Parallelisierung von Echtzeitanwendungen

Lukas Lehnert

24.11.2015

Inhaltszusammenfassung

Grundlagen
- sequentiell vs. parallel
- Chancen & Grenzen der Parallelisierung
- Verzögerungen

Sprachkonstrukte vs. BS-Schnittstelle

Analysierbarkeit

Synchronisation

Ablaufplanung

Grundlagen

Terminologie KvBK-Seminar

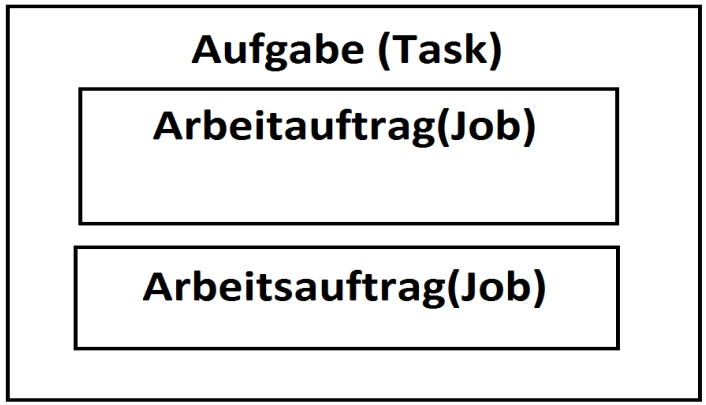


Abbildung 1: Aufgabe und Arbeitsauftrag

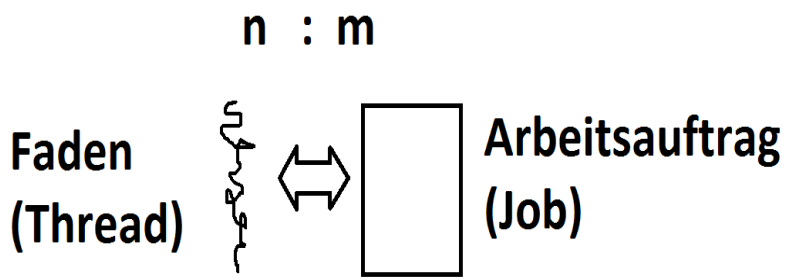


Abbildung 2: Faden und Arbeitsauftrag

Terminologie KvBK-Seminar

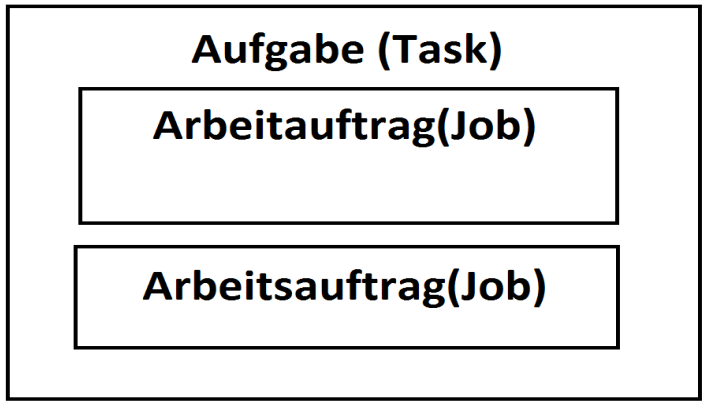


Abbildung 1: Aufgabe und Arbeitsauftrag

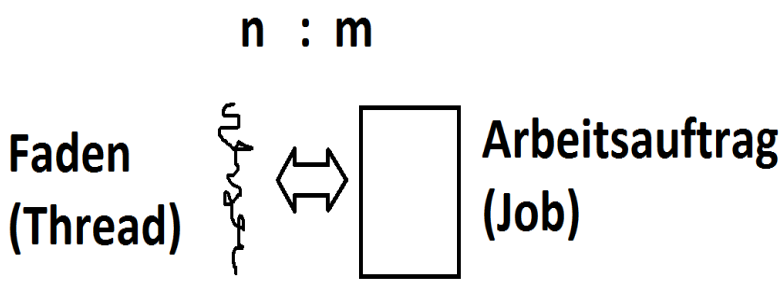


Abbildung 2: Faden und Arbeitsauftrag

Task $\neq$ Ada-Task $=$ Faden

Anwendung eines Echtzeitsystems

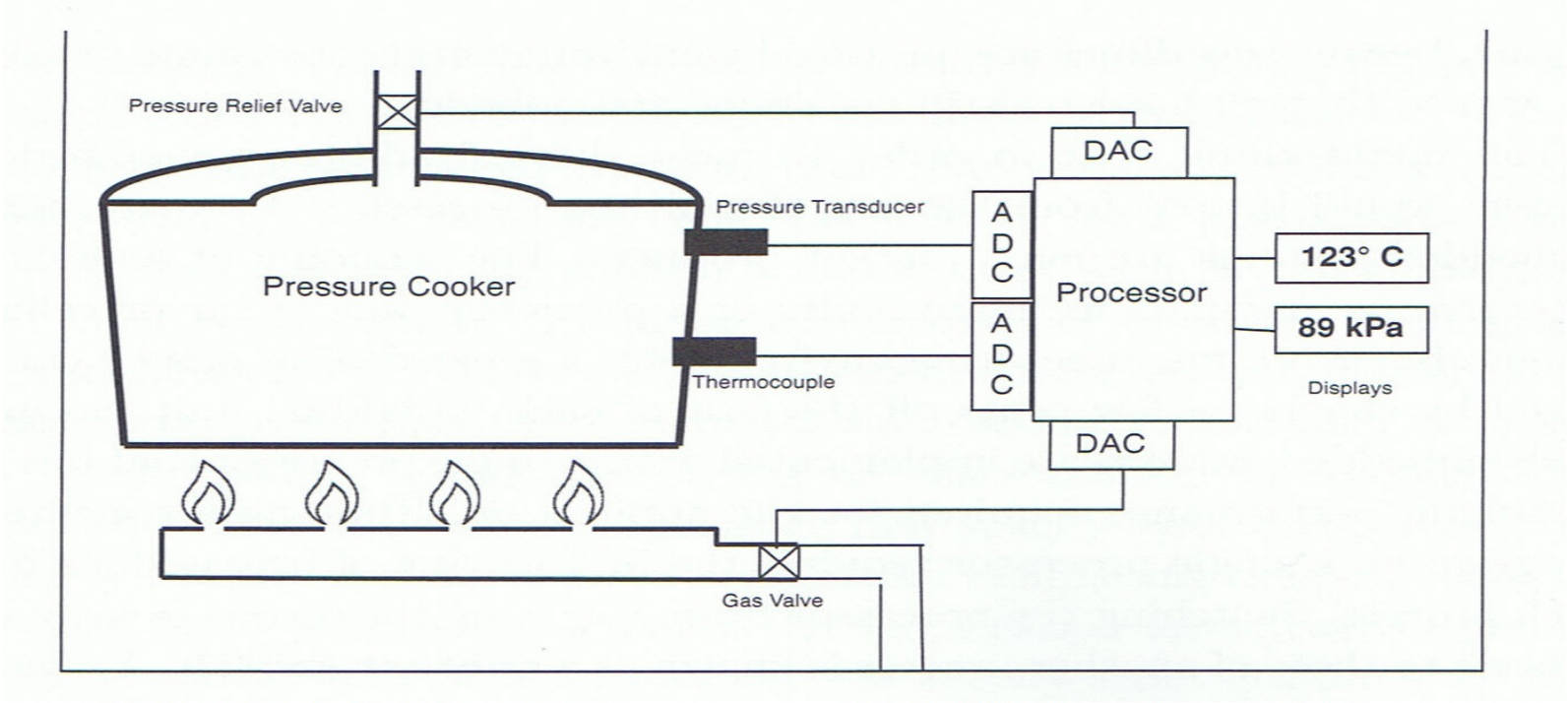


Abbildung 3: Bsp. für eine Echtzeitanwendung

Umsetzung der Arbeitsaufträge

