



Sys. Inf. III (Betriebssysteme)

Wintersemester 2004/05

Serie 2

25. Oktober 2004

Thema: Synchronisation

Ausgabetermin: 25. Oktober 2004

Abgabe: 29. Oktober 2004 (12:00)

Zwei Worte vorneweg:

1. die meisten haben es mitbekommen:

(a) Die Seite mit Informationen zum Kurs (Übungszettel etc.) lautet

<http://www.informatik.uni-kiel.de/inf/deRoever/WS0405/0S/>

Wichtige Dinge werden allerdings in der Vorlesung angesagt und ausgeteilt.

(b) das studentische Forum unter

<http://www.infmath.de>

bietet ebenfalls Informationen und Diskussionen zu (unter anderem) dieser Vorlesung (denke ich).

2. Das Semester fängt immer ein wenig ruckelig an, bis sich der Staub der Übungsgruppen-Zuteilung gelegt hat und sich der Ausgabe-Abgabe-Korrektur-Zyklus eingeschwungen hat (diesmal besonders, da *kommende Woche* die Vorlesung (nicht die Übungen) ausfällt. Wegen der Nachfrage: als Schema streben wir an

Ausgabe der Zettel am Montag in der Vorlesung oder vom Netz

Abgabe der Zettel *freitags* (12:00) in den Schrein in Zweiergruppen, die sich für die Dauer des Semesters zusammenfinden.

Besprechung und Rückgabe der Zettel dann in der folgenden Woche besprochen, wo natürlich auch Fragen zu vergangenen Vorlesungen sowie Fragen zum neuen Zettel ("wie ist die Aufgabe zu verstehen?" ...) behandelt werden.

Aufgabe 1 (Variation of Peterson (2 Punkte)) Zeigen Sie oder widerlegen Sie dass der folgende Algorithmus gegenseitigen Ausschluss gewährleistet. Inwieweit hat dieser Algorithmus was mit dem Peterson Algorithmus aus dem Script zu tun?

```
// boolean P0_wants_to_enter , P1_wants_to_enter = false ;
// last = 0

process P0 =
begin

  while true                               /* Endlosschleife      */
  do
    some_actions ;                          /* unkritischer Code    */

    last := 0 ;
    P0_wants_to_enter := true ;
    while (P1_wants_to_enter and last = 0)
    do
      skip ;
    od ;
    critical_section_0 ;
    P0_wants_to_enter := false ;
  od ;
end ;
```

Aufgabe 2 (Dekker (6 Punkte)) Untersuchen Sie den angegebenen Algorithmus auf folgende Eigenschaften:

1. Gewährleistung des gegenseitigen Ausschlusses.
2. Freiheit von Verklemmung (oder *deadlock*-Freiheit).
3. Kann die Terminierung („Absturz“) eines Prozesses in seinem nicht-kritischen Abschnitt den Partnerprozeß negativ beeinflussen?
4. Verhält sich die Lösung gutwillig, wenn ein Prozeß wesentlich öfter als der andere in seinen kritischen Abschnitt will?
5. Kann es passieren, daß ein Prozeß, obwohl er seinen kritischen Abschnitt betreten will, unendlich lange darauf wartet?

Begründen Sie jeweils Ihre Antwort stichhaltig.

```
// Dekkers Algorithmus für Gegenseitigen Ausschluss
// boolean P0_wants_to_enter, P1_wants_to_enter = false; favored = 0

process P0 =
begin

    while true                               /* Endlosschleife      */
    do
        some_actions;                         /* unkritischer Code    */

        P0_wants_to_enter := true;
        while P1_wants_to_enter
        do
            if favored = 1
            then P0_wants_to_enter := false;
                 while favored = 1 do skip od;
                 P0_wants_to_enter := true;
            fi
        od;
        critical_section_0;
        favored := 1;
        P0_wants_to_enter := false;
    od;
end;
```
