

# Systemnahe Programmierung in C (SPiC)

## 27 Programme und Prozesse

**Jürgen Kleinöder, Daniel Lohmann, Volkmar Sieh**

Lehrstuhl für Informatik 4  
Verteilte Systeme und Betriebssysteme

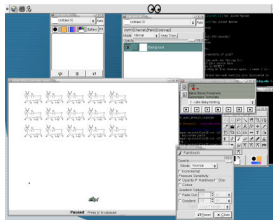
Friedrich-Alexander-Universität  
Erlangen-Nürnberg

Sommersemester 2020

[http://www4.cs.fau.de/Lehre/SS20/V\\_SPiC](http://www4.cs.fau.de/Lehre/SS20/V_SPiC)



- **Mehrere** Programme, die
- **nebenläufig**,
- **dynamisch** gestartet/beendet
- über **definierte E/A-Funktionen**
- ihre Umgebung steuern.



Quelle: [www.wikipedia.org](http://www.wikipedia.org)

Jedes laufende Programm bekommt Hardware zugeteilt:

- CPU (Zeitanteile)
  - Speicher (Teil des Gesamtspeichers)
- und kann Betriebssystem-Kern-Funktionen aufrufen.



**Programm:** Folge von Anweisungen

**Prozess:** laufendes Programm mit seinen Daten

Hinweis: ein Programm kann sich mehrfach in Ausführung befinden!



- Definition „Prozess“: laufendes Programm mit seinen Daten
- eine etwas andere Sicht:

Mikrocontroller-Prozess	UNIX-/Windows/...-Prozess
Prozessor	Zeitanteile am echten Prozessor
Speicher	virtueller Speicher
Interrupts	Signale
E/A-Geräte	E/A-Betriebssystem-Funktionen



- Mehrprogrammbetrieb („Multitasking“)
  - mehrere Prozesse können quasi gleichzeitig ausgeführt werden
  - stehen weniger Prozessoren zur Verfügung, als Prozesse ausgeführt werden sollen, werden Zeitanteile der Rechenzeit an die Prozesse vergeben (**Time Sharing System**)
  - die Entscheidung, welcher Prozess zu welchem Zeitpunkt wieviel Rechenzeit bekommt, trifft der Betriebssystem-Kern (**Scheduler**)
  - die Umschaltung zwischen Prozessen erfolgt durch den Betriebssystem-Kern (**Dispatcher**)
  - laufende Prozesse wissen nicht, an welchen Stellen auf andere Prozesse umgeschaltet wird



# Prozesszustände

---

Ein Prozess befindet sich in einem der folgenden Zustände

**Erzeugt:** (New)

Prozess wurde erzeugt, besitzt aber noch nicht alle zum Laufen notwendigen Betriebsmittel

**Bereit:** (Ready)

Prozess besitzt alle nötigen Betriebsmittel und ist bereit zu laufen

**Laufend:** (Running)

Prozess wird vom realen Prozessor ausgeführt

**Blockiert:** (Blocked)

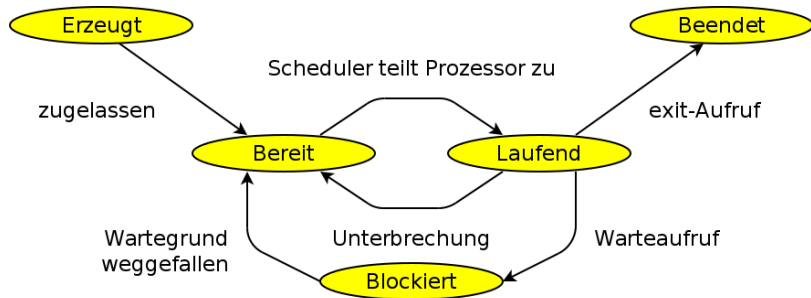
Prozess wartet auf ein Ereignis (Fertigstellung einer Ein- oder Ausgabeoperation)

**Beendet:** (Terminated)

Prozess ist beendet, seine Betriebsmittel sind noch nicht alle freigegeben



- Zustandsdiagramm mit Übergängen:



Nach Silberschatz, 1994



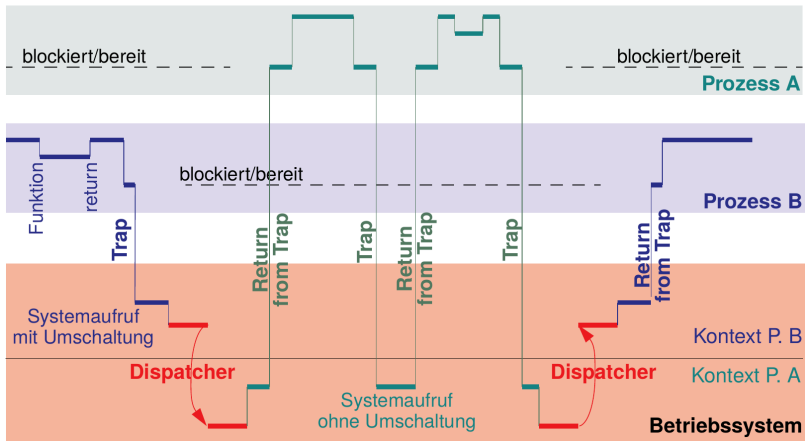
- Jeder Prozess hat Zustand/Kontext
  - Registerinhalte des Prozessors
  - Inhalte der Speicherbereiche
  - offene Dateien, aktuelles Verzeichnis, ...
- Beim Prozesswechsel (Context Switch)
  - wird der Inhalt der Prozessorregister abgespeichert,
  - ein neuer Prozess ausgewählt,
  - die Ablaufumgebung des neuen Prozesses hergestellt
    - Umprogrammierung der MMU
    - Wechsel der offenen Dateien, des aktuellen Verzeichnisses, ...
  - werden die gesicherten Register des neuen Prozesses geladen.





# Prozesswechsel

- Ablauf von zwei Prozessen in Benutzermodus und Kern mit Umschaltung



- Prozesskontrollblock (Process Control Block – PCB)

Datenstruktur des Betriebssystem-Kerns, die alle notwendigen Daten für einen Prozess enthält.

Beispiel UNIX:

- Prozess-ID (PID)
- Prozesszustand (Laufend, Bereit, ...)
- Register
- Speicherabbildung
- Eigentümer (UID, GID)
- Wurzelverzeichnis, aktuelles Verzeichnis
- offene Dateien
- ...