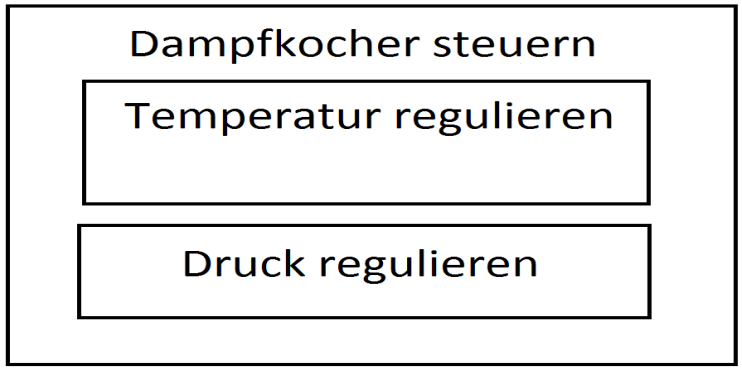


Abbildung 4: Aufgabe und Arbeitsauftrag praktisch

```
Druck regulieren:
  Hole Eingabe des ADC der Druckmessung
  Berechne neue Einstellung des Entlassungsventils
  Setzte mit dem DAC die berechnete Einstellung
  Zeige Druck am Display an.

Temperatur regulieren:
  Hole Eingabe des ADC der Temperaturmessung
  Berechne neue Einstellung des Gasventils
  Setze mit dem DAC die berechnete Einstellung
  Zeige Temperatur
```

paralleler Algorithmus

```
Druckregulierungsfaden:
loop
   Druck regulieren
   warte bis 3 Sekunden seit der letzten Druckmessung
end loop;
```

```
Temperaturregulierungsfaden:
loop
   Temperatur regulieren
   warte bis 5 Sekunden seit der letzten Temperaturmessung
end loop;
```

sequentieller Algorithmus

```
Dampfkochersteuerungsfaden:
loop
  Temperatur regulieren
  Druck regulieren

  warte bis t + 3s;t = t + 3s;
  Druck regulieren

  warte bis t + 2s;t = t + 2s;
  Temperatur regulieren

  warte bis t + 1s;t = t + 1s;
  Druck regulieren

  warte bis t + 3s;t = t + 3s;
  Druck regulieren

  warte bis t + 1s;t = t + 1s;
  Temperatur regulieren

  warte bis t + 2s;t = t + 2s;
  Druck regulieren

  warte bis t + 3s;t = t + 3s;
end loop;
```

Druck	Temperatur
0s	0s
3s	
	5s
6s	
9s	
	10s
12s	

Regel von Pollack

$$\sqrt{k} * \text{Rechenleistung} = k * \text{Komplexität}$$

bei Verdoppelung der Komplexität wäre das $\sqrt{2} \approx 1,41$ -fache der Rechenleistung

Grenzen der Parallelisierung

$$s = \frac{1}{(1 - P) + \frac{P}{N}}$$

s: **speedup**

P: parallelisierbarer Codeanteil

N: Anzahl an Prozessoren

Grenzen der Parallelisierung

$$s = \frac{1}{(1 - P) + \frac{P}{N}}$$

s: **speedup**
P: parallelisierbarer Codeanteil
N: Anzahl an Prozessoren

Beispiel:

