

Echtzeitbetriebssysteme

Oliver Jack

Ernst-Abbe-Hochschule Jena
Fachbereich Elektrotechnik und Informationstechnik

Sommersemester 2025



Ernst-Abbe-Hochschule Jena
University of Applied Sciences

Lerneinheit 6. Scheduling-Verfahren Teil 2

- 1 Lernziele dieser Lerneinheit
- 2 Zuteilung nach Prioritäten
- 3 Zeitplanung periodischer Prozesse
- 4 Vergleich der Planungsverfahren
- 5 Zusammenfassung

Lerneinheit 6. Scheduling-Verfahren Teil 2

- 1 Lernziele dieser Lerneinheit
- 2 Zuteilung nach Prioritäten
- 3 Zeitplanung periodischer Prozesse
- 4 Vergleich der Planungsverfahren
- 5 Zusammenfassung

Lernziele

- Kenntnis weiterer Planungsverfahren.
- Kenntnis der Planung nach Prioritäten.
- Kenntnis der Planung periodischer Prozesse

Lerneinheit 6. Scheduling-Verfahren Teil 2

- 1 Lernziele dieser Lerneinheit
- 2 Zuteilung nach Prioritäten**
- 3 Zeitplanung periodischer Prozesse
- 4 Vergleich der Planungsverfahren
- 5 Zusammenfassung

Allgemeines

Die einfachen Strategien S1 und S2 werden in der Praxis auch bei Einprozessorsystemen nicht explizit angewandt, da

- kein abgeschlossenes System der Aktionen (Alarmer, Interrupts erfordern dynamische Planung)
- Bereitzeitpunkt nur bei zyklischen und Termin-Prozessen bekannt,
- Laufzeiten nicht exakt bekannt, von Daten und Umgebung abhängig
- Synchronisation, Kommunikation und gemeinsame Betriebsmittel verletzen die Forderung nach Unabhängigkeit der Aktionen

Allgemeines (Forts.)

- Bei großen Systemen Zuteilung nach festen Prioritätszahlen
- Priorität aus Wichtigkeit für den technischen Prozess und aus Abschätzungen der aktuellen Fristen oder Spielräume
- Bei gleicher Priorität meist FIFO-Strategie
- Ausreichende Leistungsreserve der CPU mittels worst-case-Betrachtungen

Allgemeines (Forts.)

- Prioritätszahlen sind meist vom Typ char im Bereich [0..255]
- Je größer die Zahl, desto höher die Dringlichkeit (z.B. LYNXOS); oft auch umgekehrt (z.B. VxWorks, PEARL, UNIX)
- Einteilung in Prioritätsgruppen, je nach Härte der Zeitbedingungen

Allgemeines (Forts.)

Wann wird umgeplant ?

- Prioritätsinkonsistenz möglichst kurz
- Beim Beenden einer Aktion
- Beim Übergang einer Aktion in Wartezustand.
- Beim Eintreffen einer neuen Anforderung (neuer Prozess wird aktiv)
- Nach bestimmten Zeitintervallen Überprüfung der Situation (z.B. bei Spielraumplanung)

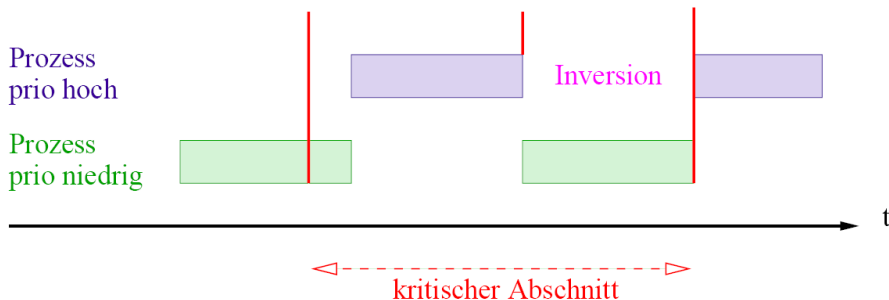
Prioritätsinversion (Priority inversion)

- **Behinderung** wichtiger Prozesse durch unwichtige darf in Echtzeitsystemen nicht auftreten, also
- Einhaltung der **Prioritätsreihenfolge** bei allen Anforderungen von Betriebsmitteln (CPU, Semaphore, Netzkommunikation, Puffer, Peripherie), d. h. **kein Vordrängen** den Warteschlangen
- Prioritätsinversion: Prozess mit **niedriger** Priorität **blockiert** einen Prozess mit **höherer** Priorität.

