

Contents

1	Formeln	1
2	Zeiten zeug	3
3	Petrie Netze	4
4	Datenfluss Diagram	4
5	Struktugram	4
6	Zeug	4
6.1	Harte und Weiche Realzeit	4
6.2	Bits	5
7	Realzeitnachweise	5
7.1	Prioritätengesteuert	5
7.2	Deadline Scheduling	9

1 Formeln

Realzeitbedingungen:

- 1. Realzeitbedingung:

$$\rho_{max,ges} = \sum_{j=1}^n \frac{t_{Emax,i}}{t_{Pin,i}} \leq c \text{ mit } c = \text{Anzahl Rechnerkerne}$$

- 2. Realzeitbedingung: Für alle Rechenzeitanforderungen i muss gelten:

$$t_{Dmin,j} \leq t_{Rmin,j} \leq t_{Rmax,j} \leq t_{Dmax,j}$$

$$t_{Pmin,i} = \text{minimal} \Rightarrow t_{max,i} = \frac{1}{t_{Pmin,i}}$$

$$t_{Pmax,i} = \text{maximal} \Leftarrow \text{uninteressant}$$

$$t_{Dmin,i} = \text{minimal zulässige Reaktionszeit}$$

$$t_{Dmax,i} = \text{maximal zulässige Reaktionszeit}$$

- Ausführungszeit (Executiontime) = Rechenzeit für eine RZ-Anforderung
(ohne Warte oder Schlafzeiten)

- WCET $t_{Emax,i}$ -> Erfahrung oder Messen Worstcase

- BCET $t_{Emin,i} = 0$ Bestcase

$T_{Rmax,i}$ = maximale Reaktionszeit

$T_{Rmin,i}$ = minimale Reaktionszeit

$T_{R,i} = t_{W,i} + t_{E,i}$ wobei $t_{W,i}$ Summe aller Wartezeiten

- Latenzzeit $t_{L,i}$ - Interrup Latenzzeit - Tasklatenzzeit

$\rho_i = \frac{t_{E,i}}{t_{P,i}}$ Auslastung der RZ-Anforderung i

$\rho_{max,i} = \frac{t_{Emax,i}}{t_{Pin,i}}$ Worstcase, max. Auslastung

1. RT Bedingung

$$\rho_{max,ges} = \sum_{j=1}^n \frac{t_{Emax,i}}{t_{Pin,i}} \leq c$$

j = für alle RZ-Anforderungen, c = Anzahl der Rechnerkerne

Technischer prozess:

- $t_{P,i}$ = Prozesszeit, zeitlicher Abstand zwischen zwei RT-Anforderungen i

$t_{Pmin,i}$

- $t_{Dmin,i}$ = minimal zulässige Reaktionszeit

- $t_{Dmax,i}$ = maximal zulässige Reaktionszeit

- $t_{Ph,i}$ = Phase, zeitlicher Abstand zwischen zwei unterschiedlicher Ereignisse

Rechenprozesse:

- $t_{Emin,i}$ = minimale Ausführungszeit BCET

- $t_{Emax,i}$ = maximale Ausführungszeit WCET

- $t_{Rmin,i}$ = minimale Reaktionszeit

- $t_{Rmax,i}$ = maximale Reaktionszeit

↳ Zeitlicher Abstand zwischen dem Eintreffen einer RT-Anforderung i und dem Ende der Bearbeitung

- $t_{W,i}$ = Wartezeit, Summe der Zeiten, in der eine Codesequenz arbeiten könnte, aber nicht dran kommt.

Systemsoftware:

- $t_{L,i}$ = Latenzzeit, zeitlicher Abstand zwischen dem Eintreffen einer RT-Anforderung i und dem Start der Bearbeitung
- Schedulingverfahren

1. RT Bedingung

$$\rho_{max,ges} = \sum_{j=1}^n \frac{t_{Emax,j}}{t_{Pmin,j}} \leq c$$

j = für alle RZ-Anforderungen, c = Anzahl der Rechnerkerne

2. RT Bedingung

Für alle RZ-Anforderungen j muss gelten:

$$t_{Dmin,j} \leq t_{Rmin,j} \leq t_{Rmax,j} \leq t_{Dmax,j}$$

$$\text{Utilization } u = \sum_{j=1}^n \frac{t_{Emax,j}}{\min(t_{Dmax,j}, t_{Pmin,j})}$$

