

# Processes & Threads

Alessio Dal Cero

# Agenda

- I. Was sind Prozesse und Threads?
  - I. Single Thread VS Multi Thread
  - II. Prozessverteilung und Ressourcenmanagement
  - III. Parallelität und Nebenläufigkeit
- II. Koordination und Synchronisation
  - I. Interprozesskommunikation
  - II. Mutual Exclusion
- III. Verteilte Prozesse
  - I. Aspekte der Implementierung
  - II. Vorteile

# Was sind Prozesse & Threads?

# Prozesse & Threads

## Was sind Prozesse & Threads?

### Prozesse:

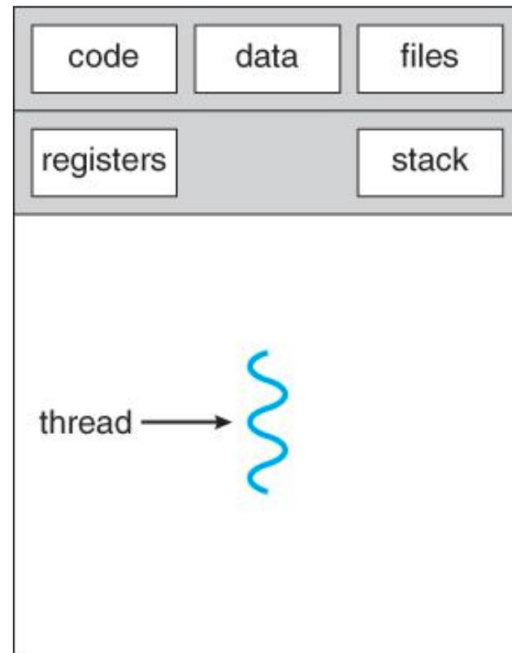
- Ein Prozess ist eine Instanz eines laufenden Programms.
- Prozesse sind unabhängige Einheiten.
- Jeder Prozess hat seinen eigenen Adressraum.
- Prozesse sind voneinander isoliert.
- Jeder Prozess hat mindestens einen Thread, der als Hauptthread bezeichnet wird.

### Threads:

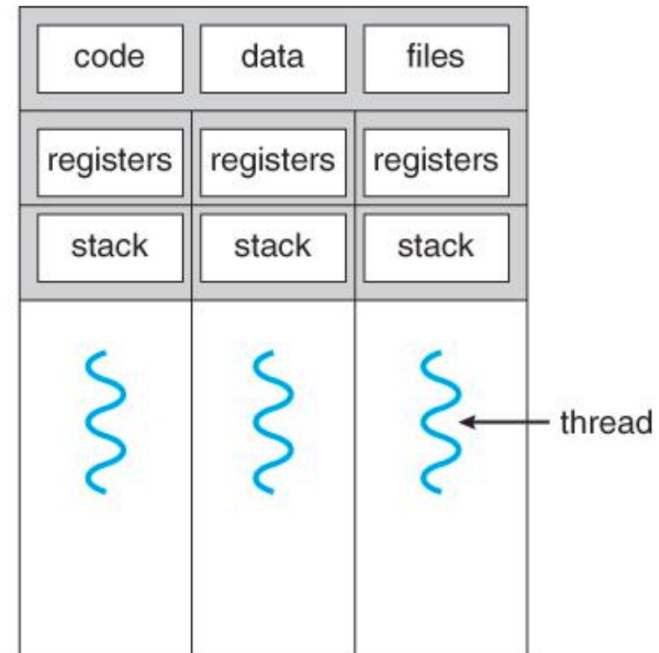
- Ein Thread ist ein ausführbarer Teil eines Prozesses.
- Threads innerhalb desselben Prozesses teilen sich den gleichen Adressraum.
- Threads teilen sich Ressourcen.
- Threads können Aufgaben unabhängig von einander ausführen.

