Übungen zur Vorlesung
Betriebssysteme I
Wintersemester 2001/02

AG BS
Jan Peleska, Aliki Tsiolakis

Blatt 1

Prozeßverwaltung

Aufgabe 1: Petersons Algorithmus

Petersons Algorithmus zur Realisierung von *mutual exclusion* ist im Folgenden in ANSI C angegeben; er besteht aus 2 Prozeduren, die von den Prozessen aufgerufen werden, die auf die kritische Region zugreifen wollen.

```
#define FALSE  0
#define TRUE   1
#define N      2                /* number of processes                   */

int turn;                       /* whose turn is it?                     */
int interested[N];              /* all values initially 0 (FALSE)        */


void enter_region (int process) /* process: who is entering (0 or 1)     */
{
  int other;                    /* number of other process               */

  other = 1 - process;          /* the opposite of process               */
  interested[process] = TRUE;   /* show that you are interested          */
  turn = process;               /* set flag                              */
  while (turn == process && interested[other] == TRUE); /* null statement  */
}


void leave_region (int process) /* process: who is leaving (0 or 1)      */
{
   interested[process] = FALSE; /* indicate departure from critical region */
}
```

a) Geben Sie (in FDR-Syntax) eine CSP-Spezifikation dieses Algorithmus' an, indem Sie zwei Prozesse spezifizieren, die möglichst direkte Umsetzungen der beiden Prozeduren `enter_region` und `leave_region` aufrufen.

b) Weisen Sie mit Hilfe von FDR nach, daß der Algorithmus auch tatsächlich *mutual exclusion* garantiert. Definieren Sie dazu eine geeignete Abstraktion des erwarteten Verhaltens. Dokumentieren Sie das Ergebnis geeignet.

Aufgabe 2: Verallgemeinerung auf $N \geq 2$ Prozesse

Die Verallgemeinerung von Petersons Algorithmus mit einem Array `interested[N]` für $N = 3$ ist keine Lösung für *mutual exclusion*. Begründen Sie diese Aussage.

Aufgabe 3: Wechselseitiger Ausschluss mit Test-Set-Lock-Variable

Modellieren Sie eine Test-Set-Lock-Variable mit den darauf verfügbaren Operationen `test-and-set` (Abfrage auf aktuellen Wert und danach setzen auf 1) und `reset` (Setzen des Wertes auf 0).

a) Geben Sie analog zu 1.a eine neue Version von `enter_region` und `leave_region` mit TSL-Verwendung an.

b) Führen Sie (analog zu 1.b) mit FDR den Beweis für den wechselseitigen Ausschluss.

Abgabe: Bis Montag, 12. November 2001, in der Übung.

Die Abgabe des gesamten Übungsblattes kann wahlweise auf Papier oder als Email an `tsio@informatik.uni-bremen.de` erfolgen. **In jedem Fall** sollen die CSP-Spezifikationen zusätzlich auch in elektronischer Form (als Mail-Attachment an genannte Email-Adresse) abgegeben werden!

Bei jeglicher Form der Abgabe die Gruppennummer und die Namen aller Gruppenmitglieder nicht vergessen!