2 Zeiten zeug

```
struct timespec {
    time_t    tv_sec;           /* seconds */
    long      tv_nsec;         /* nanoseconds */
};
```

- CLOCK_REALTIME
Systemweite realzeit Uhr. Diese Uhr zu setzten erfordert Root Rechte.
- CLOCK_MONOTONIC
Kann nicht gesetzt werden. Gibt die vergangene Zeit ab einem unbestimmten Zeitpunkt an.

```
#include <time.h>
main(int argc, char **argv)
{
    struct timespec start, end;
    clock_gettime(CLOCK_MONOTONIC, &start); /* mark start time */
    sleep(1); /* do stuff */
    clock_gettime(CLOCK_MONOTONIC, &end); /* mark the end time */
    int diff = diff (end.tv_sec - start.tv_sec) + end.tv_nsec - st
```

}

- Absolute Zeit: Zeit die überall gleich schnell ist
- Realtive Zeit: Zeit ist abhängig von der Geschwindigkeit und der Gravitation.

```
#include <sys/time.h>
```

```
void timeradd(struct timeval *a, struct timeval *b, struct timeval *re
```

```
void timersub(struct timeval *a, struct timeval *b,      struct timeval
```

```
void timerclear(struct timeval *tvp);
```

```
int timerisset(struct timeval *tvp);
```

```
int timercmp(struct timeval *a, struct timeval *b, CMP);
```

3 Petrie Netze

->siehe Klausur

4 Datenfluss Diagram

->siehe Klausur

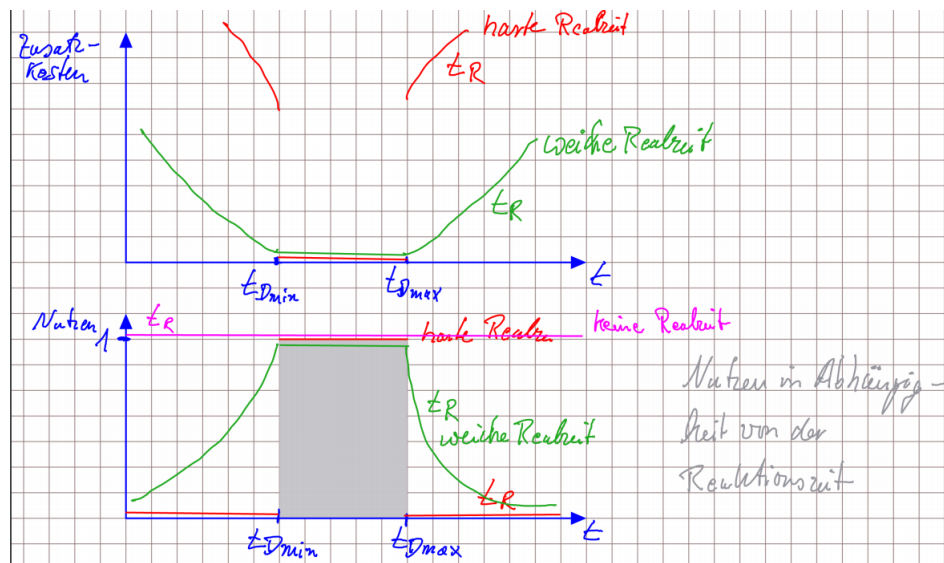
5 Struktugram

->siehe Klausur

6 Zeug

6.1 Harte und Weiche Realzeit

Harte und weiche Realzeit/Echtzeit



6.2 Bits

Oberen 4 Bits von 10101101 in Dezimal.

1010 allein stehen durch 4 mal bitshift rechts => 1010 = 0xA = 10

Bitmaskieren:

- AND 1010 & 0111 = 0010
- OR 0101 | 0011 = 0111
- XOR 0110 ^ 1011 = 1101

7 Realzeitnachweise

7.1 Prioritätengesteuert

1. Schritt: Anforderungen und zeitliche Parameter der Lösung zusammenstellen.

2. Schritt: Utilization überprüfen

Realzeitbedingungen werden grundsätzlich eingehalten, falls

$$u \leq n \times (2^{\frac{1}{n}} - 1) \text{ wobei } n = \text{Anzahl der Threads/Tasks, RT-Anforderungen}$$

