

# Einleitung

---

Autoren: Josef Mueller, Isabella Schoen

## Allgemeine Fragen

### 1. Was ist ein Realzeitsystem?

Ein Realzeitsystem ist ein System, welches sowohl funktionale als auch zeitliche Anforderungen aufweist. (S.7)

### 2. Erläutern Sie den Unterschied zwischen weicher und harter Realzeit anhand der Benefit-Funktion.

Anhand der Benefit-Funktion lässt sich aussagen, wie nützlich die Reaktion des Realzeitsystems auf die Rechenzeitanforderung ist. Der Nutzen steht somit im Fokus:

2) bei einem harten Realzeitsystem beträgt der Nutzen der Reaktion nur innerhalb der zulässigen Deadline 100%

3) bei einem weichen Realzeitsystem ist auch ein Nutzen vorhanden, wenn die Reaktion außerhalb des Zeitfensters stattfindet

(S.21)

### 3. Nennen Sie Beispiele für weiche und harte Realzeitsysteme.

1. Weiche Realzeitsysteme: Verzögerungen werden toleriert -> System für Videokonferenzen, Telefonanlagen, Linux

2. Harte Realzeitsysteme: Verzögerungen haben massive Folgen -> Überwachungssysteme, Antiblockiersysteme, Airbags, Bremsen, Sicherungen

### 4. Definieren Sie den Begriff "Reaktionszeit".

Unter der Reaktionszeit wird die Zeit verstanden in welcher eine Rechenzeitanforderung passiert. Also von Beginn der Anforderung bis zum Ende der Bearbeitung. Die Warte- und Ausführungszeit ist auch hierbei mit inbegriffen.

(S.16)

### 5. Wer gibt die maximal zulässige Reaktionszeit vor? In welchem zeitlichen Bereich liegt sie?

Der technische Prozess legt die zulässige Reaktionszeit mit dem Bereich der minimalen und maximalen Deadline fest. Diese Deadline ist ein wichtiger Aspekt für die Reaktionszeit, womit der früheste und späteste Zeitpunkt einer Reaktion auf eine Rechenzeitanforderung bestimmt wird.

(S.10)

### 6. Wie lauten die beiden Realzeitbedingungen?

Die Bearbeitung einer Rechenzeitanforderung ist erst dann fertiggestellt, wenn beide Bedingungen von der Realzeit erfüllt sind: die Auslastungs- und Rechtzeitigkeitsbedingung.

1. Auslastungsbedingung: es ist hinreichend notwendig, dass die Gesamtauslastung für jeden Kern kleiner als 100% ist. Um das zu erreichen, muss die zweite Bedingung erfüllt sein.
2. Rechtzeitigkeitsbedingung: die benötigten Daten müssen zu dem vorgegebenen Zeitpunkt genau ankommen, andernfalls haben diese keinen Wert.

(S.17-20)

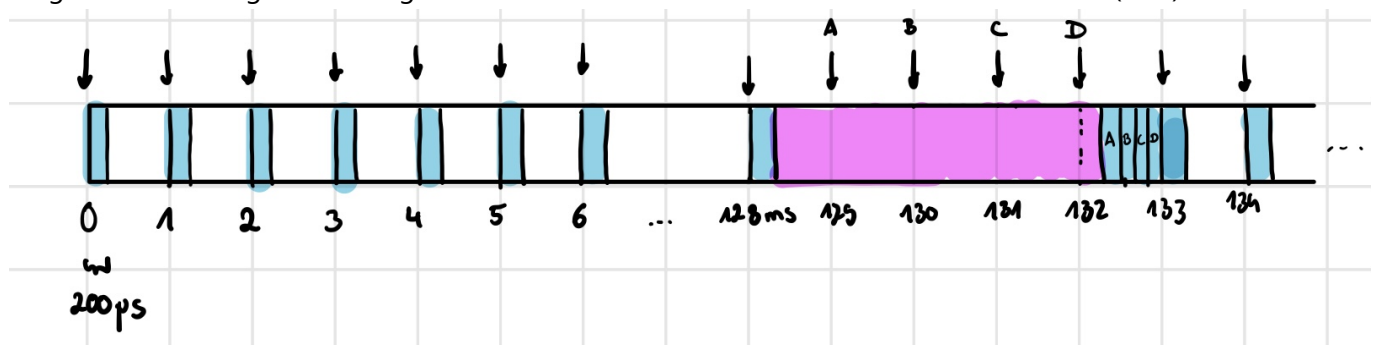
## Analogdatenverarbeitung

1. Warum kann die Analogwerterfassung ohne die Aufsplittung der Aufgabe in mehrere, gestellte Tasks nicht in Echtzeit erfolgen? Zeichnen Sie hierzu das Zeitdiagramm der Rechnerkernbelegung und **stellen Sie die Realzeitbedingungen auf.**

