

**Einführung in die Informatik: Systeme und Anwendungen**  
SS 2008

**Übungsblatt 3: Prozesse**

Abgabe bis Montag, 05.05.2008, 14:00 Uhr bzw. in den Übungen

**Aufgabe 3-1**     *Prozesse*

An einer vierarmigen Verkehrskreuzung mit geringem Verkehrsaufkommen wird der Verkehr auf bestimmte länderspezifische Weise geregelt. Wir betrachten die deutsche Verkehrsregelung „rechts vor links“, die amerikanische Verkehrsregelung “four-way stop” sowie den britischen „Kreisverkehr“.

Solch eine Situation kann man als System von Prozessen modellieren: ein Prozess  $K$  für die Kreuzung sowie Prozesse  $A_i$  für beliebig viele Autos. Für Deutschland kann man die Prozesse z.B. folgendermaßen definieren:

```
PROZESS  $K$      // benutzt eine Liste  $L$  mit 4 Plätzen
{
  REPEAT {
    FOR  $k = 1$  TO 4 {
      IF (Auto  $A$  kommt auf Kreuzungsarm  $k$  an) THEN {
        füge  $A$  in  $L$  an Platz  $k$  ein
      }
      IF (Platz  $k$  in  $L$  ist frei) THEN {
        IF ( $k = 4$ ) THEN {
           $f = 1$ 
        } ELSE {
           $f = k + 1$ 
        }
        IF (Platz  $f$  in  $L$  ist belegt mit Auto  $A$ ) THEN {
          signalisiere  $A$  ein OK
          gib Platz  $f$  in  $L$  frei
        }
      }
    }
  }
}
```

```
PROZESS  $A_i$      // für  $i = 1, 2, \dots$ 
{
  signalisiere Ankommen
  warte auf OK
  fahre
}
```

Das amerikanische System ist ähnlich. Die Prozessbeschreibung für die Kreuzung  $K$  wird ersetzt durch

```
PROZESS  $K$  // benutzt eine Liste  $L$  mit 4 Plätzen
{
  REPEAT {
    FOR  $k=1$  TO 4 {
      IF (Auto  $A$  kommt auf Kreuzungsarm  $k$  an) THEN {
        füge  $A$  am Ende von  $L$  ein
      }
    }
  }
  signalisiere erstem Auto  $A$  in  $L$  ein OK
  entferne  $A$  aus  $L$ 
}
}
```

Schließlich definieren wir für das britische Modell „Kreisverkehr“:

```
PROZESS  $K$  // benutzt eine Liste  $L$  mit 4 Plätzen
{
  REPEAT {
    FOR  $k=1$  TO 4 {
      IF (Auto  $A$  auf Platz  $k$  in  $L$  fährt aus Kreis heraus) THEN {
        gib Platz  $k$  in  $L$  frei
      }
      IF (Auto  $A$  wartet an Kreuzungsarm  $k$ ) THEN {
        IF (Platz  $k$  in  $L$  ist frei) THEN {
          signalisiere  $A$  ein OK
          füge  $A$  in  $L$  an Platz  $k$  ein
        }
      }
    }
  }
  rotiere  $L$  um einen Platz
}

PROZESS  $A_i$  // für  $i=1,2,\dots$ 
{
  signalisiere Ankommen
  warte auf OK
  fahre in den Kreis hinein
  WHILE (Zielkreuzungsarm ist nicht erreicht) {
    fahre im Kreis zum nächsten Kreuzungsarm
  }
  fahre aus dem Kreis heraus
}
```

Dabei wird davon ausgegangen, dass ein Rotieren der Liste des Prozesses  $K$  einem Fahren aller beteiligten Prozesse  $A_i$  entspricht.

Für ein Auto  $A_i$  ist die *Wartezeit* die Zeit, die zwischen dem Signalisieren des Ankommens und dem Fahren vergeht. Ein Kreuzungsprozess wird *fair* genannt, wenn er folgende Eigenschaft hat:

*Für alle Autos  $A_i$  gilt: Wenn  $A_i$  das Ankommen signalisiert hat, dann kann es nach endlicher Wartezeit fahren.*

Ein *Deadlock* ist ein Zustand, in dem kein einziges Auto mehr fahren kann. Eine Kreuzung wird *deadlock-frei* genannt, wenn sie nie zu einem Deadlock führt.