# Prozesse & Threads

## Single Thread VS Multi Thread



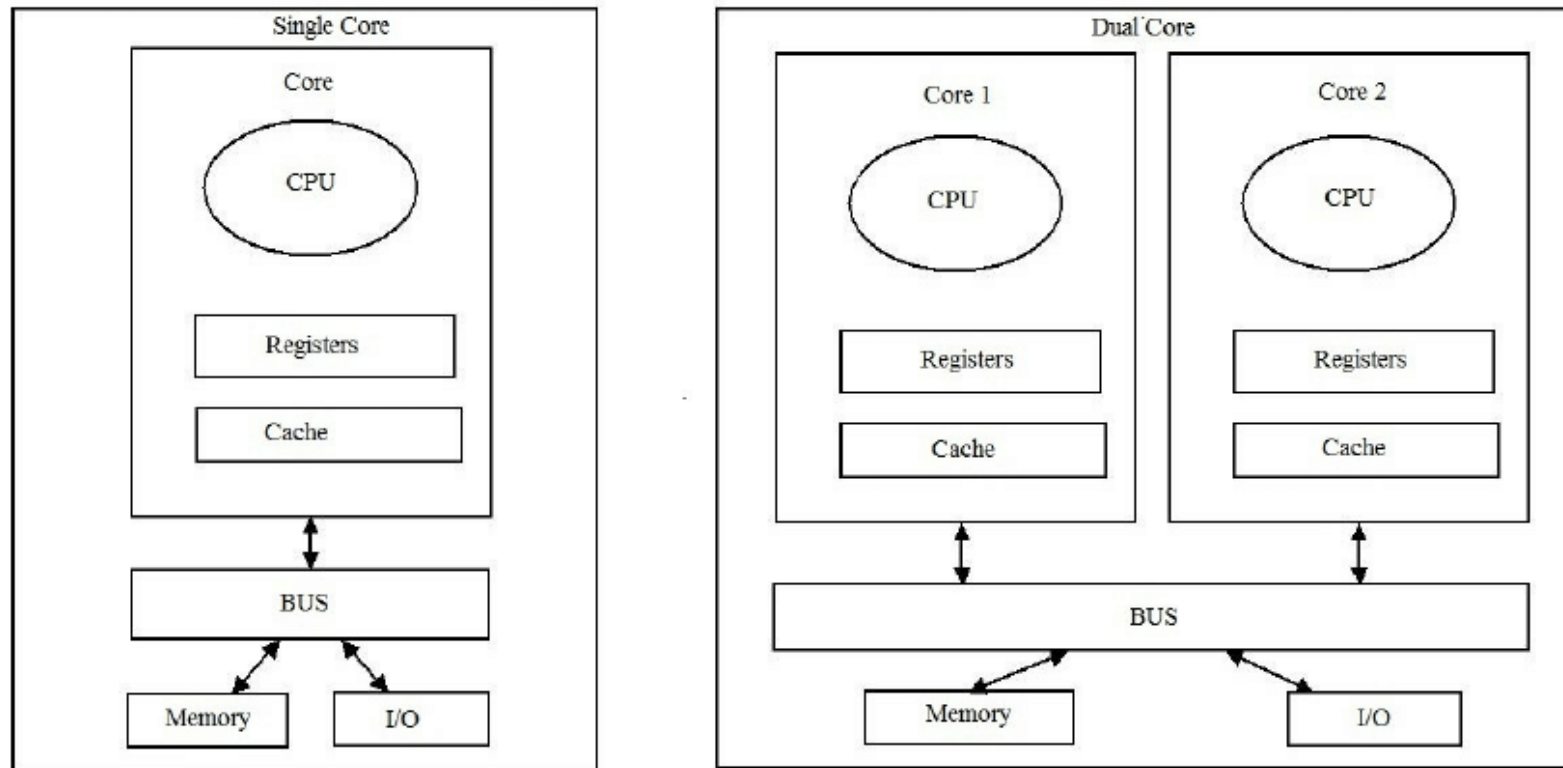
single-threaded process



multithreaded process

# Prozesse & Threads

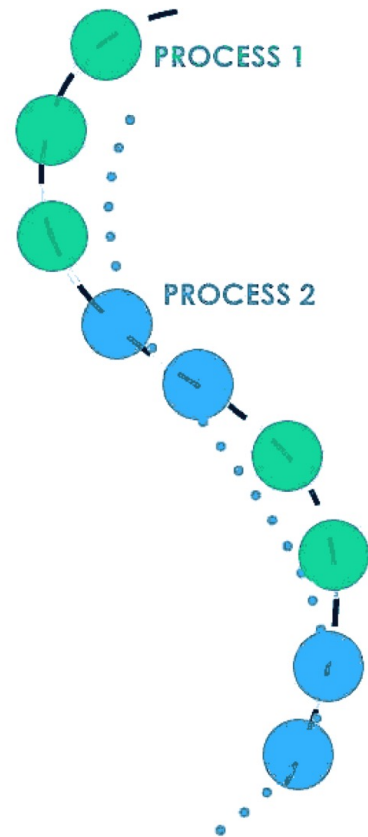
## Prozessverteilung und Ressourcenmanagement