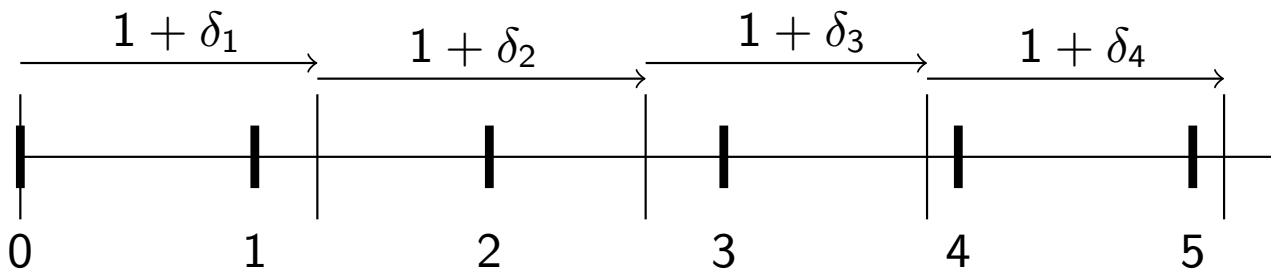
$$P = 0.7 \quad N = 2$$

$$\Rightarrow s \approx 1.54$$

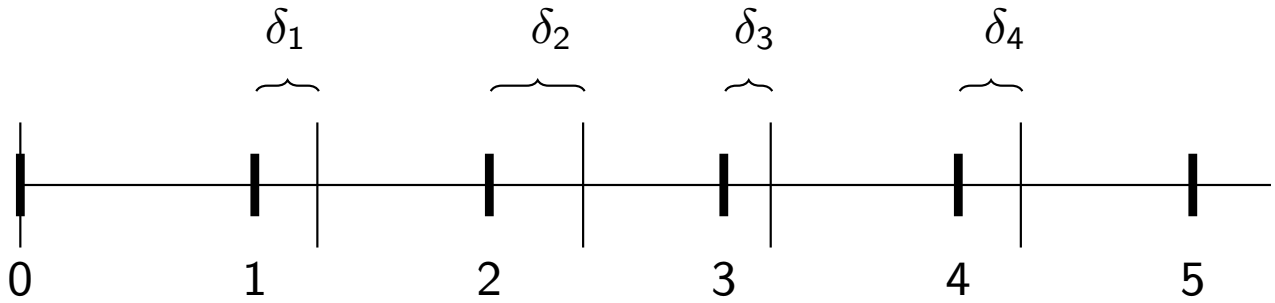
Verzögerungen

- relative Verzögerung
  - ab jetzigem Zeitpunkt
- absolute Verzögerung
  - ab Startzeitpunkt

Fehler bei Verzögerungungen



a) Phasenverschiebung bei relativer Verzögerung



b) Phasenverschiebung bei absoluter Verzögerung

Abbildung 5: Fehler bei relativer/absoluter Verzögerung

Sprachkonstrukte vs. Betriebssystemschnittstelle

absolute und relative Verzögerung

- Ada
  - `delay`
  - `delay until`
- eCos
  - `cyg_thread_delay`
- Real Time Java
  - `RelativeTime` / `AbsoluteTime`
  - `sleep`

Fäden

- Ada
  - Ada-Tasks in Sprache integriert
- eCos
  - `cyg_thread_create` erfordert Anlegung von Verwaltungsdaten
- Real Time Java
  - mit `RealTimeThread` Java-typischer Klassenaufbau

	Speicher explizit	expliziter Start
Real-Time Java		x
eCos	x	x
Ada		

Tabelle 1: Aufwand bei der Erstellung von Fäden

Beispiel: absolute Verzögerung in Ada

```
1  task Faden;
2  task body Faden is
3  Next   : Ada.Real_Time.Time := Clock;
4  Period : const Time_Span    := Seconds(3);
5  begin
6      loop
7          --...
8          Next := Next + Period;
9          delay until Next;
10     end loop;
11 end Faden;
```

Ressource Speicherplatz in Ada

- statische Allokation
- Stack
  - `pragma Storage_Size (1000)`
- Heap
  - ```
    pragma type Integer_Ptr is access Integer;
    N : Integer_Ptr;
    N := new Integer;
    ```


    ```

Zwischenresumée

- Ada
  - Ada-Tasks und delay-Anweisungen werden nativ in der Sprache unterstützt
- eCos
  - Verwendung externer Bibliotheken
  - Ressourcen müssen explizit angelegt werden
- Real Time Java
  - Vewendung externer Bibliothek
  - Klassenmodell

Analysierbarkeit

Ravenscar-Profil- Vorhersagbarkeit vs. Flexibilität

- in Ada 2005 Sprachstandard integriert
- Einschränkung der Sprachkonstrukte
- sequentielle Programmierung wird kaum eingeschränkt
- bessere Analysierbarkeit
- `pragma profile (Ravenscar)`

