

17.
015

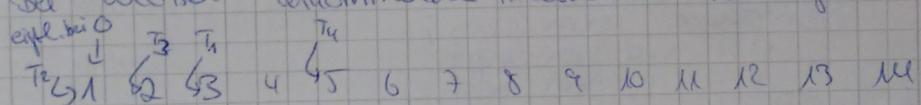
RAM & MMU SysProg Tut

(Probeklausur) 2014

- ext. Fragmentierung: es entstehen freie Bereiche zwischen zugeordneten Blöcken, die sehr klein sind (nicht nutzbar)
- inten: freier Speicher in zugeordneten Blöcken
- MBR = Master Boot Record (enthält Daten zum Starten d. Systems)
→ Partitionstabellen ist darin
- Hep: dyn. alloz. Speicher (malloc)
Stack: stat. alloz. Speicher [Stacksize: glob. Variablen]
- Eltern + Kind-Prozesse haben getrennte Adressräume (Kopie erzeugt bei Föhr.)
→ Threads haben gleichen Adressraum
- Neberläufigkeit \Rightarrow mit 1 Prozessor möglich (Parallelität nicht)
- aus dem RAM verdrängte Seiten müssen nur dann im Speicher geschrieben werden, wenn sie modifiziert werden

RR

Bei Wechsel: einkommende Prozesse vor vorhandenem in Queue



T ₁	X X X
T ₂	X X X
T ₃	X X
T ₄	X X X X

RR mit Zulässigkeit = 5 $\hat{=} FCFS$

Periodischer Scheduling

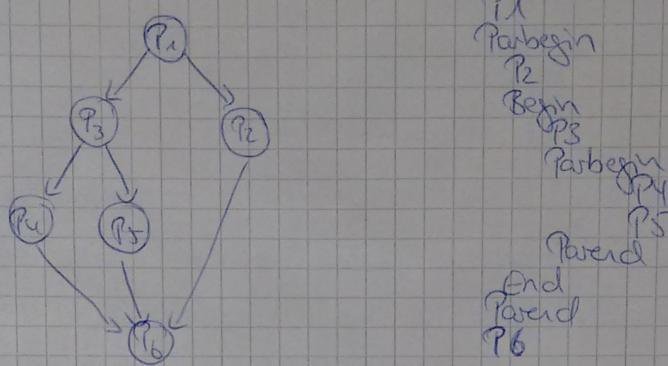
$$a) \frac{1}{6} + \frac{2}{6} + \frac{1}{6} = \frac{3}{4} \leq 1$$

} Summe aller
Bedienrest
Perioden

BRUNNEN RMS Scheduling erlaubt Verdrängung (mehrere Perioden)
Earliest Deadline First

- = 12
 - Hyperperiode: kleinste gem. Vielfaches der Perioden
 - RMS rule monotonic scheduling
 - Leerzeit: Zeit wo kein Prozess läuft

4. Prozessvorgängengraph



Semaphore

p(), v() für sperren + freigeben v. Betriebsmitteln

// Variablen

Semaphore
" sem_club;
" sem_vip;
" sem_norm;

// Nut

init(sem_fanz, 1000)

nut(sem_vip, 5)

init(sem_norm, 2)

void normale_gaudewabe (-) {

p(sem_norm)

mautel()

v(sem_norm)

← Prozess wartet bis p()
möglich

}

```

besuch-discos ( -- ) {
    P(sem-club);
    if (is-vip(person)) {
        vip-garderobe(person);
    } else {
        normal-garderobe(person);
    }
    tanzen();
    V(sem-club);
}

```

Deadlock

- wait \leftarrow hier ohne Unlock, deswegen kein Signal nicht erfolgen
- while (muss while, nie if (zg Sprawoz Wakesys))

Betriebsmittel

- physisch \rightarrow RAM
- logisch (Abstraktion v. phys. BMs) \rightarrow Fenster
- virtuell \rightarrow virt. Speicher

Bauer-Algo.

\rightarrow unsicher ~~XXXX~~, nicht zwangsläufig Deadlock

$$R = \begin{pmatrix} \text{aktuelle} \\ \text{Möglich} \end{pmatrix} \quad B = \begin{pmatrix} \text{Bereyt} \end{pmatrix} \quad \begin{matrix} \xleftarrow{\text{Pfeile v. BM}} \\ \text{zu Prozessen} \end{matrix}$$

$B_{i,j}^0$ \leftarrow Betriebsmittel mit 2 Instanzen

Second Chance - Seiktausch-Verfahren											\approx Clock-Algo	
	3	8	2	7	5	2	5	9	4	8	2	4
1	3	3	3	7	7	7	7	4	4	4	4	4
2	8	8	8	5	5	5	5	5	8	8	8	8
3	2	2	2	2	2	2	3	3	3	2	2	2
Second Chance?	1	0	1	1	1	1	1	0	1	1	0	1
BM	2	0	1	1	0	1	1	0	0	1	0	0
3	0	0	1	0	0	1	1	1	1	1	1	1

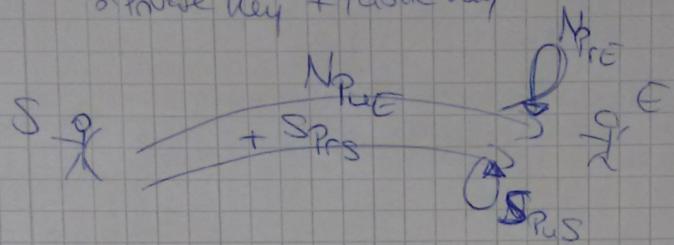
BRUNNEN

Buddy-Vorfahren

$f_{\text{re}}(0)$	(2 2 4)	(2 2 4)
$f_{\text{re}}(1)$	8	(2 2 4)

Verschlüsselung

- Private Key + Public Key



- Nachricht mit publickey d. Empfänger verschlüsseln
- Signatur mit eigenem Private Key verschlüsseln

Signierung mit MD5

verschlüsselt

→ Hashwert zu Daten mitschicken → überprüfen ob gleich um Manipulation zu entdecken