2.1



Die folgende Abbildung stellt das lineare Abarbeiten der Aufgaben ohne Aufsplittungen dar. Während die erste Realzeitbedingung nicht verletzt wird, ist die zweite Bedingung, die **Rechtzeitigkeitsbedingung**, aufgrund der verzögerten Verfügbarkeit der Werte verletzt. Die Werte sind damit wertlos. (S.23)



2. Um die Analogwerterfassung in Realzeit zu ermöglichen, werden im Speicher des Steuerrechners zwei Speicherbereiche, in denen je 128 Werte abwechselnd abgelegt werden können, zur Verfügung gestellt (Wechsellpuffer). Welche Strategie für die Erfassung der Analogwerte, dem Ablegen im Speicher und der Überspielung der 128-Werte-Blöcke auf den Zentralrechner, muss angewendet werden, um das Problem in Echtzeit lösen zu können? Welche Bedingung muss hierbei erfüllbar sein? Wie groß ist die Rechnerauslastung?

Es handelt sich um die Unterbrechbarkeit (Preemptibility). Hierfür muss der ausgeführte Code unterbrechbar sein und es muss sichergestellt werden, dass die Auslastungsbedingung nicht verletzt wird.

Auslastung (Rechner) = Summe der Auslastung der gesamten Tasks =  $0.2\text{ms} / 1\text{ms} = 0.2 = 20\%$

0.2ms, weil die Übernahme eines Wertes 200 microsec braucht; und 1ms, weil die Abtastrate 1KHz benötigt.

Wenn zwei Puffer zur Verfügung stehen, kann der eine ausgelesen werden, während der andere befüllt wird. Es ist jedoch unklar gewesen, wie die Berechnung der Auslastung hierbei erfolgt.

2.2

## Auslastung

1. Berechnen Sie die Auslastung des Rechners für die Tasks A und B.

(S.18-19)

1. Auslastung(A) =  $\text{Verarbeitungszeit(A)} / \text{Prozesszeit(A)} = (1\text{ ms} / 5\text{ ms}) = 0.2 = 20\%$
2. Auslastung(B) =  $\text{Verarbeitungszeit(B)} / \text{Prozesszeit(B)} = (4\text{ ms} / 10\text{ ms}) = 0.4 = 40\%$

2. Durch welche Werte ist der Worst Case für den Rechner gekennzeichnet? Geben Sie für diesen Fall die Gesamtauslastung an.

Der Worst-Case tritt auf, wenn die Ausführungszeit maximal und die Prozesszeit minimal wird.

Wie in der Tabelle erkennbar, variieren die Werte von Task C und Task D und bieten daher mehrere Möglichkeiten bei der Ergebnisse bei der Berechnung der Auslastung.

$$\text{Auslastung(Gesamt)} = \text{Verarbeitungszeit(Gesamt)} / \text{Prozesszeit(Gesamt)} = \text{Auslastung(A)} + \text{Auslastung(B)} + \text{Auslastung(C)} + \text{Auslastung(D)} = 0.2 + 0.4 + (5/20) + (5/50) = 0.95 = 95\%$$

3. Wie hoch ist die durchschnittliche Auslastung, wenn Sie von einer Gleichverteilung der Prozess und Ausführungszeiten ausgehen?

