

Uitwerking van opgaven 1, 2, 4, 5, 6 en 7 uit hoofdstuk 11.

van Real-Time Systems and Programming Languages (Fourth Edition).

11.1 Het Rate Monotonic scheduling algoritme geeft de volgende prioriteiten (hoe lager de periodetijd hoe hoger de prioriteit).

taak	T	C	P
P	3	1	3
Q	6	2	2
S	18	5	1

De hoogste prioriteit is 3 en de laagste prioriteit is 1. De taken worden als volgt uitgevoerd:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
P	Q	P	S1	P	Q	P	S2	P	Q	P	S3	-					

De taak S moet gesplitst worden in 3 delen S1, S2 en S3. De uitvoering is

taak	T	C	P
S1	18	2	1
S2	18	2	1
S3	18	1	1

De minor cycle is nu 3. We hebben dus een timer nodig die elke 3 een signal geeft. De code voor de cyclic executive is dan als volgt:

```
while (1) {
    wait_signal(...); P(); Q();
    wait_signal(...); P(); S1();
    wait_signal(...); P(); Q();
    wait_signal(...); P(); S2();
    wait_signal(...); P(); Q();
    wait_signal(...); P(); S3();
}
```

11.2 De volgende taken zijn gegeven:

taak	T	C	Importance
P	100	30	1
Q	5	1	3
S	25	5	2

(1) Als de prioriteit gebaseerd wordt op de belangrijkheid (importance) dan worden de prioriteiten als volgt toegekend:

taak	T	C	P
P	100	30	3
Q	5	1	1
S	25	5	2

De hoogste prioriteit is 3 en de laagste prioriteit is 1. De taak P zal dan als eerste worden gestart en niet worden onderbroken (P heeft de hoogste prioriteit). Als P is afgelopen (na 30) dan hebben Q en S hun deadline al gemist!

(2) De processor utilization (bezetting) is: $U = \sum_{i=1}^3 \frac{C_i}{T_i} = \frac{30}{100} + \frac{1}{5} + \frac{5}{25} = \frac{70}{100} = 0.70$. We weten dat 3 taken

scheduable zijn als $U \equiv \sum_{i=1}^3 \frac{C_i}{T_i} \leq N(2^{1/3} - 1) = 0.78$. Er moet dus wel een schedule zijn waarbij alle deadlines worden gehaald omdat $0.70 < 0.78$.

(3) Omdat rate monotonic scheduling een optimal schedule levert moet RMS dus voldoen.

taak	T	C	P
P	100	30	1
Q	5	1	3
S	25	5	2

De hoogste prioriteit is 3 en de laagste prioriteit is 1.

(4) De taken worden als volgt uitgevoerd:

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
Q	S1				Q	S2	P1			Q	P2				Q	P3				Q	P4			
26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
Q	S1				Q	S2	P5			Q	P6				Q	P7				Q	P8			
51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
Q	S1				Q	S2	-			Q	-				Q	-				Q	-			
76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
Q	S1				Q	S2	-			Q	-				Q	-				Q	-			

We zien dat alle deadlines nu worden gehaald.

11.4 Uitvoering met statische prioriteit:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
w	-	-	-	-	-	-	-	E ₄				A ₄	E ₄			B ₄	-	-
x	-	-	E ₃	B ₃										AB ₃	B ₃		E ₃	-
y	-	-	-	-	-	E ₂	E ₂		E ₂	E ₂	-	-	-	-	-	-	-	-
z	E ₁	A ₁			A ₁						A ₁							E ₁

Legenda:

E _n	Running met prioriteit n
A _n	Running met A gelockt met prioriteit n
-	Niet actief = Nog niet gestart of al geëindigd.
	Preempted
	Blocked

Uitvoering met priority inheritance:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
w	-	-	-	-	-	-	-	E ₄	A ₄	A ₄		B ₄	-	-	-	-	-	-
x	-	-	E ₃	B ₃			AB ₃				B ₄		E ₃	-	-	-	-	-
y	-	-	-	-	-									E ₂	E ₂	E ₂	E ₂	-
z	E ₁	A ₁			A ₃	A ₃												E ₁

