Christian-Albrechts-Universität zu Kiel
Institut für Informatik und Praktische Mathematik
Prof. Dr. W.-P. de Roever
Dipl. Inf. K. Baukus · Dipl. Inf. K. Stahl

Informatik III

Wintersemester 2001/02 — 29. Oktober 2001

Serie 2

Thema :

Ausgabetermin: 29. Oktober 2001

Abgabe: 5. November 2001, bis 12:00 Uhr im Schrein

Aufgabe 1: [Stack]

Zeichnen sie den Verlauf des Laufzeitstacks für folgende kleine Prozedur. Beim Aufruf mit dem Wert 3 auf.[1]

```
int fib(int i) {
  if (i==0) return 1;
  if (i==1) return 1;
  return (fib(i-1) + fib(i-2));
};
```

Aufgabe 2: [Produzent/Konsument]

Ein klassisches Prozeßsynchronisationsproblem ist das *Produzenten/Konsumenten*-Problem[2] Gegeben sei folgendes Programmfragment, welches 2 Prozesse definiert:

```
/* producer/consumer Pseudocode                    */

#define N   100                /* Plaetze im Puffer  */
int buffer[N];
int count    0;                /* belegte Plaetze    */

...
```

[1] Richtig. Fibonnacci soll man nicht auf diese Weise programmieren.

[2] Auch als das Problem des beschränkten Puffers bekannt (auf Englisch *producer/consumer*, *bounded-buffer*).

```
producer ()
{
  while (TRUE) {
     ...
     buffer [count++] = produce_item();
     ...
  }
}

consumer ()
{
  while (TRUE) {
     ...
     consume_item(buffer[count--]) ;
     ...
  }

}
```

Die Absicht ist, die notwendige *Synchronisation* zwischen Konsument und Produzent, um einen Zugriff auf den Buffer außerhalb der Arraygrenzen zu verhindern, durch Verwendung von Semaphoren zu bewerkstelligen. Ergänzen Sie den Code entsprechend und begründen Sie die Korrektheit Ihrer Lösung!