RECHENZENTRUM TH MÜNCHEN
ARBEITSGRUPPE BETRIEBSSYSTEME

INTERNSCHRIFT Nr. 16

THEMA:

Semaphore

VERFASSER:

SAPPER

DATUM:

WS 68/69

FORM DER ABFASSUNG

- ENTWURF
- [X] AUSARBEITUNG
- ENDFORM

SACHLICHE VERBINDLICHKEIT

- [X] ALLGEMEINE INFORMATION DISKUSSIONSGRUNDLAGE
- ERARBEITETER VORSCHLAG
- VERBINDLICHE MITTEILUNG
- VERALTET

ÄNDERUNGSZUSTAND

BEZUG AUF BISHERIGE INTERNSCHRIFTEN

Vorkenntnisse aus:

Erweiterung von:

Ersatz für:

BEZUG AUF KÜNFTIGE INTERNSCHRIFTEN

Vorkenntnisse zu:

Erweiterung in:

Ersetzt durch:

ANDERWEITIGE LITERATUR

Arbeitsseminar

über Betriebssysteme

WS 1968/69

G.R. Sapper

6. "S e m a p h o r e"

I. Simultanarbeit

Mehrere "Prozesse" (im Sinne der "cooperating sequential processes" von E.W. Dijkstra), die aus Elementarschritten bestehen, können simultan oder quasisimultan ausgeführt werden.

Es gibt dabei kritische Abschnitte, in denen gemeinsame Daten gelesen und verändert werden.

Während dieser Zeit muß jeder andere Prozeß am Zugriff auf diese Daten gehindert werden.

Dies muß sicher und ohne unnötiges gegenseitiges Blockieren erfolgen.

II. Unangemessene Sicherungsmethoden

1. Eine "Freizelle", die die Werte 1 oder 2 annehmen kann, zeigt an, ob Prozeß 1 oder Prozeß 2 auf die Daten zugreift. Dies führt dazu, daß sich die Prozesse unnötig synchronisieren. Die Freizelle nimmt nacheinander die Werte 1, 2, 1, 2, ... an und ein Prozeß, der gerade

das Ende seines Zugriffs durch Umbesetzen der Freizelle angezeigt hat, muß erst den Zugriff seines Partners abwarten, bevor er erneut zugreifen darf.

2. Hält man für jeden Prozeß eine "Freizelle", die "Zugriff" oder "frei" anzeigt, gewinnt man eine Zustandsmöglichkeit gegenüber oben mehr: "keiner greift zu". Jedoch besteht nun die Möglichkeit, daß der (verbotene) Zustand "beide greifen zu" erreicht wird. Das Abfragen der fremden Freizelle und das Umbesetzen der eigenen erfordern mindestens zwei elementare Schritte, zwischen denen der Prozeß von seinem Partner unterbrochen werden kann. Versuche, diese Möglichkeit auszuschließen, bringen die Gefahr, daß aus Symmetriegründen die Entscheidung, wer aus dem Zustand "keiner greift zu" bei gleichzeitigem Belegungswunsch zuerst zugreifen darf, erst im Unendlichen entschieden werden kann.

III. Lösung von Th.J. Dekker

Die notwendige Unsymmetrie schafft Dekker, der beide oben beschriebene Methoden kombiniert. In ALGOL formuliert beginnt sein Lösungsvorschlag für zwei Prozesse mit der Deklaration dreier Größen

```
integer  c1, c2, turn;
c1 := c2 := turn := 1;
```

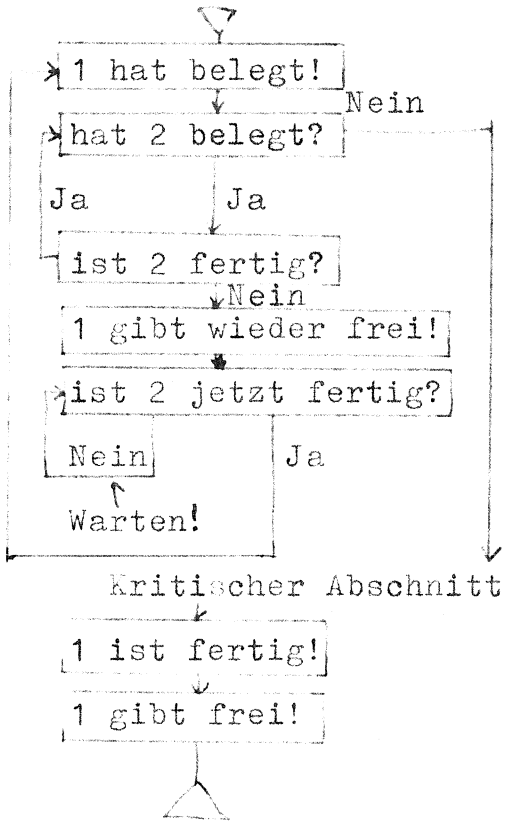
die allen Prozessen zugänglich sind.

c1, c2 haben als Wertevorrat 0 ("belegt")
oder 1 ("frei")

und turn kann 1 ("Prozess 1 belegt die Daten")
oder 2 sein.

