

## Aufgabe 7:

### Beobachter/Protokollierer

Implementieren Sie das in der Vorlesung vorgestellte Problem von Beobachter und Protokollierer. Ihr Programm **record** soll dabei sowohl für den Beobachter, als auch für den Protokollierer einen eigenen Prozess erzeugen und auf deren Terminierung warten. Die einzelnen Prozesse sollen anschließend folgende Aufgaben übernehmen:

**a) Beobachter:**

Dieser Prozess liest Zeilen von der Tastatur und zählt diese in einer mit dem Protokollierer gemeinsam verwalteten Variable. Der Beobachter soll solange Zeilen zählen, bis Ihm durch Ctrl-D (EOF) mitgeteilt wird, dass die Eingabe beendet ist. In diesem Fall schickt er dem Protokollierer ein SIGTERM und terminiert anschließend.

**b) Protokollierer:**

Der Protokollierer liest im Abstand von 5 Sekunden den Inhalt des gemeinsamen Zählers aus, summiert dessen Inhalt auf eine private Variable und setzt die gemeinsame Variable auf den Wert 0. Anschließend gibt er den gelesenen Wert sowie die Gesamtsumme aller gezählten Zeilen auf die Standardausgabe aus.

Ihre Lösung soll zur Kommunikation weder Unix-Shared-Memory noch Semaphoren verwenden. Vielmehr soll der gemeinsame Speicher durch eine Datei realisiert werden, in der die gemeinsamen Variablen abgelegt werden. Bearbeiten Sie die Aufgabe in zwei Schritten:

**a) Ohne Koordinierung:**

Implementieren Sie das oben beschriebene Programm ohne Koordinierung. Fügen Sie an geeigneten Stellen sleep-Anweisungen ein, damit Sie die in der Vorlesung behandelten Effekte beobachten können. Beschreiben Sie in Ihrer Dokumentation sowohl die auftretenden Effekte als auch den Grund dafür. Welche Operationen müssten Sie als atomar voraussetzen, damit in Ihrer Lösung Zeilen weder verloren gehen, noch doppelt gezählt werden?

**b) Mit Koordinierung**

Verwenden Sie jetzt das Ihnen bekannte Verfahren von Peterson, um den Zugriff auf die gemeinsame Zählervariable in der Datei zu koordinieren. Überzeugen Sie sich, dass die vorher aufgetretenen Fehler jetzt nicht mehr auftreten! Welche Operationen müssen Sie weiterhin als atomar für die korrekte Funktion voraussetzen?

Denken Sie daran, dass die gemeinsame Datei vor Ihrer Benutzung erzeugt und initialisiert, sowie am Ende des Programmlaufes wieder gelöscht werden muss.

**Abgabe: bis spätestens Mittwoch, 09.01.2002, 16:30 Uhr**