

6. Übung

Ausgabe Abgabe
21.11.14 5.12.14

Bitte bei der Abgabe Name der Mitglieder einer Gruppe, Nummer der Übung/Teilaufgabe und Datum auf den Lösungsblättern nicht vergessen! Darauf achten, dass die Lösungen beim richtigen Tutor abgegeben werden. Achten Sie bei Programmieraufgaben außerdem darauf, dass diese im Linuxpool kompilierbar sind. Nutzen Sie dazu die Flags `-std=c99`, `-Wall` und `-pedantic`. Es sollten keine Warnungen auftauchen.

Zu spät abgegebene Lösungen werden nicht berücksichtigt!

Aufgabe 1: Scheduling-Strategien (2 Punkte)

Sie haben in der Vorlesung nicht-preemptive Scheduling Strategien kennengelernt. In der Vorlesung wurden die Strategien jeweils für eine CPU vorgestellt. Sie sind aber leicht so erweiterbar, dass sie ankommende Prozesse auf mehreren CPUs verteilen können.

Betrachten wir als Beispiel die Kassen eines Supermarktes. Jede Kasse entspricht einer CPU und jeder Kunde, der an die Kasse kommt, stellt einen Prozess dar. Die Zahl der Artikel, die ein Kunde eingekauft hat, entspricht der Zeit, die ein Prozess zur Ausführung benötigt. Wir nehmen dabei an, dass die Zeit zum Registrieren eines Artikels an der Kasse konstant ist und setzen diese mit einer Zeiteinheit an, den eigentlichen Bezahlvorgang vernachlässigen wir.

Da es Kunden im Supermarkt nicht zugemutet werden kann, andere Kunden vorzulassen, kommt üblicherweise die FIFO-Strategie zum Einsatz, d.h. wer zuerst die Kasse erreicht, wird zuerst abkassiert. Da die Kunden eines Supermarktes auf ihren Vorteil bedacht sind, stellen sie sich immer an der Kasse an, an der die Wartezeit am geringsten ist, d.h. an der vor ihnen die wenigsten Artikel registriert werden müssen.

- a) Ein Supermarkt habe drei Kassen (K1, K2 und K3). An keiner der Kassen wartet ein Kunde. Nun treffen (fast gleichzeitig) Kunden mit folgenden Artikelanzahlen (in dieser Reihenfolge) ein:

6, 15, 23, 10, 3, 13, 15, 7, 20, 40, 19, 4, 6, 21 (14 Kunden)

O.B.d.A. stellt sich der erste Kunde an K1, der zweite an K2, der dritte an K3 an.

- Dokumentieren Sie die sich ergebenden Warteschlangen an den Kassen K1, K2 und K3.
 - Wie viele Zeiteinheiten muss ein Kunde im Durchschnitt warten, bis er an der Kasse ankommt?
- b) Der Leiter des Supermarktes ist mit dieser mittleren Wartezeit für seine Kunden nicht zufrieden und führt eine Kasse ein, an der nur Kunden bedient werden, welche weniger oder gleich 11 Artikel eingekauft haben (K1 wird in diesem Sinn ausgezeichnet). Kunden mit wenigen Artikeln dürfen sich natürlich nach wie vor auch an den anderen Kassen anstellen, sofern sie eine geringere Wartezeit erwarten.
- Dokumentieren Sie die sich ergebenden Warteschlangen (K1 ist die Kasse für Kunden mit weniger als 12 Artikeln; die ersten Kunden stellen sich an die Kassen K1, K2, K3 - in dieser Reihenfolge).

- Berechnen Sie die mittlere Wartezeit der Kunden.
 - Was für eine Strategie versucht der Supermarktleiter näherungsweise zu erreichen?
 - Was für einen Nachteil hat die neue Regelung?
- c) Der Supermarktleiter ist nicht zufrieden mit der unter b) eingeführten Lösung. Er entscheidet, die Kasse K1 nur noch für Kunden mit 20 oder mehr Artikeln zu erlauben.
- Dokumentieren Sie die sich ergebenden Warteschlangen (K1 ist die Kasse für Kunden mit mehr als 19 Artikeln; die ersten Kunden stellen sich an die Kassen K2, K3, K1 - in dieser Reihenfolge).
 - Berechnen Sie die mittlere Wartezeit der Kunden.
 - Wird der Supermarktleiter bei dieser Lösung bleiben?

Aufgabe 2: Implementieren eines Schedulers (3 Punkte)

Implementieren Sie die Schedulingalgorithmen Round Robin, First Come First Serve und Shortest Process Next. Sie müssen hierbei nur die vorgefertigten Rahmenmethoden

```
1 LINK roundRobin(LINK head, LINK current, int tStep);  
2 LINK fcfs(LINK head, LINK current, int tStep);  
3 LINK spn(LINK head, LINK current, int tStep);
```

implementieren.

Hinweis:

Verwenden Sie das bereitgestellte Framework oder ergänzen Sie Ihre Lösung der vierten Programmieraufgabe.

Das bereitgestellte Framework ist eine Erweiterung des Frameworks der vierten Programmieraufgabe.

Nutzen Sie die bereitgestellten Hilfsfunktionen.