$$n = 1 \Rightarrow u \leq 100\%$$

$$n = 2 \Rightarrow u \leq 82.8\%$$

$$n = 3 \Rightarrow u \leq 78\%$$

$$n \rightarrow \infty \Rightarrow u \leq 69.3\%$$

3. Schritt: 2. Realzeitbedingung überprüfen (falls $u \leq u \leq n \times (2^{\frac{1}{n}} - 1)$)

Problem: Bestimmung von t_{Rmax}

Idee: Die t_{Emax} der höheren oder gleich Prioren Tasks werden aufsummiert über die Zeit (= Arbeit für den Rechner zum Zeitpunkt t). Es wird der Zeitpunkt gesucht, an dem die benötigte Zeit nicht mehr größer als die zur Verfügung gestellte Rechenzeit ist.

$$t_{c,p}(t) = \sum_{j \in J} \left\lceil \frac{t}{t_{Pmin,j}} \right\rceil \times t_{Emax,j}$$

J = alle höher oder gleich Prioren Jobs

p = Priorität

Die zur Verfügung stehende Rechenzeit ergibt sich zu

$$t_{available}(t) = t$$

Gesucht ist damit die Lösung der Gleichung

$$t_c(t) = t$$

Diese Gleichung lässt sich iterativ lösen:

$$\text{Startwert: } t_p^{(1)} \leftarrow \text{Iterationsschritt} = \sum_{j \in J} t_{Emax,j}$$

$$\text{Iteration: } t_p^{(l+1)} = t_{c,p}(t_p^{(l)}) = \sum \left\lceil \frac{t_p^{(l)}}{t_{Pmin,j}} \right\rceil \times t_{Emax,j}$$

$$\text{Abbruch: } t_p^{(l)} == t_p^{(l+1)} \quad t_{Rmax,p} = t_p^{(l)}$$

Table 1: Beispiel:

RZ-Anf	t_{Pmin}	t_{Dmin}	t_{Dmax}	t_{Ph}	t_{Emin}	t_{Emax}	t_{Rmin}	t_{Rmax}	Prio
A	30ms	0ms	20ms	0ms	2ms	10ms	2ms		1
B	45ms	0ms	45ms	0ms	3ms	15ms	3ms		2
C	60ms	0ms	60ms	0ms	4ms	15ms	4ms		3

$$t_{Emin} = t_{Rmin}$$

Berechnung der Utilization u:

$$u = \sum_{j=1}^n \frac{t_{Emax,j}}{\min(t_{Dmax,j}, t_{Pmin,j})} = \frac{10ms}{20ms} + \frac{15ms}{45ms} + \frac{15ms}{60ms} = 1.083$$

$$\rho_{max,ges} = \sum_{j=1}^n \frac{t_{Emax,j}}{t_{Pmin,j}} \leq c \Rightarrow \frac{10ms}{30ms} + \frac{15ms}{45ms} + \frac{15ms}{60ms} = 0.917 \leq 1$$

Berechnung der Schranke für n = 3

$$s = n \times (2^{\frac{1}{n}}) = 0.78$$

Bedingung $u \leq 0.78$ ist nicht erfüllt -> weiter rechnen

Bestimmung von $t_{Rmax,1}$ für die Jobs der Priorität 1

$$1. \text{ Aufstellen von } t_{c,p}(t) = \sum_{j \in J} \left\lceil \frac{t}{t_{Pmin,j}} \right\rceil \times t_{Emax,j} = t_{c,1}(t) = \left\lceil \frac{t}{30ms} \right\rceil \times 10ms$$

$$2. \text{ Startwert: } t_1^{(1)} = 10ms (= t_{Emax,1})$$

$$3. \text{ Iteration: } t_1^{(2)} = t_{c,1}(10ms) = \left\lceil \frac{10ms}{30ms} \right\rceil \times 10ms = 10ms$$

$$\text{Abbruch, da: } t_1^{(1)} = t_1^{(2)} \Rightarrow t_{Rmax,1} = 10ms$$

Priorität 2

$$1. \text{ Aufstellen von } t_{c,2}(t) = \underbrace{\left\lceil \frac{t}{30ms} \right\rceil \times 10ms}_A + \underbrace{\left\lceil \frac{t}{45ms} \right\rceil \times 15ms}_B$$

$$2. \text{ Startwert: } t_2^{(1)} = t_{Emax,A} + t_{Emax,B} = 10ms + 15ms = 25ms$$