```
No_Relative_Delay
No_Implicit_Heap_Allocations
```

Ada 2005 vs. POSIX

- weite POSIX-Unterstützung in Betriebssystemen
- POSIX sprachunabhängiger Standard
- schlechtere Portabilität von Ada-POSIX Anwendungen

WCET

"From its beginnings, Ada was designed to allow such analysis in as easy a way as possible"

Building Parallel, Embeded, and Real-Time Appllications with Ada

Ansätze

- Messungen
- statische Analyse

WCET

"From its beginnings, Ada was designed to allow such analysis in as easy a way as possible"

Building Parallel, Embeded, and Real-Time Appllications with Ada

Ansätze

- Messungen
- statische Analyse

Synchronisation

Synchronisation

- nicht blockierende Synchronisation
  - transaktionaler Speicher
    - Volatile/Volatile_Components
  - atomares Lesen/Schreiben
    - Atomic/Atomic_Componentents
  - transaktionales Programm
    - CAS
    - LL/SC
- blockierende Synchronisation
  - Semaphor
  - Monitor
    - Ada: protected
    - Java: synchronized

Monitor

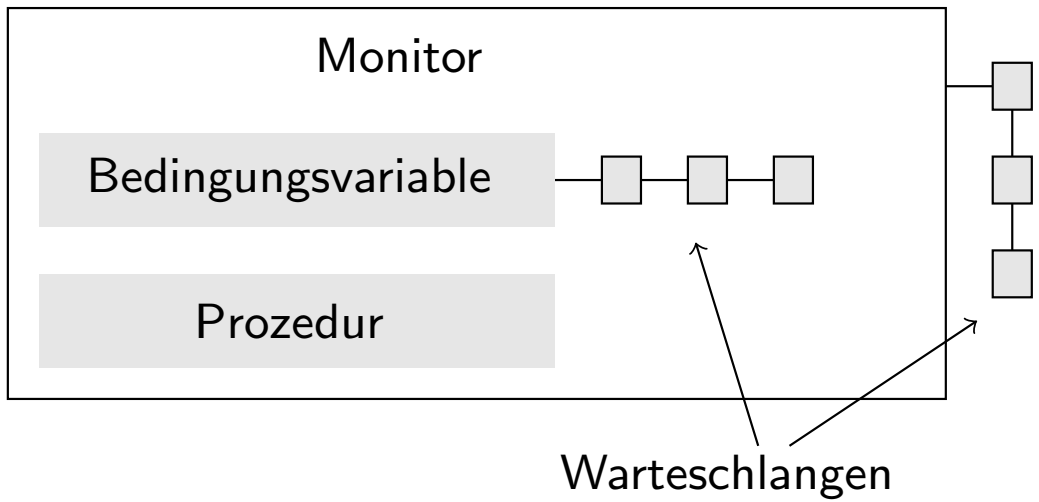


Abbildung 6: Monitorkonzept

