TECHNISCHE UNIVERSITÄT
IN DER KULTURHAUPTSTADT EUROPAS
CHEMNITZ

Professur
Betriebssysteme

8. Übung

Verteilte Betriebssysteme

Jonas Henschel

Aufgabe 1 (Gleichzeitiger Zugriff)

Ines und ihre Freunde müssen ein Referat halten. Nachdem Anna und Ines zum zweiten Mal gleichzeitig auf die Google Docs Powerpoint-Datei ihrer Gruppenpräsentation zugreifen und ihre Arbeit gegenseitig ruinieren, schlägt Hannes vor, dass jeder eine lokale Kopie der Datei machen soll und er am Ende die Ergebnisse zusammenführt. Ist das die beste Lösung? Welche anderen Lösungen fallen Ihnen ein, die Ines und ihr Team anwenden könnten?

Aufgabe 2 (Mutex)

a) Betrachten Sie das folgende Programm-Beispiel. Gehen Sie davon aus, dass die Funktion `writing_member()` nicht-verteilt aber durch zwei Threads parallel ausgeführt wird. Wie kann es im unten aufgeführten Code-Schnipsel zu Problemen kommen? Wie können diese verhindert werden?

```
double buffer[N];
int write = 0;
int read = 0;

void writing_member()
{
 // do something

 buffer[write] = result;
 write = (write + 1) % N;
 ...
}
```

b) Was ist ein Semaphor und wie funktioniert er (im nicht-verteilten Fall)?

c) Eine triviale Lösung für die Absicherung einer critical section im verteilten Fall ist ein Ressourcen Koordinator/Manager. Erläutern Sie das Prinzip. Was sind Mögliche Probleme dieser Lösung?

Aufgabe 3 (*Lamport* Broadcast)

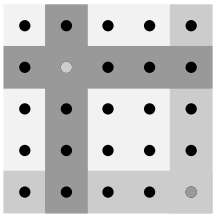
Eine Implementationsmöglichkeit für einen verteilten Mutex ist der Broadcast Algorithmus von *Lamport*. Erläutern Sie den Algorithmus für den Fall

a) dass einer von drei Beteiligten auf die critical section Zugreifen möchte.

b) dass zwei von drei Beteiligten auf die critical section Zugreifen möchten.

Aufgabe 4

Um bei verteilten Anwendungen kritische Abschnitte durch gegenseitigen Ausschluss zu schützen, bedarf es eines Algorithmus, der über alle Knoten hinweg entscheidet, welcher Knoten wann Zugriff erhält. Eine Möglichkeit sind Quoren-basierte Algorithmen. Erklären Sie in diesem Zusammenhang das folgende Bild und machen Sie ein Beispiel, wie es hier zum Deadlock kommen kann und wie der Algorithmus verändert werden muss um dies zu vermeiden.

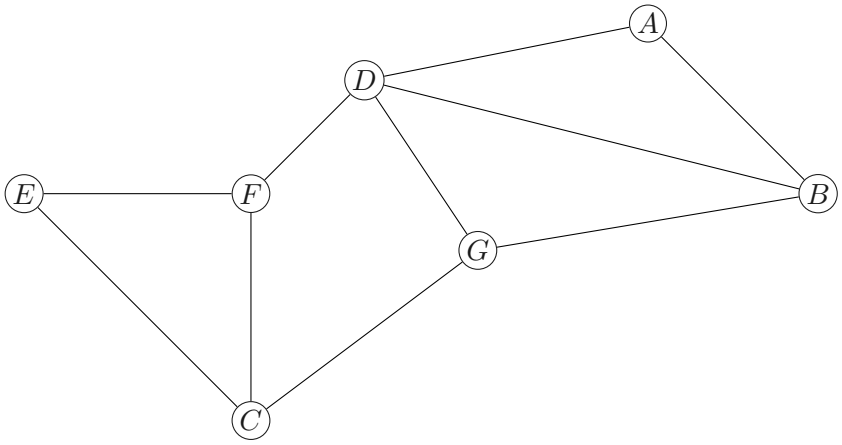


Aufgabe 5 (Token)

Eine andere Variante den gegenseitigen Ausschluss zu realisieren, sind Token-basierte Ansätze.

a) Was sind Nachteile eines Tokenrings?

b) Um den Lift-Algorithmus nutzen zu können, muss ein Spannbaum über dem Netzwerk konstruiert werden. Dafür kann der Echo-Algorithmus verwendet werden. Konstruieren Sie den Spannbaum für das folgende Netzwerk. Dabei soll Knoten A den Algorithmus initialisieren. Nehmen Sie an, dass die Länge der Kanten proportional zur Nachrichtenlaufzeit über die Kante ist.



c) Nehmen Sie an, dass A das Token besitzt und E, C und B das Token anfragen. Notieren Sie alle möglichen Sequenzen wie das Token durchs Netzwerk wandern kann, um die Anfragen abzuarbeiten. Machen sie einen Vorschlag, wie unnötig lange Wege zur Abarbeitung vermieden werden können.

Aufgabe 6 (Verbesserter Echo-Algorithmus)

Der Verbesserte Echo-Algorithmus nutzt Tabu-Mengen. Wie viele Nachrichten werden in den folgenden Topologien dadurch gespart?

a) (balancierter) Baum mit n Knoten

b) vollständiger Graph mit n Knoten