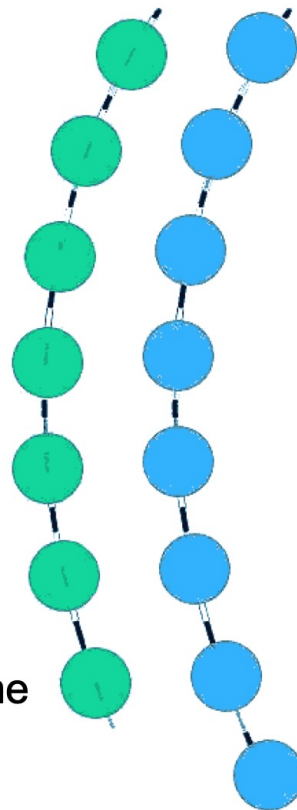
# Prozesse und Threads

## Parallelität und Nebenläufigkeit

### CONCURRENCY

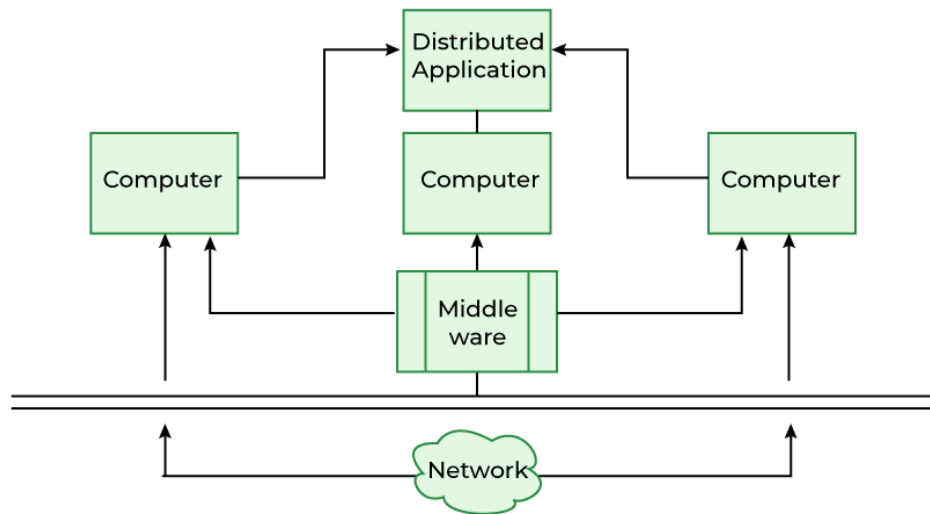


### PARALLELISM



# Prozesse und Threads

## Parallelität und Nebenläufigkeit



- Multitasking
- Reaktionsfähigkeit
- Ressourcennutzung
