

Übung 1

Aufgabe 1 Cobegin in Ada

Zeigen Sie, wie eine Cobegin-Anweisung in Ada implementiert werden kann.

Gehen Sie z.B. davon aus, dass A und B zwei Ada-Anweisungen sind, die nebenläufig ausgeführt werden sollen. Schreiben Sie dazu ein Ada-Programm, das in seinem Verhalten

```
cobegin A; B; coend;
```

entspricht.

Aufgabe 2 Fork & join in Ada

Wie kann in Ada Fork & Join realisiert werden.

Aufgabe 3 Task-Hierarchien in Ada

Bestimmen Sie im folgenden Ada-Programm für jede Task ihre Eltern-Task und ihre Master-Task sowie, falls vorhanden, ihre Kinder-Tasks und die abhängigen Tasks.

```
procedure Main is
  procedure Hierarchy is
    task A;
    task type B;

    type Pb is access B;
    Pointerb : Pb;

    task body A is
      task C;
      task D;
      task body C is
        begin
          -- sequence of statements including
          Pointerb := new B;
        end C;
      task body D is
        Another_Pointerb : Pb;
        begin
          -- sequence of statements including
          Another_Pointerb := new B;
        end D;
      begin
        -- sequence of statements
      end A;

      task body B is
        begin
          -- sequence of statements
        end B;
      begin
        -- sequence of statements
      end Hierarchy;
    begin
      -- sequence of statements
    end Main;
```

Aufgabe 4 *Prozesssteuerung in C/Real-Time POSIX*

Implementieren Sie das "einfache eingebettete System" aus der Vorlesung (S. 2-41 ff) in C/Real-Time-Posix.

Aufgabe 5 *Zufahrtskontrolle zu einem Parkplatz*

Skizzieren Sie mit Hilfe nebenläufiger Tasks die Struktur eines Programms zur Steuerung der Zufahrt auf einen Parkplatz. Der Parkplatz ist mit einer Einfahrtsschranke, einer Ausfahrtsschranke und einem Signal, dass den Parkplatz als voll belegt oder frei anzeigt ausgestattet.

Zeigen Sie, wie diese Struktur in Ada und in C/Real-Time POSIX realisiert werden kann.