# **A5**

Prioritäten: T1: 2 (20, 5, 20, Φ1)  
 T2: 3 (40, 10, 40, Φ2)  
 T3: 2 (20, 3, 20, Φ3)  
 T4: 1 (10, 2, 10, Φ4)

1. - CPU-Auslastung  
   - Zeitbedarfsanalyse  
   - Antwortzeitanalyse  
   - Simulation

Hier: ANTWORTZEITANALYSE

Pr. 1: T4: ω4(10) = e4 = 2 ≤ 10 ZULÄSSIG

Pr. 2: T1: ω1(10) = e1 + e3 + e4 = 10 ≤ 10 ZULÄSSIG

T3: ω3(10) = e3 + e1 + e4 = 10 ≤ 10 ZULÄSSIG

Pr. 3: T2: ω2(10) = e2 + e4 + e1 + e3 = 20 > 10 **NICHT ZULÄSSIG**

ω2(20) = e2 + 2e4 + e1 + e3 = 22 > 20 **NICHT ZULÄSSIG**

ω2(30) = e2 + 3e4 + 2e1 + 2e3 = 32 > 30 **NICHT ZULÄSSIG**

ω2(40) = e2 + 4e4 + 2e1 + 2e3 = 34 ≤ 40 ZULÄSSIG

* PLANBAR

1. Hyperperiode:  
   - kgV(10, 20, 40) = 40

Rahmenlänge:  
- f ≥ max(ei) -> f ≥ 10  
- fmöglich = {10, 20, 40}  
- T4: f ≤ p4 -> fmöglich = {10}

f = 10: 20 – ggT(10, 10) ≤ 10 -> 10 ≤ 10 ZULÄSSIG

T1: f ≤ p1 -> fmöglich = {10, 20}

f = 10: 20 – ggT(20, 10) ≤ 20 -> 10 ≤ 20 ZULÄSSIG

T3: f ≤ p3 -> fmöglich = {10, 20}

f = 10: 20 – ggT(20, 10) ≤ 20 -> 10 ≤ 20 ZULÄSSIG

T2: f ≤ p2 -> fmöglich = {10, 20, 40}

f = 10: 20 – ggT(40, 10) ≤ 40 -> 10 ≤ 40 ZULÄSSIG

**e\_i sollte kumulativ sein!!**

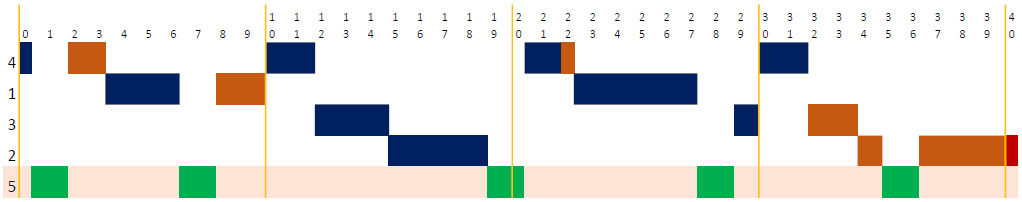
**-> Neues Aufgabensystem:** T1: (20, 5, 20, Φ1)  
 T2: (40, 10, 40, Φ2)  
 T3: (20, 3, 20, Φ3)  
 T4: (10, 2, 10, Φ4)

Hyperperiode = 40, Rahmenlänge = 10

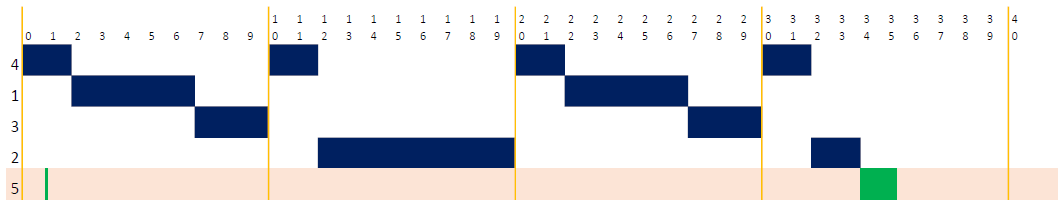
1. Innerhalb eines Rahmens ist keine Verdrängung möglich, da Ausführung der Aufträge, sowie Ereignisbehandlung, nur an Rahmengrenze planbar bzw. kontrollierbar. Anwendungsentwickler muss ggf., wie z.B. hier, Aufträge aufteilen um diese ins Ablaufplan zu kriegen.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