- Hansen: alle aus Bedingungswartschlange "bereit"; blockiert
- Hoare: einer aus Bedingungswarteschlange "bereit"; blockiert
- Mesa: einer oder alle aus Bedingungswartschlange "bereit"

Ablaufplanung

Prozesszustände

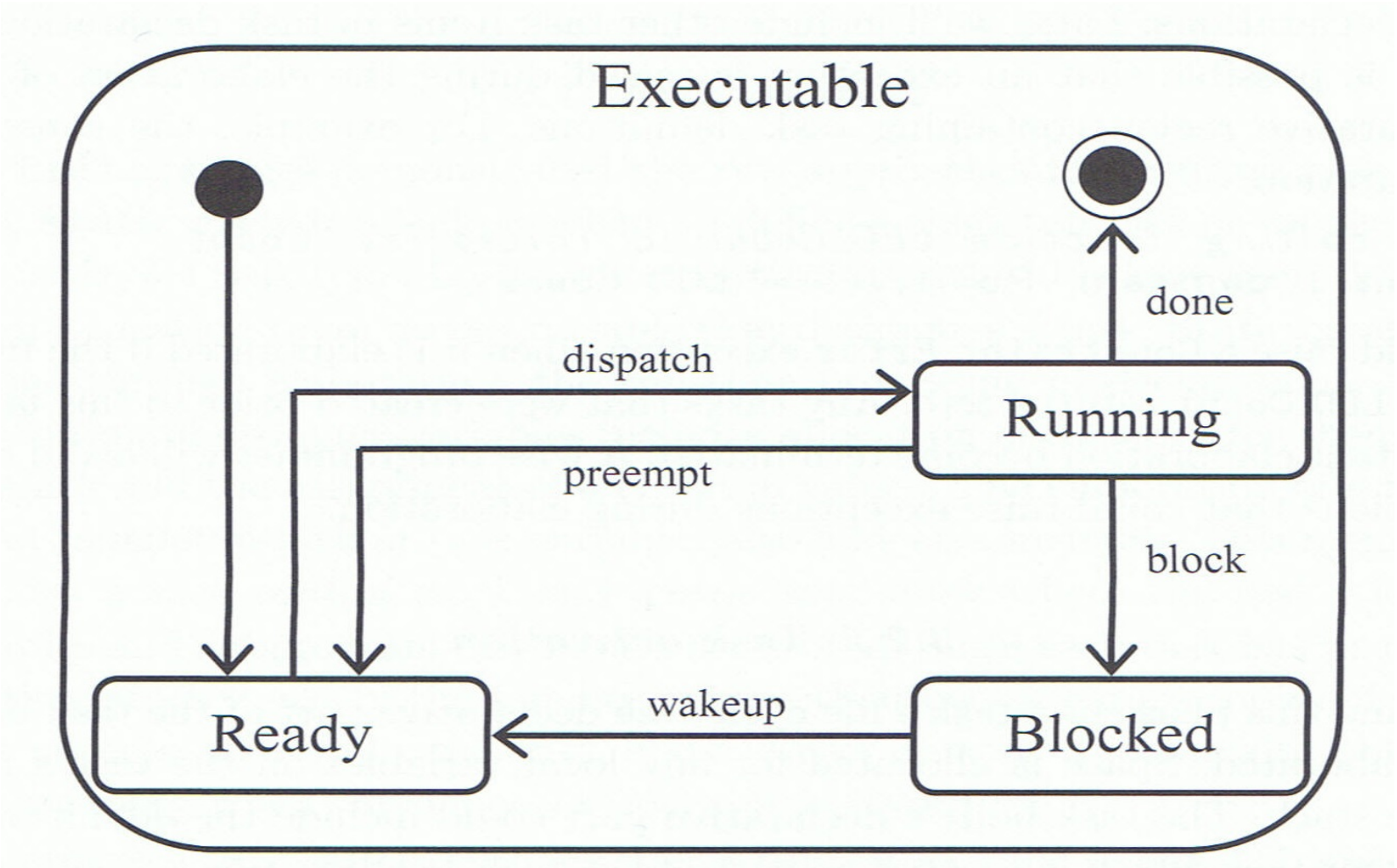


Abbildung 7: Prozesszustände

Übersicht Ablaufplanung

- zur Laufzeit dynamische Priorität (dynamic priority scheduling)
- vor Laufzeit festgelegte Priorität (fixed priority scheduling)
  - Dringlichkeit (urgency)
    - in Abh. von Periode (Rate Monotonic)
    - kürzeste Deadline (Deadline Monotonic)
  - Relevanz (criticality)
  - geteilte Ressourcen (shared ressources)
  - Vorrang (precedence)

Übersicht Ablaufplanung

- zur Laufzeit dynamische Priorität (dynamic priority scheduling)
- vor Laufzeit festgelegte Priorität (fixed priority scheduling)
  - Dringlichkeit (urgency)
    - in Abh. von Periode (Rate Monotonic)
    - kürzeste Deadline (Deadline Monotonic)
  - Relevanz (criticality)
  - geteilte Ressourcen (shared ressources)
  - Vorrang (precedence)

- dynamisch zur Laufzeit bestimmte Ablaufplanung (on-line scheduler)
