



## Übungsblatt 07

### Aufgabe 22 Protected Typen (6 Punkte)

- Erläutern Sie kurz das Konzept der *protected* Typen in Ada. Erläutern Sie insbesondere den Schutz des Zugriffs auf interne Strukturen und das Konzept eines **entry**.
- Vergleichen Sie den **entry** eines *protected* Typen mit dem eines *Tasks*.
- Wie unterscheiden sich *Tasks* und *protected* Typen?

### Aufgabe 23 Ada-Philosophen (5 Punkte)

Eines der bekanntesten Beispiele aus dem Bereich verteilter Programmierung, Betriebsmittelverwaltung und Deadlock-Verwaltung ist das der fünf Philosophen:

*Um einen Tisch sitzen fünf Philosophen. Jeder Philosoph hat einen Teller mit Reis und zwischen den Tellern liegt jeweils ein Essstäbchen. Die Philosophen sind so in ihre Probleme vertieft, dass sie für die Außenwelt keine Wahrnehmung haben. Lediglich ihren Hungermühsen sie gelegentlich stillen. Allerdings benötigen sie zum Essen unbedingt zwei Stäbchen; es können also gleichzeitig höchstens zwei Philosophen essen. Bekommt ein Philosoph Hunger, so muss er zunächst versuchen die beiden zu seinem Teller benachbarten Stäbchen aufzunehmen und kann dann essen: Sobald er satt ist, legt er die Stäbchen wieder zurück und grübelt weiter über die Probleme der Welt.*

Schreiben sie ein nebenläufiges Ada-Programm, dass dieses Problem modelliert. Realisieren Sie dazu jeden Philosophen als *Task*.

#### Hinweise:

- Die Verwendung der Stäbchen muss synchronisiert werden. Jedes Stäbchen darf nur von einem Philosophen gleichzeitig benutzt werden.
- Jeder Philosoph isst nur endlich lange.
- Folgende Probleme gilt es zu vermeiden:
  - Verklemmung (Deadlock): Jeder Philosoph will essen, aber keiner kann, weil jeder genau ein Stäbchen hält und auf das zweite wartet.
  - Verhungern: Ein Philosoph bekommt niemals etwas zu essen, da seine Nachbarn ihm bei den Stäbchen immer zuvor kommen.

## **Aufgabe 23 Typ `protected_buffer` (4 Punkte)**

Implementieren Sie einen Typ `protected_buffer`, der für alle Puffer im CAN-Treiber genutzt werden soll. Dieser Typ soll generisch sein und exklusiven Zugriff auf interne Datenstrukturen bieten, da aus verschiedenen konkurrierenden Tasks darauf zugegriffen wird. Der lesende Zugriff soll für einen leeren Puffer nicht blocken, entsprechendes gilt für das Schreiben in den vollen Puffer. In diesen Fällen soll eine entsprechende Ausnahme erweckt werden.

Nutzen Sie für die Pufferung der Nachrichten (Eingang/Ausgang) verschiedene Puffer dieses generischen Typs.

*Abgabe: Donnerstag, 14.07.2005*

---

Sie können Ihre Lösung zum obigen Termin in Papierform in der Vorlesung oder in elektronischer Form per eMail an [ada@i3.informatik.rwth-aachen.de](mailto:ada@i3.informatik.rwth-aachen.de) abgeben.

Bitte vermerken Sie in jedem Fall die Namen und Matrikelnummern aller beteiligten Personen (maximal 3), beim Versenden per eMail auch alle eMail-Adressen. Abgaben in elektronischer Form können ausschließlich in den Dateiformaten *Plain-Text* oder *PDF* erfolgen.

Aktuelle Informationen zur Vorlesung finden Sie auf den Webseiten des Lehrstuhls unter <http://www-i3.informatik.rwth-aachen.de>.