11.5 Uitvoering met immediate priority ceiling inheritance:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
w	-	-	-	-	-	-	-	-	E ₄	A ₄	E ₄	B ₄	-	-	-	-	-	-
x	-	-	-	-	E ₃	B ₄	AB ₄	B ₄					E ₃	-	-	-	-	-
y	-	-	-	-	-									E ₂	E ₂	E ₂	E ₂	-
z	E ₁	A ₄	A ₄	A ₄														E ₁

11.6 Bij het priority ceiling protocol kan de maximale blocking time als volgt worden berekend:

$$B_i = \max_{k=1}^K usage(k,i)C(k).$$

B_i = maximale blocking time voor taak i

K = aantal resources.

$usage(k,i) = 1$ als er een taak is met prioriteit lager dan P_i en een taak met prioriteit hoger of gelijk aan P_i (dit kan taak i zelf zijn) die de resource k delen.

De gegevens in tabelvorm:

Resource	$C(k)$
R1	50
R2	150
R3	75
R4	300
R5	250
R6	175

Taak	P_i	Gebruikt resources:
a	5	R3
b	4	R1, R2
c	3	R3, R4, R5
d	2	R1, R5, R6
e	1	R2, R6

We kunnen nu $usage(k,i)$ voor elke combinatie van k (resource) en i (taak) berekenen:

	R1	R2	R3	R4	R5	R6
a	0	0	1	0	0	0
b	1	1	1	0	0	0
c	1	1	0	0	1	0
d	0	1	0	0	0	1
e	0	0	0	0	0	0

Het berekenen van de maximale blocking time voor elke taak is nu eenvoudig:

$$B_a = \max\{C(R3)\} = \max\{75\} = 75$$

$$B_b = \max\{C(R1), C(R2), C(R3)\} = \max\{50, 150, 75\} = 150$$

$$B_c = \max\{C(R1), C(R2), C(R5)\} = \max\{50, 150, 250\} = 250$$

$$B_d = \max\{C(R2), C(R6)\} = \max\{150, 175\} = 175$$

$$B_e = \max\{\} = 0$$

Extra opgave: bereken de ceiling van elke resource.

Resource	Ceiling
R1	4
R2	4
R3	5
R4	3
R5	3
R6	2

11.7 De gegevens in tabelvorm:

taak	T	C
a	50	10
b	40	10
c	30	9

De processor utilization (bezetting) is: $U = \sum_{i=1}^3 \frac{C_i}{T_i} = \frac{10}{50} + \frac{10}{40} + \frac{9}{30} = \frac{75}{100} = 0.75$. We weten dat 3 taken

scheduable zijn als $U \equiv \sum_{i=1}^3 \frac{C_i}{T_i} \leq N(2^{1/3} - 1) = 0.78$. Er moet dus wel een schedule zijn waarbij alle deadlines worden gehaald omdat $0.75 < 0.78$. Omdat rate monotonic scheduling een optimal schedule levert moet RMS dus voldoen.

taak	T	C	P
a	50	10	1
b	40	10	2
c	30	9	3

We kunnen de response tijden als volgt berekenen (response time analyse):

$$w_i^{n+1} = C_i + \sum_{j \in hp(i)} \left\lceil \frac{w_i^n}{T_j} \right\rceil C_j$$

We berekenen de response tijden voor elke taak in volgorde van prioriteit (van hoog naar laag):

$$R_c = C_c = 9$$

$$R_b : w_b^1 = 10 + \left\lceil \frac{10}{30} \right\rceil \cdot 9 = 19 \rightarrow w_b^2 = 10 + \left\lceil \frac{19}{30} \right\rceil \cdot 9 = 19 \rightarrow R_b = 19$$

$$R_a : w_a^1 = 10 + \left\lceil \frac{10}{40} \right\rceil \cdot 10 + \left\lceil \frac{10}{30} \right\rceil \cdot 9 = 29 \rightarrow w_a^2 = 10 + \left\lceil \frac{29}{40} \right\rceil \cdot 10 + \left\lceil \frac{29}{30} \right\rceil \cdot 9 = 29 \rightarrow R_a = 29$$

Extra opgave: maak dezelfde som nogmaals maar nu met $C_c = 10$.