- vor Laufzeit festgelegte Ablaufplanung (off-line scheduler)

Übersicht Ablaufplanung

- zur Laufzeit dynamische Priorität (dynamic priority scheduling)
- vor Laufzeit festgelegte Priorität (fixed priority scheduling)
  - Dringlichkeit (urgency)
    - in Abh. von Periode (Rate Monotonic)
    - kürzeste Deadline (Deadline Monotonic)
  - Relevanz (criticality)
  - geteilte Ressourcen (shared ressources)
  - Vorrang (precedence)

- dynamisch zur Laufzeit bestimmte Ablaufplanung (on-line scheduler)
- vor Laufzeit festgelegte Ablaufplanung (off-line scheduler)

- verdrängende Ablaufplanung (preemtive scheduler)
- nicht verdrängende Ablaufplanung (non-preemptive scheduler)

EDF vs. fixed priority scheduling

	Fixed Priority scheduling	EDF
Effizienz	69 % (Rate Monotonic)	100%
Vorhersagbarkeit	hoch	niedrig
Systemüberlastung	deterministischer	nicht deterministisch

Prioritätsinversion

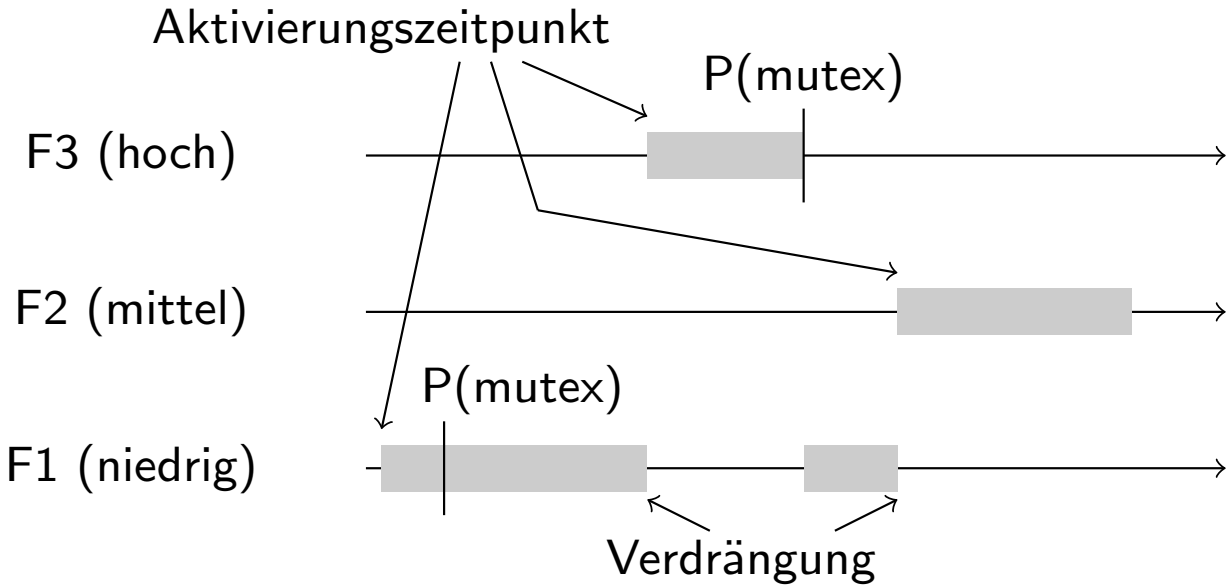


Abbildung 8: Prioritätsinversion

Ablaufplanung in Ada

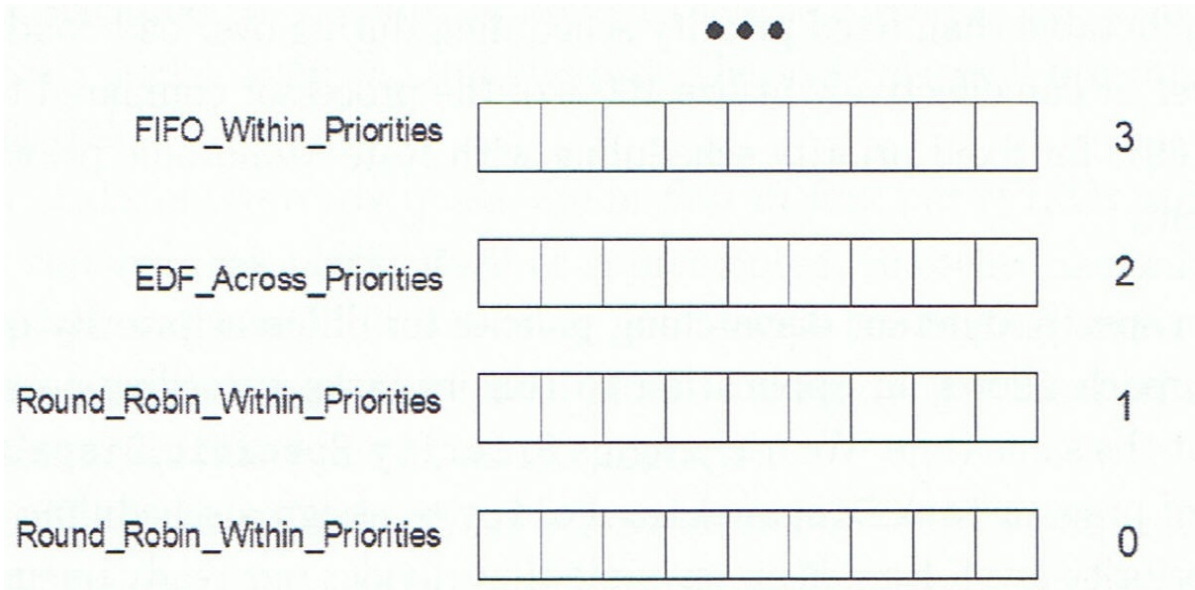


Abbildung 9: Ablaufplanung in Ada