Sperrmechanismus des Prozesses 1:
(für Prozeß 2 ersetze man überall 1 durch 2 und umgekehrt)

```
begin A1 : c1 := 0
L1 : if c2 = 0 then
     begin
     if turn = 1 then goto L1;
     c1 := 1;
B1 : if turn = 2 then goto B1;
     goto A1;
     end;
     Kritischer Abschnitt
     turn := 2;
     c1 := 1
     goto A1
end Prozeß 1;
```



Zwei Prozesse sind nie gleichzeitig in einem kritischen Abschnitt, da die fremde Freizelle c erst abgefragt wird, wenn die eigene Freizelle auf "gesperrt" zeigt. Die Größe turn kommt nur ins Spiel, wenn zwei Prozesse "gleichzeitig" in ihre kritischen Abschnitte eintreten wollen.
(Anmerkung: In der Praxis wird diese Gleichzeitigkeit allein durch Warteschlangenbildung bei der Speicheransteuerung hardwareseitig vermieden.)

IV. Semaphore

In der Praxis erweist es sich als unzweckmäßig, Prozesse mittels dynamischem Stop "warten" zu lassen, da sowohl ein Rechen- als auch Speicherwerk eines Rechners damit blockiert werden; besitzt eine Rechenanlage nur ein Rechenwerk, ist dies überhaupt undenkbar.

Dijkstra postuliert nun die Existenz von "Semaphoren", die folgende Eigenschaften haben

Sie sind Common-Variable für zusammenarbeitende Prozesse.

Sie sind ganzzahlig und ≥ 0; für binäre Semaphore sind nur die Werte 0 oder 1 zulässig.

Auf sie dürfen nur die Operationen P und V angewandt werden, die nicht unterbrechbare Elementarschritte sind.

Die V-Operation erhöht den Wert des Semaphores um Eins.

Die P-Operation erniedrigt den Wert des Semaphores um 1, wird das Resultat dabei negativ, so muß gewartet werden, bis durch inzwischen eintreffende V-Operationen als Resultat ein Wert ≥ 0 geliefert werden kann. Die P-Operation kann also für einen Prozeß eine zeitliche Verzögerung bedeuten.

V. Implementierungsfragen

Mit P und V lassen sich eingriffsunempfindliche Zugriffe mehrerer Prozesse auf gemeinsame Daten einfach und übersichtlich formulieren und dokumentieren.

Dijkstra macht wenig Angaben über die praktische Implementierung von Semaphore-Operationen. In Wirklichkeit kommt die Hardware der jeweiligen Maschine sehr stark ins Spiel. Die P- und V-Operationen greifen in die Vergabe der Rechenwerke an verschiedene Prozesse ein. Die als elementar postulierten Operationen P und V werden selbst durch einzelne, unterbrechbare Befehle realisiert. Das hat zur Folge, daß während eines Teils dieser Operationen Unterbrechungen (EA-Eingriffe, Alarme) hardwareseitig gesperrt werden müssen: Alle Semaphore haben als gemeinsame Daten die "Regieliste" und der Regie-

vergabemechanismus darf sich nicht selbst rekursiv aufrufen (indem er z.B. durch einen Alarm unterbrochen wird, solange die Regielisten in einem undefinierten Zwischenzustand sind). Ist die Zahl der konkurrierenden Prozesse und die Zahl der Semaphores nicht stark beschränkt, ergeben sich sehr schnell Speicherplatzschwierigkeiten, da im Zusammenhang mit der Rechnervergabe umfangreiche Listen (für Warteschlangen etc.) geführt werden.

VI. Beispiele

Sehr häufig sind in Betriebssystemen Prozesse, die sich gegenseitig Daten zuspielen. Beispielsweise arbeiten Ein-Ausgabe-(EA) Vorgänge über gemeinsame Puffer, die in Blöcken ein- bzw. auszugebende Information zwischenspeichern und einen möglichst gleichmäßigen Betrieb der meist in recht unregelmäßigen Abständen angesprochenen EA-Geräte ermöglichen. Der Produzent wartet erst dann, wenn der Pufferspeicher überlaufen würde, der verarbeitende Prozeß, wenn der gesamte Pufferspeicher leer vorgefunden wird.

Beim nachfolgenden Beispiel dieser Art werden drei Semaphores benötigt, die wie folgt vorbesetzt sind:

Leerplätze	=	N
Pegel	=	0
Manipuliere	=	1