

**Einführung in die Informatik: Systeme und Anwendungen**  
SS 2009

**Übungsblatt 2: Betriebssysteme, Prozesse**

**Aufgabe 2-1**    *Betriebssysteme*

- (a) Welche typischen Aufgaben können über System-Aufrufe realisiert werden?
- (b) Was sind die Basiskomponenten eines Rechners in der von-Neumann-Architektur?

## Aufgabe 2-2 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
input ...
variables  $k: \text{Nat}, f: \text{Nat}, \dots$ 
begin
  while true
  do {
     $k = 1$  ;
    while  $k \leq 4$ 
    do {
      if Auto  $A$  kommt auf Kreuzungsarm  $k$  an then
        <füge  $A$  in  $L$  an Platz  $k$  ein> ;
      end if
      if Platz  $k$  in  $L$  ist frei then {
        if  $k = 4$  then
           $f = 1$  ;
        else  $f = k + 1$  ;
        end if
        if Platz  $f$  in  $L$  ist belegt mit Auto  $A$  then {
          <signalisiere  $A$  ein OK> ;
          <gib Platz  $f$  in  $L$  frei> ;
        }
        end if
      }
      end if
       $k = k + 1$  ;
    }
  }
end
```

```
prozess  $A_i$  // für  $i = 1, 2, \dots$ 
...
begin
  <signalisiere Ankommen> ;
  <warte auf OK> ;
  <fahre> ;
end
```

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

```
prozess  $K$  // benutzt eine Liste  $L$  mit 4 Plätzen
input ...
variables  $k : \text{Nat}, \dots$ 
begin
  while true
  do {
     $k = 1$ ;
    while  $k \leq 4$ 
    do {
      if Auto  $A$  kommt auf Kreuzungsarm  $k$  an then
        <füge  $A$  am Ende von  $L$  ein> ;
      end if
       $k = k + 1$ ;
    }
    <signalisiere erstem Auto in  $L$  ein OK> ;
    <entferne  $A$  aus  $L$ > ;
  }
end
```

Schließlich definieren wir für das britische Modell „Kreisverkehr“: (Pseudocode dazu siehe nächste Seite)

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.

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

```

prozess K           // benutzt eine Liste L mit 4 Plätzen
input              ...
variables k: Nat , ...
begin
  while true
  do {
    k = 1 ;
    while k <= 4
    do {
      if Auto A auf Platz k in L fährt aus Kreis heraus then
        <gib Platz k in L frei> ;
      end if
      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> ;
        }
      end if
    }
    end if
    <rotiere L um einen Platz> ;
    k = k + 1;
  }
}
end

```

```

prozess Ai         // für i = 1, 2, ...
...
begin
  <signalisiere Ankommen> ;
  <warte auf OK> ;
  <fahre in den Kreis hinein> ;
  while Zielkrenzungsarm ist nicht erreicht
  do {
    <fahre im Kreis zum nächsten Kreuzungsarm> ;
  }
  <fahre aus dem Kreis heraus> ;
end

```

### Aufgabe 2-3 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  $S_j$ 
...
begin
  while true
  do {
(1)     $x_j = k_i$ ;
        if Kunde  $i$  kauft für Betrag  $b$  ein then
(2)       $x_j = x_j - b$ ;
        end if
        if Kunde  $i$  zahlt Betrag  $w$  ein then
(3)       $x_j = x_j + w$ ;
        end if
(4)       $k_i = x_j$ ;
  }
end
```

- (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?