$$3. \text{ Iteration: } t_2^{(2)} = t_{c,2}(25ms) = \left\lceil \frac{25ms}{30ms} \right\rceil \times 10ms + \left\lceil \frac{25ms}{45ms} \right\rceil \times 15ms = 25ms$$

$$\text{Abbruch, da: } t_2^{(1)} = t_2^{(2)} \Rightarrow t_{Rmax,2} = 25ms$$

Priorität 3

$$1. \text{ Aufstellen von } t_{c,3}(t) = \underbrace{\left\lceil \frac{t}{30ms} \right\rceil \times 10ms}_A + \underbrace{\left\lceil \frac{t}{45ms} \right\rceil \times 15ms}_B + \underbrace{\left\lceil \frac{t}{60ms} \right\rceil \times 15ms}_C$$

2. Startwert:

$$t_3^{(1)} = t_{Emax,A} + t_{Emax,B} + t_{Emax,C} = 10ms + 15ms + 15ms = 40ms$$

3. Iteration:

$$t_3^{(2)} = t_{c,3}(40ms) = \left\lceil \frac{40ms}{30ms} \right\rceil \times 10ms + \left\lceil \frac{40ms}{45ms} \right\rceil \times 15ms + \left\lceil \frac{40ms}{60ms} \right\rceil \times 15ms = 50ms$$

$$t_3^{(3)} = t_{c,3}(50ms) = \left\lceil \frac{50ms}{30ms} \right\rceil \times 10ms + \left\lceil \frac{50ms}{45ms} \right\rceil \times 15ms + \left\lceil \frac{50ms}{60ms} \right\rceil \times 15ms = 65ms$$

$$t_3^{(4)} = t_{c,3}(65ms) = \left\lceil \frac{65ms}{30ms} \right\rceil \times 10ms + \left\lceil \frac{65ms}{45ms} \right\rceil \times 15ms + \left\lceil \frac{65ms}{60ms} \right\rceil \times 15ms = 90ms$$

$$t_3^{(5)} = t_{c,3}(90ms) = \left\lceil \frac{90ms}{30ms} \right\rceil \times 10ms + \left\lceil \frac{90ms}{45ms} \right\rceil \times 15ms + \left\lceil \frac{90ms}{60ms} \right\rceil \times 15ms = 90ms$$

Abbruch, da: $t_3^{(4)} == t_3^{(5)} \Rightarrow t_{Rmax,3} = 90ms$

Überprüfen der 2.RT-Bedingung

	$t_{Dmin,j}$	$t_{Rmin,j}$	$t_{Rmax,j}$	$t_{Dmax,j}$
Prio 1	0ms ≤	2ms ≤	10ms ≤	20ms
Prio 2	0ms ≤	3ms ≤	25ms ≤	45ms
Prio 3	0ms ≤	4ms ≤	90ms ≤	60ms

Jobs der Priorität 3 (Task C) werden unter Umständen nicht schritthal-
tend abgearbeitet.

7.2 Deadline Scheduling

Table 2: Beispiel:

RZ-Anf	t_{Pmin}	t_{Dmin}	t_{Dmax}	t_{Ph}	t_{Emin}	t_{Emax}	t_{Rmin}	t_{Rmax}	Prio
A	30ms	0ms	20ms	0ms	2ms	10ms	2ms		1
B	45ms	0ms	45ms	0ms	3ms	15ms	3ms		2
C	60ms	0ms	60ms	10ms	4ms	15ms	4ms		3

2. Schritt: Auslastungsbedingung überprüfen

Grenze $S = 1$ <- auf Singelcore

$$\rho_{max,ges} = \sum_{i=1}^n \frac{t_{Emax,i}}{t_{Pmin,i}} \leq c \Rightarrow \frac{10ms}{30ms} + \frac{15ms}{45ms} + \frac{15ms}{60ms} = 0.917 \leq 1$$

3. Schritt: Utilization überprüfen

$$u = \sum_{i=1}^n \frac{t_{Emax,i}}{\min(t_{Dmax,i}, t_{Pmin,i})} \leq 1 \Rightarrow \sum_{i=1}^n \frac{10}{20} + \frac{15}{45} + \frac{15}{60} = 1.083$$

weiterrechnen, da $u > s$.