De processor utilization (bezetting) is: $U = \sum_{i=1}^3 \frac{C_i}{T_i} = \frac{10}{50} + \frac{10}{40} + \frac{10}{30} = 0.783$. We weten dat 3 taken

scheduable zijn als $U \equiv \sum_{i=1}^3 \frac{C_i}{T_i} \leq N(2^{1/3} - 1) = 0.780$. Het is dus niet zeker of er een schedule is waarbij

alle deadlines worden gehaald omdat $0.783 > 0.780$. Omdat rate monotonic scheduling een optimal schedule levert zou RMS kunnen voldoen.

We kunnen de response tijden als volgt berekenen (response time analyse):

$$R_i = C_i + \sum_{j \in hp(i)} \left\lceil \frac{R_i}{T_j} \right\rceil C_j \text{ op te lossen met de recurrente betrekking: } w_i^{n+1} = C_i + \sum_{j \in hp(i)} \left\lceil \frac{w_i^n}{T_j} \right\rceil C_j$$

We berekenen de response tijden voor elke taak in volgorde van prioriteit (van hoog naar laag):

$$R_c = C_c = 10$$

$$R_b : w_b^1 = 10 + \left\lceil \frac{10}{30} \right\rceil \cdot 10 = 20 \rightarrow w_b^2 = 10 + \left\lceil \frac{20}{30} \right\rceil \cdot 10 = 20 \rightarrow R_b = 20$$

$$R_a : w_a^1 = 10 + \left\lceil \frac{10}{40} \right\rceil \cdot 10 + \left\lceil \frac{10}{30} \right\rceil \cdot 10 = 30 \rightarrow w_a^2 = 10 + \left\lceil \frac{30}{40} \right\rceil \cdot 10 + \left\lceil \frac{30}{30} \right\rceil \cdot 10 = 30 \rightarrow R_a = 30$$

Uit deze response time analyse blijkt dat alle taken hun deadline halen.

Extra opgave: maak dezelfde som nogmaals maar nu met $C_c = 12$.

De processor utilization (bezetting) is: $U = \sum_{i=1}^3 \frac{C_i}{T_i} = \frac{10}{50} + \frac{10}{40} + \frac{12}{30} = 0.85$. We weten dat 3 taken

scheduable zijn als $U \equiv \sum_{i=1}^3 \frac{C_i}{T_i} \leq N(2^{1/3} - 1) = 0.78$. Het is dus niet zeker of er een schedule is waarbij alle deadlines worden gehaald omdat $0.85 > 0.78$. Omdat rate monotonic scheduling een optimal schedule levert zou RMS kunnen voldoen.

We kunnen de response tijden als volgt berekenen (response time analyse):

$$R_i = C_i + \sum_{j \in hp(i)} \left\lceil \frac{R_j}{T_j} \right\rceil C_j \text{ op te lossen met de recurrente betrekking: } w_i^{n+1} = C_i + \sum_{j \in hp(i)} \left\lceil \frac{w_i^n}{T_j} \right\rceil C_j$$

We berekenen de response tijden voor elke taak in volgorde van prioriteit (van hoog naar laag):

$$R_c = C_c = 12$$

$$R_b : w_b^1 = 10 + \left\lceil \frac{10}{30} \right\rceil \cdot 12 = 22 \rightarrow w_b^2 = 10 + \left\lceil \frac{22}{30} \right\rceil \cdot 12 = 22 \rightarrow R_b = 22$$

$$R_a : w_a^1 = 10 + \left\lceil \frac{10}{40} \right\rceil \cdot 10 + \left\lceil \frac{10}{30} \right\rceil \cdot 12 = 32 \rightarrow w_a^2 = 10 + \left\lceil \frac{32}{40} \right\rceil \cdot 10 + \left\lceil \frac{32}{30} \right\rceil \cdot 12 = 44 \rightarrow$$

$$w_a^3 = 10 + \left\lceil \frac{44}{40} \right\rceil \cdot 10 + \left\lceil \frac{44}{30} \right\rceil \cdot 12 = 54 \rightarrow R_a \geq 54$$

Uit deze response time analyse blijkt dat taak a zijn deadline niet haalt.