- ```
  pragma Priortiy_Specific_Dispatching
         (FIFO_Within_Priorities, 3, 31);
  ```
- ```
  pragma Task_Dispatching_Policiy
      (FIFO_Within_Priorities);
  ```
- ```
  pragma Locking_Policy (Ceiling_Locking);
  ```


  ```

sequentieller Algorithmus

```
Dampfkochersteuerungsfaden:
loop
  Temperatur regulieren
  Druck regulieren

  warte bis t + 3s;t = t + 3s;
  Druck regulieren

  warte bis t + 2s;t = t + 2s;
  Temperatur regulieren

  warte bis t + 1s;t = t + 1s;
  Druck regulieren

  warte bis t + 3s;t = t + 3s;
  Druck regulieren

  warte bis t + 1s;t = t + 1s;
  Temperatur regulieren

  warte bis t + 2s;t = t + 2s;
  Druck regulieren

  warte bis t + 3s;t = t + 3s;
end loop;
```

Druck	Temperatur
0s	0s
3s	
	5s
6s	
9s	
	10s
12s	
ab 15s Hyperperiodenende	

Hyperperiode:
kleinstes gemeinsames Vielfaches aller Zykluszeiten.

Zusammenfassung

- Motivation
  - seq. vs. parallele Algorithmen
  - Regel von Pollack
  - speedup
- Verzögerungen
  - absolut
  - relativ
- Sprachunterstützung vs. externe Bibliotheken
- gute Analysierbarkeit von Ada
- blockierende vs. nicht blockierende Synchronisation
- Ablaufplanung
  - dynamisch/festgelete Prioritäten
  - Prioritäten & unteilbare Betriebsmittel

Quellen 1

- G. Bollella, B. Brosgol, P. Dibble, S. Furr, J. Gosling, D. Hardin, and M. Turnbull. The Real-Time Specifcation for Java. Addison-Wesley, 2000
- B. Brandenburg, J. Calandrino, A. Block, H. Leontyev, and J. Anderson. Synchronization on real-time multiprocessors: To block or not to block, to suspend or spin? In Proceedings of the 14th IEEE Real-Time and Embedded Technology and Applications Symposium, pages 342-353, 2008.
- A. Burns and A. Wellings. Concurrent and Real-Time Programming in Ada. Cambridge University Press, Cambridge, 2007.
- A. Burns and A. Wellings. Locking Policies for Multiprocessor Ada. ACM Ada Letters, 33(2):59-65, 2013.

Quellen 2

- E. G. Coffman, Jr., M. J. Elphick, A. Shoshani. System deadlocks. ACM Computing Surveys (CSUR), 3(2):67-78, 1971.
- Free Software Foundation, Inc. eCos Reference Manual, http://ecos.sourceware.org/docs-3.0/pdf/ecos-3.0-refa4. pdf 2008.
- J. W. McCormick, F. Singhoff, and J. Hugues. Building Parallel, Embedded, and Real-Time Applications with Ada. Cambridge University Press, Cambridge, 2011.
- Oracle Corporation. Deterministic garbage collection: Unleash the power of java with oracle jrockit real time, http://www.oracle.com/us/technologies/java/oraclejrockit-real-time-1517310.pdf 2008.
- B. Venu. Multi-core processors - an overview. CoRR, abs/1110.3535, 2011.

Abbildungsverzeichnis

John W. McCormick, Frank Singhoff and Jérôme Hugues: Building Parallel, Emebedded, and Real-Time Applications with Ada, Cambridge University Press, Cambridge, 2011

- Abbildung 3: S. 16
- Abbildung 5: S. 301
- Abbildung 7: S. 112
- Abbildung 8: S. 279
- Abbildung 9: S.312

Fragen?

42