- Aufgabenparallelität



# Koordination und Synchronisation

# Koordination und Synchronisation

## Interprozesskommunikation

### Interprozesskommunikation (IPC)

- Pipes
- Named Pipes

### Nachrichtenaustausch

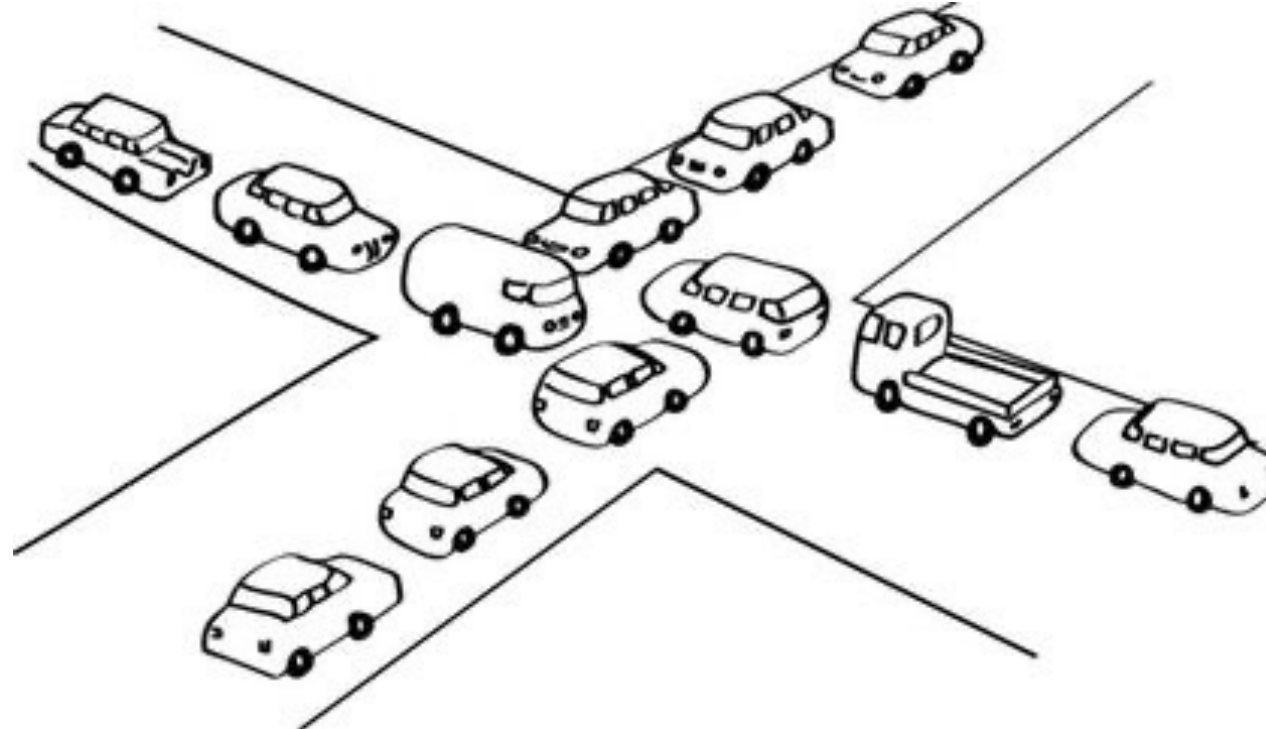
- Sockets
  - Client-Server-Modell
  - Peer-to-Peer-Modell