Hierbei sind wir uns nicht ganz sicher. Vorstellbar wäre die Abhängigkeit von Task C und Task D. Mittlerer Wert des vorgegebenen Intervalls für die Rechnung verwenden? Oder ist das Ergebnis doch 1? 3.1

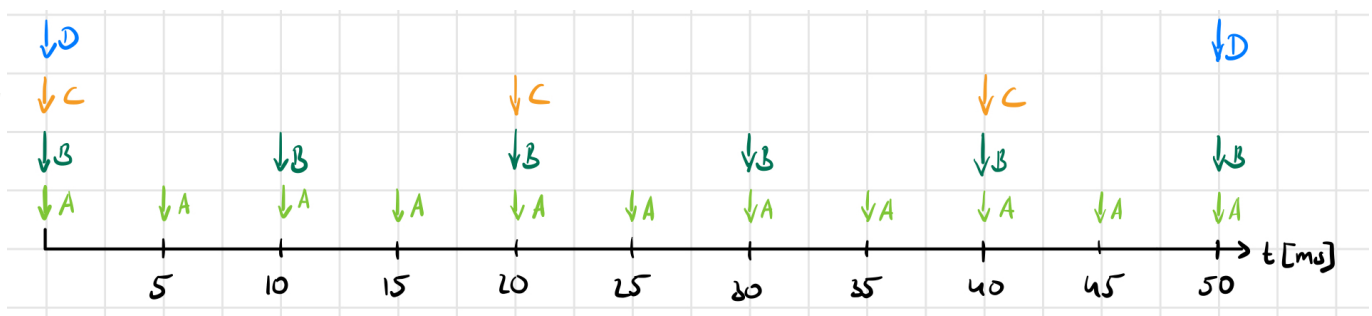
4. Geben Sie den 4 Tasks Prioritäten.

Die Priorität beschreibt die Ablaufreihenfolge der einzelnen Tasks.

Mehrere Scheduling-Verfahren stehen zur Verfügung, um die Priorität zu bestimmen. In diesem Fall steht die niedrige Zahl für eine hohe und die hohe Zahl für eine niedrige Priorität:

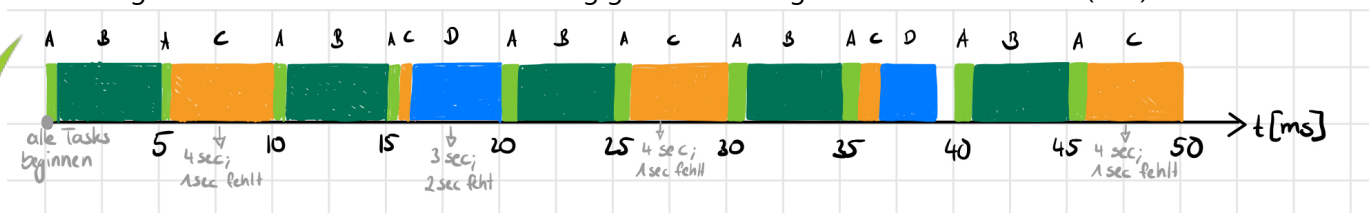
- A->Prio1
- B->Prio2
- C->Prio3
- D->Prio4

5. Zeichnen Sie die Anforderungsfunktion (Worst Case).



6. Zeichnen Sie die Rechnerkernbelegung durch die 4 Tasks (Worst Case).

Wird die Verarbeitungszeit (tE) über die Zeit aufgetragen, erhält man als Ergebnis die Rechnerkernbelegung, welche angibt zu welcher Zeit eine Verarbeitung geschieht. Wir gehen von der Deadline(min) = 0 aus:



(S. 12)

7. Geben Sie die maximal zulässigen Reaktionszeiten der 4 Tasks an.

(S.11)

- TaskA -> 5ms
- TaskB -> 10ms
- TaskC -> 20ms
- TaskD -> 50ms

## Parameter Bestimmung

1. Leiten Sie aus dem Diagramm die Prozesszeiten und die Verarbeitungszeiten der einzelnen Tasks Task A, Task B und Task C her. Gehen Sie davon aus, dass die Prozesszeiten und die Verarbeitungszeiten der einzelnen Tasks konstant sind.

