

Simulation der Scheduling-Algorithmen (10 Punkte)

Aufgabenstellung

1. Führen Sie eine Handsimulation der Scheduling-Algorithmen auf dem beiliegenden Übungsblatt mit den angegebenen Werten durch. (2 Punkte)
2. Implementieren Sie folgende in der Vorlesung behandelte Scheduling-Algorithmen:
 - First Come First Served
 - Shortest Job First
 - prioritätsgesteuertes Scheduling
 - Round Robin mit konstanter Zeitscheibe unabhängig von der Priorität
 - Round Robin mit Zeitscheibendauer proportional zur Priorität

Ihr Programm soll die Algorithmen vollständig simulieren. Außerdem soll die mittlere Verweilzeit bestimmt werden. Verwenden Sie zur Simulation der Abläufe eine Liste von Jobs. Jeder Job speichert seinen Namen, seine Abarbeitungszeit und seine Priorität. Sie können zur vereinfachten Umsetzung die beiliegende Liste verwenden. Die Ausgaben sollen folgendermaßen aussehen und bei jedem Zeitschritt ausgegeben werden:

Beispielausgabe zu First Come First Served:

a wurde abgearbeitet (Aktuelle Zeit: xx).
b wurde abgearbeitet (Aktuelle Zeit: xx).
c wurde abgearbeitet (Aktuelle Zeit: xx).
d wurde abgearbeitet (Aktuelle Zeit: xx).
e wurde abgearbeitet (Aktuelle Zeit: xx).
Mittlere Verweilzeit: xx.xx s.

Beispielausgabe zu Round Robin:

Es wird an den Jobs zu folgenden Anteilen gearbeitet:

Es wurde 1s an a gearbeitet.

Es wurde 1s an b gearbeitet.

Es wurde 1s an c gearbeitet.

Es wurde 1s an d gearbeitet.

Es wurde 1s an e gearbeitet.

Es wird an den Jobs zu folgenden Anteilen gearbeitet:

...

Testen Sie alle fünf Algorithmen mit einer Reihe von Jobs verschiedener Größe und vergleichen Sie die Ergebnisse. Testen Sie die Algorithmen ebenfalls für die im ersten Aufgabenteil vorgegebenen Jobs. Anhand der Ergebnisse können Sie überprüfen, ob Sie die Algorithmen korrekt implementiert haben.

Hinweis: Binden Sie zur Verwendung der Liste die Dateien `lists.h`, `lists.c`, `global.h` und `global.c` in Ihr Projekt ein. Linken Sie zudem die Bibliothek `pthread`. Dies ist nur notwendig, da die Liste synchronisiert arbeitet; Sie müssen in dieser Aufgabe keine Threads verwenden.

(8 Punkte)