Begrenzte Inversion (Bounded inversion)

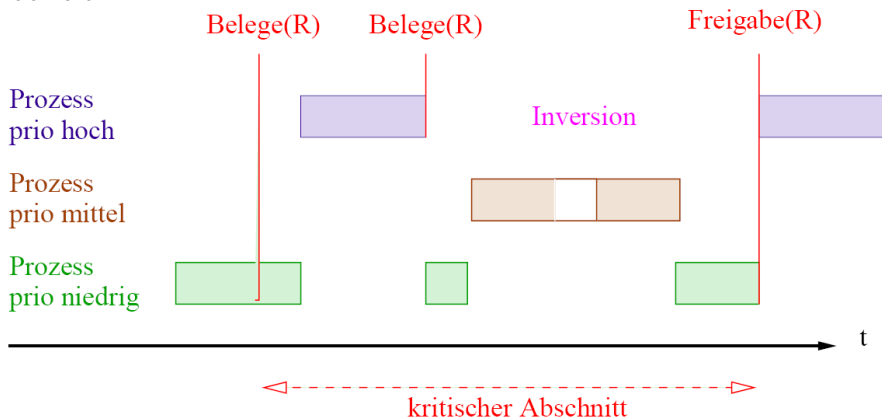
Die Inversion ist durch die **Dauer des kritischen Abschnitts** beschränkt.

Belege(R) Belege(R) Freigabe(R)



Unbegrenzte Inversion (Unbounded inversion)

Der kritische Abschnitt wird durch weitere Prozesse auf **unbestimmte Zeit** blockiert.



Prioritätsvererbung (Priority inheritance)

- Der Prozess **erbt** die höhere Priorität des Prozesses, solange dieser das gemeinsame Betriebsmittel blockiert.
- Verhindert unbegrenzte Blockierung
- Die Dauer der Blockierung wird auf die Dauer des kritischen Abschnitts begrenzt
- Die Blockierungen werden hintereinander gereiht (Blockierungsketten)
- Es verhindert keine deadlocks



Prioritätsgrenzen (Priority ceiling)

- Jedes Betriebsmittel (Semaphor) s erhält eine Prioritätsgrenze $ceil(s) = \text{Maximum der Prioritäten der Prozesse, die auf } s \text{ zugreifen}$
- Der Prozess p darf ein BM (Betriebsmittel) nur blockieren, wenn er von keinem anderen Prozess, der andere Betriebsmittel besitzt, verzögert werden kann

Prioritätsgrenzen (Forts.)

- Aktuelle Prioritätsgrenze für Prozess p

$$\text{aktceil}(p) = \max\{\text{ceil}(s) \mid s \in \text{lockedsem}\}$$

lockedsem = Menge aller von anderen Prozessen blockierten BM

- Prozess p darf Betriebsmittel s benutzen, wenn

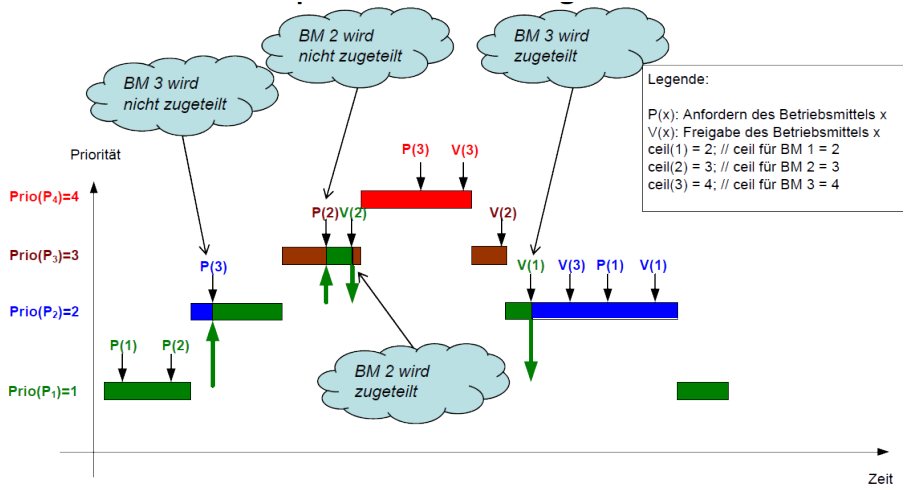
$$\text{aktprio}(p) > \text{aktceil}(p)$$