1. TaskA: Prozesszeit = 50, Verarbeitungszeit = 5
2. TaskB: Prozesszeit = 20, Verarbeitungszeit = 5
3. TaskC: Prozesszeit = 25, Verarbeitungszeit = 10

2. Nach welcher Faustregel erfolgt im allgemeinen die Prioritätenvergabe?

In der Regel werden hohe Prioritäten durch niedrige Zahlen und niedrige Prioritäten durch hohe Zahlen repräsentiert. 4.1

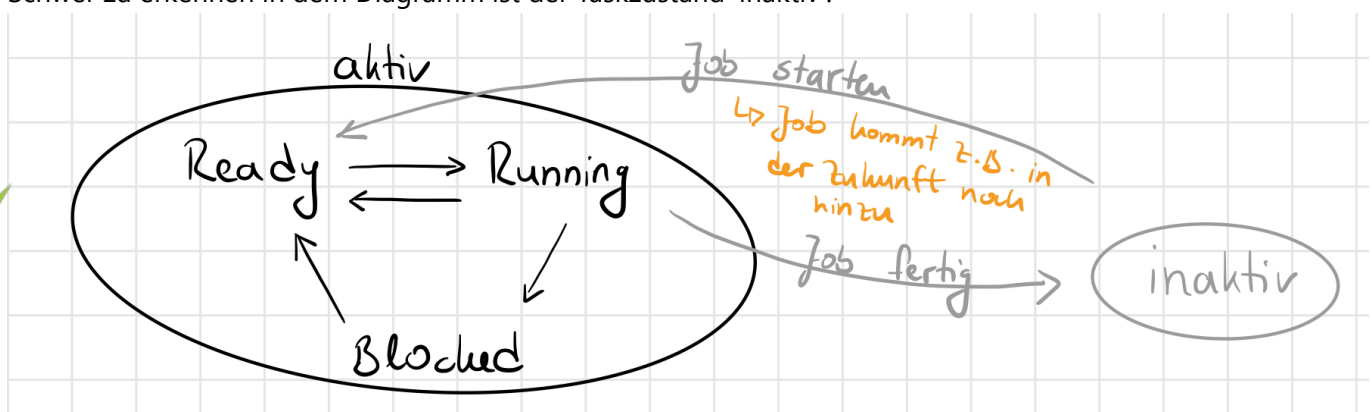
Ein Prozess, der viel CPU-Zeit in Anspruch nimmt, erhält meist eine niedrige Priorität, also eine hohe Zahl. Hingegen erhalten kurz laufende Tasks eine höhere Priorität, also eine niedrige Zahl.

3. Geben Sie die Prioritäten (von 1 bis 3) der Tasks an, wobei 1 der höchsten und 3 der niedrigsten Priorität entspricht.

