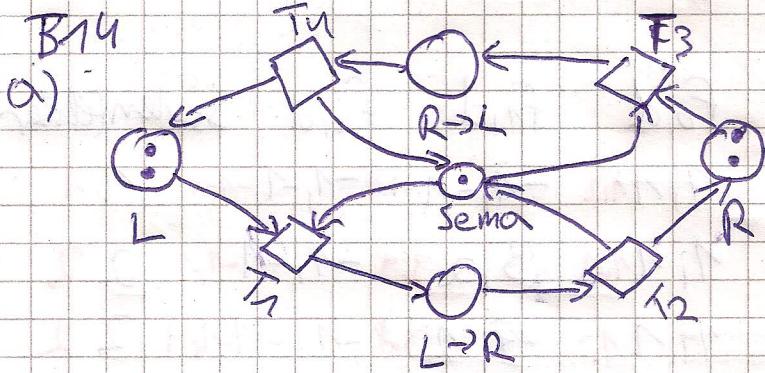


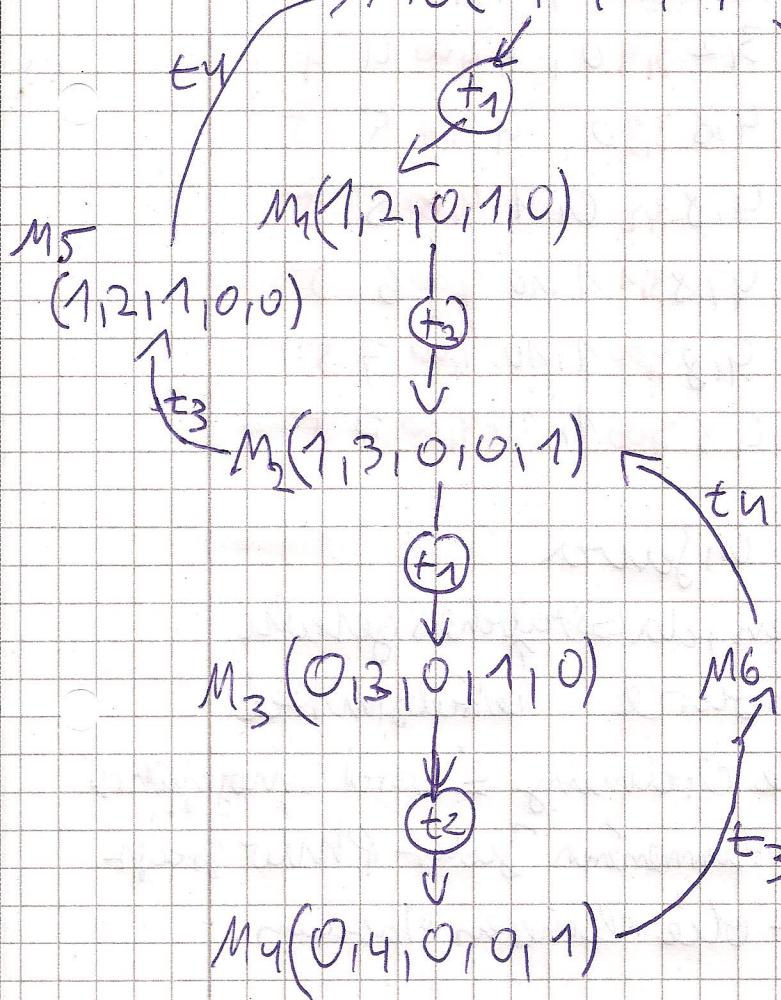
Bs Sonderklausurum



Nur 1 Auto
gleichzeitig
→ Semaphore

b) Markierung: $(L_1, R_1, R \rightarrow L_1, L \rightarrow R_1, S)$

$$\rightarrow M_0(2, 1, 0, 0, 1)$$



Analog zu t_1 auf der anderen Seite

Fair \rightarrow Begründung

Ja, weil langfristig alle Seiten gleich behandelt werden

Kunststück kann jedoch eine Seite vornelegen!

Teilweise Verklemmung ist auch ein Deadlock!