1. **Sollte nicht unterbrochen sein!! (wegen Fehler oben)  
     
     
     
     
   -> Neues Aufgabensystem:** T1: (20, 5, 20, 2)  
    T2: (40, 10, 40, 12)  
    T3: (20, 3, 20, 7)  
    T4: (10, 2, 10, 0)
3. Unterbrecherbetrieb:

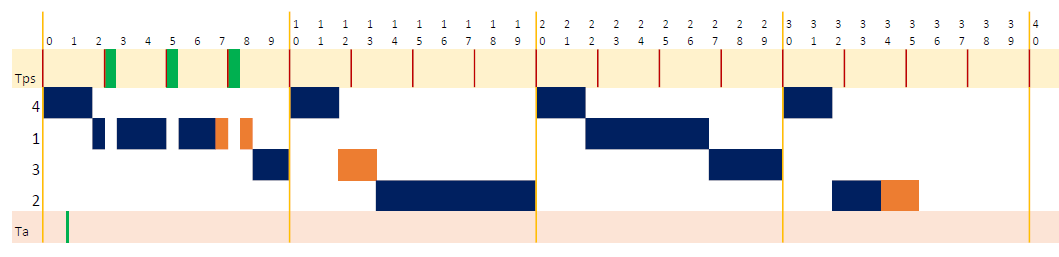
T5: Antwortzeit = Ausführungszeit  
T4: Antwortzeit = e4 + e5, Termineinhaltung möglich, da höchste Priorität  
T1: Antwortzeit = e1 + e5 + e4, Terminüberwachung möglich, da p=20  
T3: Antwortzeit = e3 + e1 + e5 + e4, Terminüberwachung möglich, da p=20  
T2: Termin kann verpasst werden, da geringste Priorität

Hintergrundbetrieb:



T1 – T4: Antwortzeit = Ausführungszeit + Ausführungszeit Aufträge mit höherer Priorität  
T5: Wird erst ausgeführt, wenn keine periodische Aufträge zur Ausführung anstehen (hier Antwortzeit=32,5)

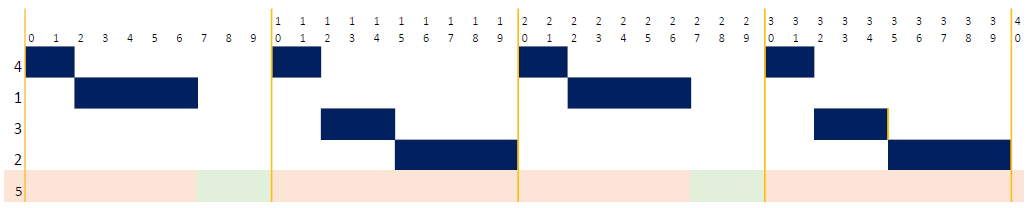
Periodischer Zusteller: (mit pPS = 2,5 und ePS = 0,5)



T1 – T4: Werden ggf. vom Zusteller unterbrochen (auch mehrmals). Antwortzeit = Ausführungszeit + Ausführungszeit Aufträge höherer Priorität + x \* ePS (x = Anzahl Unterbrechungen)  
T5: Antwortzeit = pPS \* (e5 / ePS) + tnächste Zustellerperiode – t Auslösung

**p\_ps > Rahmenlaenge (weil sonst nicht steuerbar)!!**

1. Sinnvoll mit periodischem Zusteller, da Antwortzeit nicht zu lang im Vergleich mit Hintergrundbetrieb und Risiko Termine zu verpassen geringer als bei Unterbrecherbetrieb.



**Schlupf immer ganz vorne!!**

T5 kann innerhalb der grünmarkierten Bereiche eingeplant werden. Grundsätzlich kann man nicht mehr Schlupf erzeugen, aber man kann ihn auf der Hyperperiode aufteilen, damit man öfter Zeit für die aperiodische Aufgabe hat.

1. Ablaufpläne eignen sich für Slack-Stealing gut, weil man vom Anfang an weiß, wann die CPU belegt ist, und wann nicht (bei Schlüpfe).