- (a) Welche der angegebenen Kreuzungen sind fair? Geben Sie in jedem der drei Fälle eine Begründung oder ein Gegenbeispiel dafür an.
- (b) Wie hängen die Begriffe fair und deadlock-frei zusammen, d.h. wie kann man von dem einen auf das andere schließen?
- (c) Zeigen Sie, wo und wie Deadlocks in den obigen Beispielen auftreten können.

### **Aufgabe 3-2**     *Prozesse*

Gegeben seien zwei Prozesse  $D_1$  und  $D_2$ , die jeweils eine Datei drucken möchten. Da zu einem bestimmten Zeitpunkt immer nur eine Datei gedruckt werden kann, wird zur Synchronisation eine Warteschlange  $W$  und eine Variable  $pos$  verwendet. Die Warteschlange ist nichts anderes als eine Liste von Dateinamen. In ihr werden die Dateien in der Reihenfolge eingetragen, in der sie vom Drucker abgearbeitet werden. Die Variable  $pos$  enthält den Index der nächsten Position in der Warteschlange.

- (a) Geben Sie Prozessbeschreibungen für  $D_1$  und  $D_2$  ohne Prozesskoordination an. Welches Problem kann auftreten, wenn  $D_1$  und  $D_2$  parallel ausgeführt werden?
- (b) Geben Sie Prozessbeschreibungen für  $D_1$  und  $D_2$  mit Prozesskoordination nach Dekker an, so dass das in (a) beobachtete Problem bei paralleler Ausführung nicht mehr auftritt.

Veranschaulichen Sie jeweils mögliche Abläufe von  $D_1$  und  $D_2$  in Prozessablauftabellen (Eine Vorlage finden Sie auf der Homepage).

### Aufgabe 3-3 Prozesse

Eine Firma speichert für einen Kunden mit der Kundennummer  $i$  jeweils einen Kontostand  $k_i$ . Dieser kann positiv, negativ oder ausgeglichen sein, je nachdem wieviel der Kunde bei der Firma gekauft hat und wieviel er schon bezahlt hat. Auf die Kontostände können zwei Sachbearbeiter (Prozesse)  $S_1$  und  $S_2$  zugreifen. Die Prozesse  $S_1$  und  $S_2$  haben jeweils eine eigene lokale Variable  $x_1$  bzw.  $x_2$  zur Berechnung des neuen Kontostandes und verfahren nach folgendem Schema (die einzelnen Aktionen sind zusätzlich mit einer Nummer versehen):

```
PROZESS  $S_j$       // für  $j \in \{1,2\}$ 
{
  REPEAT {
(1)     $x_j = k_i$ 
      IF (Kunde  $i$  kauft für Betrag  $b$  ein) THEN {
(2)     $x_j = x_j - b$ 
      }
      IF (Kunde  $i$  zahlt Betrag  $w$  ein) THEN {
(3)     $x_j = x_j + w$ 
      }
(4)     $k_i = x_j$ 
  }
}
```

- (a) Der Kunde Nr. 1 hat anfangs einen Kontostand von -100 EUR. Er kauft ein Produkt beim Sachbearbeiter  $S_1$  für 80 EUR. Im selben Durchlauf wickelt der Sachbearbeiter  $S_1$  eine Einzahlung des Kunden über 180 EUR ab. Der neue Kontostand des Kunden Nr. 1 beträgt also 0 EUR. Veranschaulichen Sie diese Vorgänge, indem Sie den zeitlichen Ablauf der Aktionen sowie die Werte der Variablen  $k_1$  und  $x_1$  in einer Prozessablauf-tabelle an. (Eine Vorlage finden Sie auf der Homepage)
- (b) Die Sachbearbeiter  $S_1$  und  $S_2$  bearbeiten parallel *ohne Prozesskoordination* nach obiger Prozessbeschreibung die folgenden Vorgänge: Kunde Nr. 2 hat anfangs einen Kontostand von 0 EUR. Er bestellt zwei Produkte im Wert von 30 EUR bzw. 40 EUR. Beide Bestellungen werden vom Sachbearbeiter  $S_1$  entgegengenommen. Dieser behandelt zuerst die Bestellung über 30 EUR, danach die über 40 EUR. Außerdem überweist der Kunde 70 EUR an die Firma, was von Sachbearbeiter  $S_2$  abgewickelt wird. Welche verschiedenen Kontostände können sich dadurch für den Kunden Nr. 2 ergeben? Geben Sie für jeden der möglichen Kontostände einen Prozessablauf, der zum jeweiligen Kontostand führt, an. (Eine Vorlage finden Sie auf der Homepage)
- (c) Welche der Aktionen (1) – (4) umfasst der kritische Bereich in der obigen Prozessbeschreibung?