### Remote Procedure Call (RPC)

### Publish/Subscribe

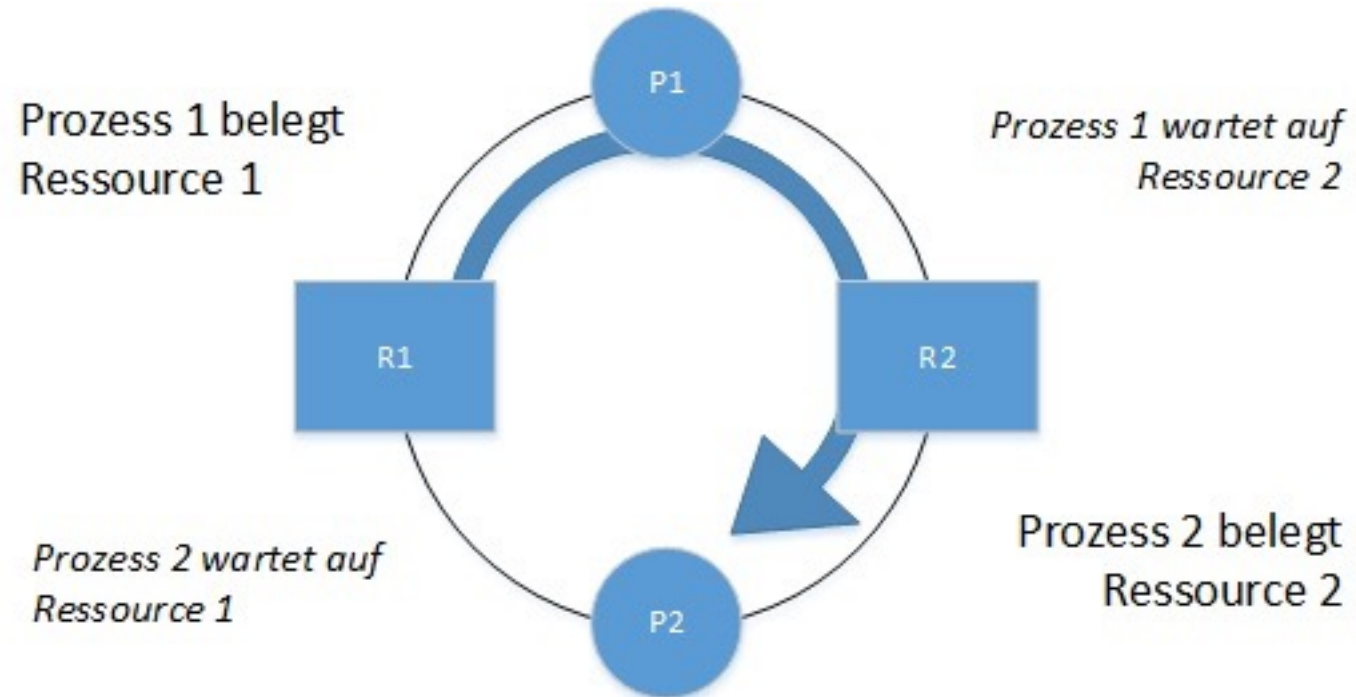
# Koordination und Synchronisation

## Synchronisation



# Koordination und Synchronisation

## Synchronisation



# Koordination und Synchronisation

## Synchronisation

- Mutual Exclusion
  - Semaphore
  - Mutex-Lock
  - Monitore
  - Condition Variables
- Verteilte Algorithmen
  - Lamport-Zeitstempelprotokoll
  - Distributed Mutual Exclusion
    - Ricart-Agrawala
    - Maekawa

# Verteilte Prozesse & Threads

# Verteilte Prozesse & Threads

## Aspekte der Implementierung

- Netzwerkkommunikation
- Protokolle und Nachrichtenformat
- Koordination und Synchronisation
- Fehlertoleranz
- Skalierbarkeit
- Sicherheit

# Verteilte Prozesse & Threads

## Vorteile

- Skalierbarkeit
- Ausfallsicherheit
- Ressourcennutzung
- Lastausgleich
- Geografische Verteilung
- Flexibilität



**Vielen Dank**