- im privilegierten Modus des Prozessors („kernel-mode“)
- in einem allen Adressräumen gemeinsamen Speicher

abläuft.

Enthält mindestens („Mikrokerne“):

- Thread- und Adressraum-Implementierung
- elementare Unterbrechungsbehandlung
- Kommunikationsprimitive

Meistens auch („monolithische“ Systeme):

- Ein-/Ausgabe-Treiber, Protokolle
- Speicherverwaltung, Dateisysteme, ...