M_1

M_2
 M_3

Teilweise, da M_1 nicht erreicht werden kann

A 66 Scheduling Paging

Fifo OPT

T	Rejs	$f_{0,d}$	$f_{1,d}$	$f_{2,d}$	Seitenfehler
1	1	1, 1 1, 2	-1, -1 -1 -1 -1, -1 -1 -1	1, 1	1, 1
2	3	1, 1 1, 1	3, 2 3, 9	-1, -1 -1, 1	2, 2
3	1	1, 1 1, 2	3, 1 3, 8	-1, -1 -1, 1	2, 2
4	0	1, 1 1, 1	3, 2 3, 7	0, 1 0, 1 2	3, 3
5	1	1, 1 1, 5	3, 1 2 3, 6	0, 1 0, 1 3	3, 3
6	0	1, 1 1, 4	3, 2 3, 5	0, 1 0, 0 3	3, 3
7	2	2, 1 7 1, 3	3, 1 2 3, 4	0, 1 0, 0 4	4, 4
8	4	2, 1 7 1, 2	4, 1 8 3, 3	0, 1 4 4, 1 5	5, 5
9	4	2, 1 7 1, 1	4, 1 8 3, 2	0, 1 4 4, 0 5	5, 5
10	1	2, 1 7 1, 2	4, 1 8 3, 1	1, 1 0 4, 0 6	5, 5
11	3	3, 1 1 1, 1	4, 1 8 3, 0	1, 1 0 4, 0 7	5, 5
12	1	3, 1 1 1, 0	4, 1 8 3, 0	1, 1 0 4, 0 7	5, 5

A 67 Philosophers Problem in Java

Kritischer Bereich \rightarrow Essen, da Essstäbchen geteilte
Resource Betriebsmittel

Diskreter Ansatz Partielle Ordnung \leq wird eingeführt
Hierarchie der Durchnummierung Jeder Philo greift
zuerst die kleineren Stäbchen

Semaphore Semaphore mit "erlaube_Essen"
init(erlaube_Essen, 4)

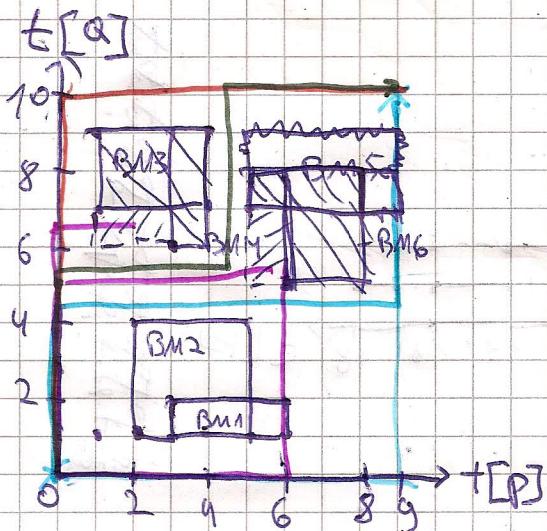
Merkmale des Readlocks:

- Mutual Exclusion
- Hold and Wait
- No Preemption
- Circular Wait \rightarrow wird aufgelöst

Get mit wait

Put mit Notify all
(wird bewusst = false)

T62



Gerade Linien, Da 1 CPU-System
Zeitlich korrekt arbeitet

Schwäige Linie geht nicht, da
sowas impliziert, dass 2 Prozesse
arbeiten, was bei einem 1 Prozess System
nicht erlaubt ist

Kritischen (unsicherer Pfad)
Müssen in die kritischen Bereiche

geodreieck Metnahmen

Semaphore T64

a) Kritische Bereiche: Parkplatz, Rampe

b) Parkplatz-frei Zählsemaphor

Rampe

Binärsemaphor (wenn init mit 1)

c) init (Parkplatz-frei, 4)

init (Rampe, 1)

d) auto()

while(true)

wait (parkplatz-frei);

wait (rampe);

<fahr ins PP rein>;

signal (rampe);

<parken>;

wait (rampe);

<verlassen>;

signal (rampe);

signal (parkplatz-frei)

e) Jede mit LKW

(LKW { while (true) {

wait (PP_frei);

if (is LKW PP_frei) {

wait (LKW_PP_frei);

und dann analog

zur Auto() aus

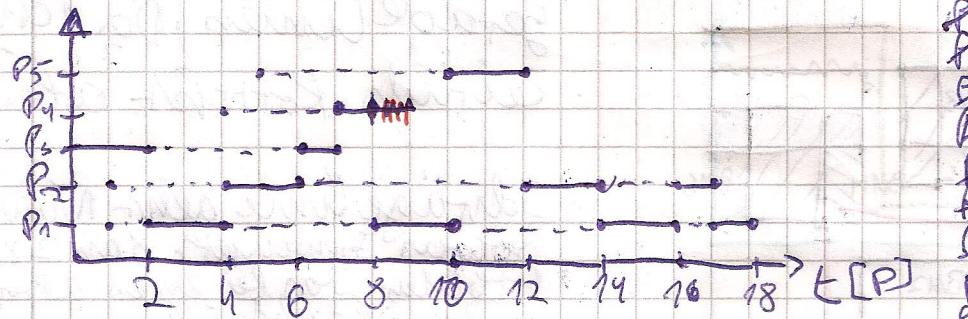
f) nur mit

LKW PP

→ Schwimmbrad
Aufgabe

T61 Preemptive Scheduling

RR Zeit-Schreibe 2



P₃
P₁
P₂
P₃
P₄
P₁
P₅
P₂
P₁
P₂
P₁
→ Fertig.