- Andernfalls gibt es genau einen Prozess, der s besitzt. Die Priorität dieses Prozesses wird auf $\text{aktprio}(p)$ gesetzt

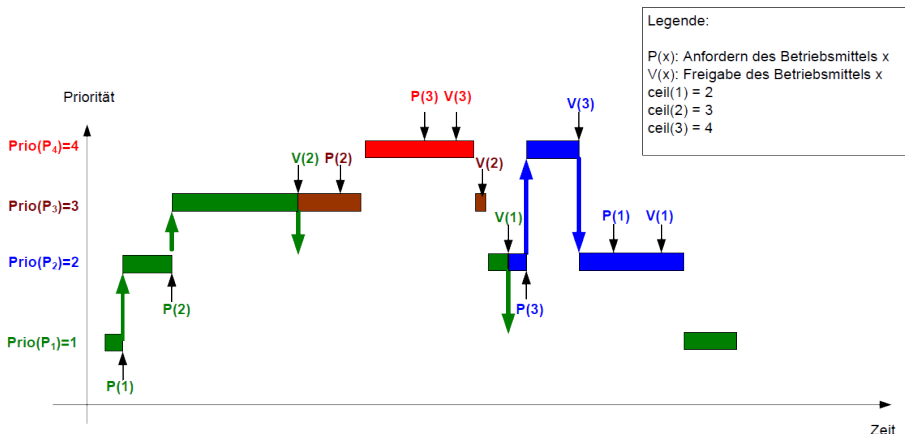
Prioritätsgrenzen (Forts.)

- Blockierung nur für die Dauer eines kritischen Abschnitts
- Verhindert Deadlocks
- Schwieriger zu realisieren, zusätzlicher Prozesszustand
- Vereinfachtes Protokoll:
Immediate priority ceiling
Prozesse, die Betriebsmittel s belegen, erhalten die Priorität $\text{ceil}(s)$

Beispiel mit Prioritätsgrenzen



Beispiel mit Immediate Priority Ceiling



Lerneinheit 6. Scheduling-Verfahren Teil 2

- 1 Lernziele dieser Lerneinheit
- 2 Zuteilung nach Prioritäten
- 3 Zeitplanung periodischer Prozesse**
- 4 Vergleich der Planungsverfahren
- 5 Zusammenfassung

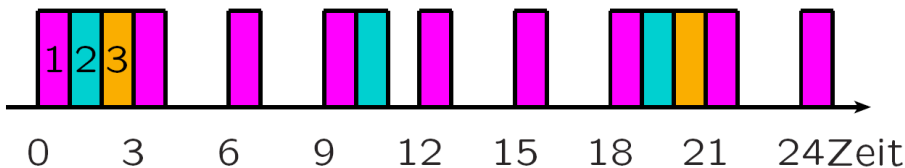
Bezeichnungen

- Prozesse werden periodisch aktiviert
- n Prozesse $p_i, i = 1, \dots, n$
- Periode $T(p_i)$
- Ausführungszeit $a(p_i)$
- $B(p_i)$ Bereitzeit relativ zum Beginn einer Periode
- $F(p_i)$ Frist relativ zum Beginn einer Periode

Beispiel: Messwerterfassung

- Drei Prozesse p_i
- Jeweils Werte einlesen, umrechnen, skalieren, auf Platte speichern
- Kenngrößen der Prozesse:

i	$T(p_i)$	$B(p_i)$	$a(p_i)$	$F(p_i)$
1	3	0	1	3
2	9	0	1	9
3	18	0	1	18



Einplanbarkeit auf Grund von Last

- Auslastung durch Prozess P_i

$$\varrho_i = \frac{a(p_i)}{T(p_i)}$$

- Gesamte Auslastung

$$\varrho = \sum_{i=1}^n \varrho_i = \sum_{i=1}^n \frac{a(p_i)}{T(p_i)}$$

- Bei m Einheiten eines Betriebsmittel gilt: $\varrho \leq m$ notwendig für Zeiteinhaltung, aber nicht hinreichend!