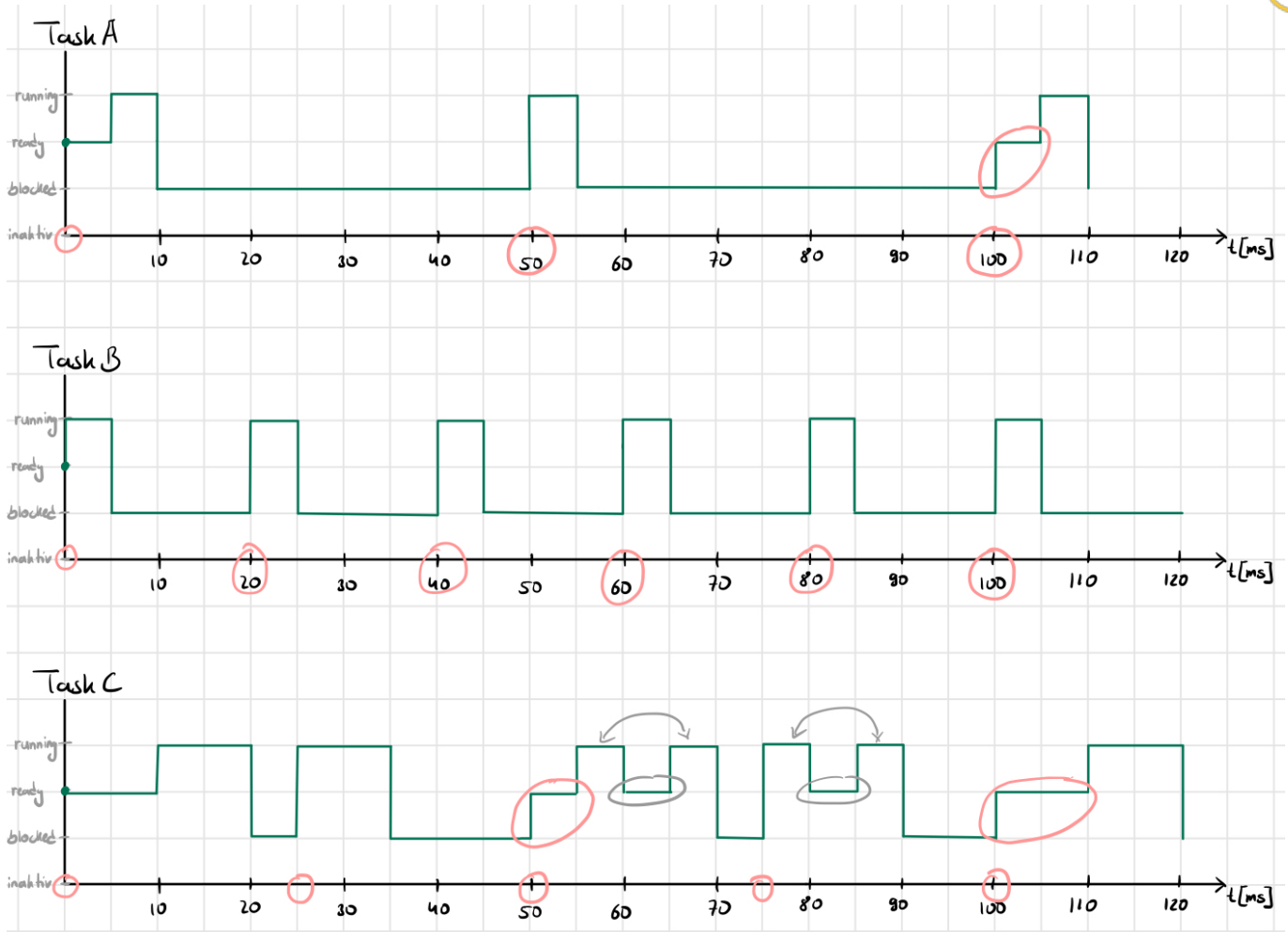
- B -> Prio1
- A -> Prio2
- C -> Prio3

4. Skizzieren Sie die möglichen Zustände und Zustandsübergänge einer Task in einem Realzeitbetriebssystem. Welche Taskzustände lassen sich im obigen Diagramm nicht eindeutig unterscheiden?

Schwer zu erkennen in dem Diagramm ist der Taskzustand 'inaktiv'.



5. Erstellen Sie für jede Task ein Diagramm, in dem Sie den Taskzustand über der Zeit eintragen, wenn von der oben angegebenen Prozessorbelegung der einzelnen Tasks ausgegangen wird.



6. Geben Sie die relative Belastung durch die einzelnen Tasks, sowie die Gesamtbelastung für den Realzeitrechner an.

1. Auslastung(A) =  $\text{Verarbeitungszeit(A)} / \text{Prozesszeit(A)} = (5\text{ms} / 50\text{ms}) = 0.1 = 10\%$
2. Auslastung(B) =  $\text{Verarbeitungszeit(B)} / \text{Prozesszeit(B)} = (5\text{ms} / 20\text{ms}) = 0.25 = 25\%$
3. Auslastung(C) =  $\text{Verarbeitungszeit(C)} / \text{Prozesszeit(C)} = (10\text{ms} / 25\text{ms}) = 0.4 = 40\%$
4. Gesamt =  $(0.1 + 0.25 + 0.4) \text{ ms} = 0.75\text{ms} = 75\%$

7. Wie groß sind im Beispiel die maximalen Reaktionszeiten der Tasks bei Einsatz eines prioritätengesteuerten Scheduling (Rechnung!)?

1. TaskA: 10ms
2. TaskB: 5ms
3. TaskC: 20ms

# Index der Kommentare

---

- 2.1 Das Diagramm ist super!  
Rechnerische Aufstellung fehlt aber.
- 2.2 Die Auslastung mit diesem Verfahren ist identisch zum vorherigen Verfahren, da sich bis auf den Wechsellpuffer nichts am grundlegenden Konzept ändert und dieser keinen signifikanten Mehraufwand bringt.
- 3.1 Gleichverteilt bedeutet, wir nehmen hier den Mittelwert der Tasks mit variierenden Werten. Wir kommen somit auf ca. 87%.
- 4.1 Generell gilt: Die Task mit der kürzesten Rechenzeit UND der geringsten Prozesszeit bekommt die höchste Priorität.