

EINFÜHRUNG IN DIE SITZUNG "BETRIEBSSYSTEME"

H.J. Siegert

Institut für Informatik
Technische Universität München

Da die Vorträge dieser Sitzung sehr verschiedene Teilaspekte von Betriebssystemen behandeln, soll auf Wunsch des Programm-Ausschusses eine Einführung in die Sitzung gegeben werden. Ich möchte hierzu wichtige Grundbegriffe und Grundkonzepte, die bei den Vorträgen vorausgesetzt werden, erläutern.

1. Monitor

Der Begriff des Monitors wurde ca. 1973 von Brinch Hansen [1,2] und Hoare [12] eingeführt. Die Vorstellung hierbei war, daß in einer Rechenanlage mehrere Prozesse parallel ablaufen und in kontrollierter Weise auf gemeinsame Betriebsmittel zugreifen müssen. Um eine solche Kontrolle zu realisieren, sollten die Verwaltung, die Zuteilung und die Zugriffsoperationen für jedes Betriebsmittel zu einem Monitor zusammengefaßt werden. In anderen Worten: Jeder Monitor repräsentiert einen abstrakten Datentyp des Betriebssystems. In Anlehnung an den Klassenbegriff von SIMULA67 hat nach Hoare [12] ein Monitor folgenden Aufbau:

```
monitor <monitorname>;  
begin  
    <Vereinbarung lokaler Daten>;  
    <Vereinbarung lokaler Prozeduren>;  
    <Vereinbarung öffentlich zugänglicher Prozeduren>;  
    <Initialisierung lokaler Daten>;  
end;
```

Zur Behandlung von Wartezuständen werden monitorlokale Variable von der Art condition eingeführt. Hierauf sind 2 Operationen erklärt:

```
wait(<Variable von der Art condition>),
signal(<Variable von der Art condition>).
```

Es gelten folgende Regeln:

1. Ein Monitor wird erzeugt und ist ab diesem Zeitpunkt existent, unabhängig davon, ob momentan eine Prozedur des Monitors ausgeführt wird oder nicht.
2. Auf die lokalen Daten und die lokalen Prozeduren des Monitors kann nur innerhalb des Monitors zugegriffen werden.
3. Ein Monitor ist exklusives Betriebsmittel, d.h. er kann nur dann von einem Prozeß durch Aufruf einer öffentlichen Prozedur des Monitors betreten werden, wenn kein anderer Prozeß im Monitor ist.
4. Ein Prozeß verläßt den Monitor beim Rücksprung aus der aufgerufenen öffentlichen Prozedur.
5. Ein Prozeß tritt in einen Wartezustand ein, wenn er auf eine Anweisung `wait(ci)` trifft. Er wird hierbei in eine Warteschlange bzgl. `ci` eingereiht. Der Monitor wird temporär freigegeben, so daß andere Prozesse eintreten können.
6. Wird die Anweisung `signal(ci)` ausgeführt, so verläßt der erste Prozeß in der Warteschlange bzgl. `ci` den Wartezustand. Er setzt "sobald der Monitor wieder verfügbar ist" mit dem auf `wait(ci)` folgenden Befehl fort.

Für die Implementierung der Regel 6 gibt es eine Reihe von Vorschlägen, die im Vortrag "Implementierung von Monitoren" von Herrn Nehmer diskutiert werden.

Das Monitorkonzept wurde bisher nur bei Experimentalsystemen, z.B. [3,23] eingesetzt. Es bringt Vorteile gegenüber der direkten Verwendung von V-Operationen:

- Die Programmierung ist nicht so fehleranfällig wie die direkte Verwendung von P- und V-Operationen.
- Der Zugriff auf das Betriebsmittel ist nur in der vorgesehenen Weise über die Prozeduren im Monitor möglich.
- Es wird eine klare Programmstruktur erzwungen.