Einplanung nach Fristen

- Bei einem Betriebsmittel, $\varrho \leq 1$ und $T(p_i) = F(p_i)$ für alle i ist Einplanung nach Fristen optimal.
- Beweisidee: vor dem Verletzen einer Frist ist das Betriebsmittel nie unbeschäftigt

Einplanung nach Raten (Rate-monotonic scheduling, Smallest period-first)

- Rate $R_i = 1/T(p_i)$
- hohe Rate = hohe Priorität
- Prozesse mit hohen werden Raten zuerst bedient.
- Ratenplanung benutzt also statische Prioritäten.
- Optimal, falls eine Lösung mit statischen Prioritäten existiert.
- Verfahren mit dynamischen Prioritäten können aber evtl. bessere Ergebnisse liefern.

Einplanbarkeitstest 1 für Ratenplanung

- Bei n Aktionen ist Ratenplanung sicher erfolgreich, falls

$$\begin{aligned}\varrho = \varrho_{\max} &= n * (2^{1/n} - 1) = n * (\sqrt[n]{2} - 1) \\ \lim_{n \rightarrow \infty} \varrho_{\max} &= \ln 2 \approx 0.69\end{aligned}$$

- Beweis (Liu, Layland 1973) durch Worstcase-Analyse
- Das heißt aber nicht, dass Ratenplanung für Systeme mit $\varrho > \varrho_{\max}$ nicht doch Lösungen findet, nur ist dies nicht garantiert.
- Für Spezialfälle (Perioden gleich oder Perioden Teiler einer größeren) gibt es günstigere Abschätzungen.

Einplanbarkeitstest 2 für Ratenplanung

- Für einen Satz periodischer Aktionen p_i gibt die Ratenplanung einen erfolgreichen Plan, falls unter der Annahme gleichzeitigen Starts (worst case) alle Aktionen innerhalb ihrer ersten Periodenzeit $T(p_i)$ beendet werden.
- Der Test ist auch bei grosser Anzahl von Prozessen durch Simulation effizient ausführbar, falls die Perioden nicht teilerfremd sind.
- Beweisidee: Gleichzeitiger Start ist worst case-Situation, da alle höherprioren Aktionen eine Ausführung von p_i verzögern

Lerneinheit 6. Scheduling-Verfahren Teil 2

- 1 Lernziele dieser Lerneinheit
- 2 Zuteilung nach Prioritäten
- 3 Zeitplanung periodischer Prozesse
- 4 Vergleich der Planungsverfahren**
- 5 Zusammenfassung

Planungsverfahren

Strategie	präemptiv	nicht präemptiv
Suchen		optimale Pläne $O(n!)$, NP-vollständig
Fristen Spielraum	optimal	optimal bei gleicher Bereitzeit
Spielraum (Mehrpr.)	optimal gleicher Bereitzeit	bei NP-vollständig Anomalien
Monotone Raten	optimal bei beschränkter Auslastung	

Lerneinheit 6. Scheduling-Verfahren Teil 2

- 1 Lernziele dieser Lerneinheit
- 2 Zuteilung nach Prioritäten
- 3 Zeitplanung periodischer Prozesse
- 4 Vergleich der Planungsverfahren
- 5 Zusammenfassung**

Zusammenfassung

- In der Praxis wird oft nach **Prioritäten** geplant.
- Zuteilung nach Prioritäten muss das Problem der **Prioritätsinversion** beachten und behandeln. Verfahren dazu sind **Prioritätsvererbung** und **Prioritätsgrenzen**.
- Ein erstes Verfahren zur Planung **periodischer Prozesse** ist die **Ratenplanung** mit statischen Prioritäten.