4. RT-Bedingung prüfen :

$$\begin{aligned}
 t_c(I) &= \sum_{i=1}^n \left\lfloor \frac{I + t_{Pmin,i} - t_{Dmax,i} - t_{Ph,i}}{t_{Pmin,i}} \right\rfloor \times t_{Emax,i} \\
 &= \left\lfloor \frac{I + 30ms - 20ms - 0ms}{30ms} \right\rfloor \times 10ms \text{ A} \\
 &= \left\lfloor \frac{I + 45ms - 45ms - 0ms}{45ms} \right\rfloor \times 15ms \text{ B} \\
 &= \left\lfloor \frac{I + 60ms - 60ms - 10ms}{60ms} \right\rfloor \times 15ms \text{ C} \\
 &= \left\lfloor \frac{I + 10ms}{30ms} \right\rfloor \times 10ms + \left\lfloor \frac{I}{45ms} \right\rfloor \times 15ms + \left\lfloor \frac{I - 10ms}{60ms} \right\rfloor \times 15ms
 \end{aligned}$$

5. :

$$0 \leq JkgV(30ms, 45ms, 60ms) + \max(0ms, 0ms, 10ms)$$

$$0 \leq J \leq 180ms + 10ms$$

$$0 \leq J \leq 190ms$$

6. :

Für Term A, B und C alle J bestimmen damit Term Ganzzahlig wird. J muss unter 190ms bleiben.

$$\left\lfloor \frac{I + 10ms}{30ms} \right\rfloor = n \text{ für } n = 1, 2, 3, \dots$$

$$I = n * 30ms - 10ms \text{ für } n = 1, 2, 3, \dots$$

$$I_A = \{20ms, 50ms, 80ms, 110ms, 140ms, 170ms\}$$

$$\left\lfloor \frac{I}{45ms} \right\rfloor = n \text{ für } n = 1, 2, 3, \dots$$

$$I = n * 45ms \text{ für } n = 1, 2, 3, \dots$$

$$I_B = \{45ms, 90ms, 135ms, 180ms\}$$

$$\left\lfloor \frac{I + 10ms}{60ms} \right\rfloor = n \text{ für } n = 1, 2, 3, \dots$$

$$I = n * 60ms + 10ms \text{ für } n = 1, 2, 3, \dots$$

$$I_C = \{70ms, 130ms\}$$

$$I_g = \{ \underbrace{20}_{+10}, \underbrace{45}_{+15}, \underbrace{50}_{+10}, \underbrace{70}_{+15}, \underbrace{80}_{+10}, \underbrace{90}_{+15}, \underbrace{110}_{+10}, \underbrace{130}_{+15}, \underbrace{135}_{+15}, \underbrace{140}_{+10}, \underbrace{170}_{+10}, \underbrace{180}_{+15} \}$$

$$t_c(20ms) = \left\lfloor \frac{I + 10ms}{30ms} \right\rfloor \times 10ms + \left\lfloor \frac{I}{45ms} \right\rfloor \times 15ms + \left\lfloor \frac{I - 10ms}{60ms} \right\rfloor \times 15ms$$

$$= \left\lfloor \frac{20ms + 10ms}{30ms} \right\rfloor \times 10ms + \left\lfloor \frac{20ms}{45ms} \right\rfloor \times 15ms + \left\lfloor \frac{20ms - 10ms}{60ms} \right\rfloor \times 15ms$$

$$= 10ms$$

$$I \geq t_c(I)$$

$$t_c(45ms) = 25ms\checkmark$$

$$t_c(50ms) = 35ms\checkmark$$

$$t_c(70ms) = 50ms\checkmark$$

$$t_c(80ms) = 60ms\checkmark$$

$$t_c(90ms) = 75ms\checkmark$$

$$t_c(100ms) = 85ms\checkmark$$

$$t_c(130ms) = 100ms\checkmark$$

$$t_c(135ms) = 115ms\checkmark$$

$$t_c(140ms) = 125ms\checkmark$$

$$t_c(170ms) = 135ms\checkmark$$

$$t_c(180ms) = 150ms\checkmark$$

Überprüfen der unteren Schranken:

Table 3: Beispiel:

RZ-Anf	$t_{Dmin,i}$	$t_{Rmin,i}$
A	0ms \leq	2ms \checkmark
B	0ms \leq	3ms \checkmark
C	0ms \leq	4ms \checkmark

Alle Bedingungen sind erfüllt. EDF klappt also.