

Musterlösungen zu den Tutoraufgaben auf
Blatt 7
der Übungen zur Vorlesung
“Grundlagen Betriebssysteme und Systemsoftware

G.Groh, 1.12.2007

Das 'short-term-scheduling' unterstützt das interaktive Arbeiten am Rechner. Die Prioritäten von Prozessen, die lange auf die Ausführung von E/A-Operationen warten, werden erhöht. Dagegen werden die Prioritäten von Prozessen, die wesentliche CPU-Zeiten auf sich vereinigen, verringert. Dies bedeutet zusammenfassend, dass interaktive Prozesse den Batch-Prozessen vorgezogen werden.

In der Regel werden somit genau die Prozesse der Programme bevorzugt, mit denen der Benutzer gerade arbeitet, deshalb wirkt das System schneller.

$X = 0 \Rightarrow$ sofortiger Deadlock

$X = 1 \Rightarrow$ P D kann abgearbeitet werden $\Rightarrow a = (11221)$

\Rightarrow P A $\Rightarrow a = (21432)$

\Rightarrow P C $\Rightarrow a = (32442)$

\Rightarrow P D \Rightarrow sicherer Systemzustand

Aufgabe 5.2 Lösungsvorschlag

Suppose that process A requests the records in the order a, b, c . If process B also asks for a first, one of them will get it and the other will block. This situation is always deadlock free since the winner can now run to completion without interference. Of the four other combinations, some may lead to deadlock and some are deadlock free. The six cases are as follows:

a b c	deadlock free
a c b	deadlock free
b a c	possible deadlock
b c a	possible deadlock
c a b	possible deadlock
c b a	possible deadlock

Since four of the six may lead to deadlock, there is a $1/3$ chance of avoiding a deadlock and a $2/3$ chance of getting one.

Two-phase locking eliminates deadlocks, but introduces potential starvation. A process has to keep trying and failing to acquire all of its records. There is no upper bound on how long it may take.

Aufgabe 5.4 Lösungsvorschlag

- **Exclusiveness Condition:** Die gemeinsam benutzbaren Ressourcen können nicht parallel genutzt werden, d.h. sie sind nur exklusiv benutzbar.
- **No Preemption Condition:** Die zugeteilten/belegten Ressourcen können nicht entzogen werden, d.h. die Nutzung ist nicht unterbrechbar.
- **Hold and Wait Condition:** Prozesse belegen die schon zugeteilten Ressourcen auch dann, wenn sie auf die Zuteilung weiterer Ressourcen warten, d.h. wenn sie weitere Ressourcen anfordern.
- **Circular Waiting Condition:** Es gibt eine zyklische Kette von Prozessen, von denen jeder mindestens eine Ressource belegt, die der nächste Prozess in der Kette benötigt, d.h. zirkuläre Wartebedingung.

Unterscheidung Vermeidung, Verhinderung, Banker, Angreifen der 4 Bedingungen, Prevention, Avoidance:

- Deadlock Avoidance == Verklemmungs-Vermeidung
↔ Banker's Algorithmus
- Deadlock Prevention == Verklemmungs-Verhinderung
↔ Angreifen der 4 Deadlock Bedingungen