Christian-Albrechts-Universität zu Kiel
Institut für Informatik und Praktische Mathematik
Prof. Dr. W.-P. de Roever
Martin Steffen

Informatik III (OS)

Wintersemester 2003/04 — **Serie 4** — 18. November 2003

Thema : **Calling Conventions, Scheduling, Queues**

Ausgabetermin: 18. November 2003

Abgabe: 26. November 2003

Aufgabe 1 (Calling Conventions (3 Punkte)) Beschreiben Sie das Ablaufgeschehen beim Aufruf einer Prozedur **B** von Prozedur **A**:

```
int A (arg1, ... , argn)
{
      .....

      B(arg1, ..., argm);

      .....
}
```

Erläutern Sie die Abarbeitung der Routinen *csv* und *cret* zeilenweise und protokollieren Sie den Stackinhalt währenddessen!

Aufgabe 2 (Scheduling (4 Punkte)) Unter *Scheduling* verteht man ganz allgemein die Festlegung der Ausführungsreihenfolge verschiedener Jobs/Tasks/Prozesse oder Ähnlichem. Unter *nicht-unterbrechendem* (*non-preemptive*) Scheduling versteht man Strategien, bei denen ein Job, einmal „gescheduled“, bis zu Ende arbeiten muß und nicht unterbrochen werden kann.[1]

Betrachten Sie folgendes Problem des *nicht-unterbrechenden* Scheduling. Gegeben seien Jobs mit bekannter Bearbeitungsdauer von

4, 7, 9, 4 und x

Zeiteinheiten. Welches Scheduling minimiert die mittlere Zeit, bis ein Job terminiert ist (Minimierung der mittleren *response time*)? Formulieren Sie eine allgemeine Strategie, die die mittlere *response time* minimiert, und zeigen Sie deren Korrektheit. Diskutieren Sie mögliche Anwendungen in Betriebssystemen.

[1] Das *Prozeß*scheduling in Timesharingsystemen wie in Xinu ist selbstverständlich *unterbrechend* (*preemptive*).

Aufgabe 3 (Datenstrukturen (6 Punkte))

- In der Vorlesung wurde eine bestimmte Implementierung von *priority queues* mittels doppelt-verzeigerter Listen vorgestellt (s. Kapitel 3 aus [Com83]). Verändern Sie die Implementierung auf *einfach verzeigerte* Listen und re-implementieren sie die Prozeduren `enqueue`, `dequeue` und `insert` entsprechend neu. Diskutieren sie Vor- und Nachteile dieser Implementierung gegenüber der mit doppelter Verzeigerung.

- Verbessern Sie die Prozedur `newqueue` (S. 50 in [Com83]) so, daß sie verhindert, daß die maximale Kapazität der $\mathcal{Q}$-Struktur überschritten wird. (`NQENT` ist auf Seite 44 erklärt).

Literatur

[Com83] Douglas Comer. *Operating System Design, The Xinu Approach.* Prentice Hall, 1983.