

Portfolioprüfung – Werkstück A – Alternative 4

Aufgabe 1 Aufgabe

Entwickeln Sie einen Simulator für die Scheduling-Verfahren (Algorithmen) **FCFS** (First Come First Served), **Round Robin** mit frei definierbarem Zeitquantum und **Multilevel-Feedback-Scheduling**.

Der Simulator soll die Ausführungsreihenfolge für eine bestimmte Anzahl an Prozessen berechnen und entweder als Gantt-Diagramm (Zeitleiste) oder in einer anderen geeigneten Form ausgeben. Die Anzahl der Prozesse und deren jeweilige Laufzeiten und Ankunftszeiten kann der Benutzer (mit sinnvollen Einschränkungen!) frei festlegen.

Entwickeln und implementieren Sie Ihre Lösung als Bash-Skript oder als C-Programm als freie Software (Open Source) und verwenden Sie hierfür ein Code-Repository, z.B. bei GitHub.

Bearbeiten Sie die Aufgabe in Teams zu **4 Personen**.

Schreiben Sie eine aussagekräftige und ansehnliche Dokumentation (Umfang: **8-10 Seiten**) über Ihre Lösung.

Bereiten Sie einen Vortrag mit Präsentationsfolien und eine Live-Demonstration (Umfang: **15-20 Minuten**) vor. Demonstrieren Sie die Funktionalität der Lösung in der Übung.

Aufgabe 2 Anforderungen an den Simulator

- Das fertige Programm soll eine Kommandozeilenanwendung sein.
- Der Quellcode soll durch Kommentare verständlich sein.
- Benutzer sollen das Scheduling-Verfahren FCFS, Round Robin mit frei definierbarem Zeitquantum und Multilevel-Feedback-Scheduling über eine Benutzereingabe (interaktiv) oder per Kommandozeilenargument definieren, also z.B. `-strategy FCFS` oder `-strategy FCFS -quantum 2`.
- Es soll auch möglich sein mehrere Scheduling-Verfahren gleichzeitig auszuwählen (als Benutzereingabe oder als Kommandozeilenargument) und so verschiedene Seitenersetzungsstrategien nacheinander zu simulieren.
- Benutzer sollen die Prozesse über eine einzulesende Eingabedatei definieren. Konkret sollen die Benutzer die Möglichkeit haben, per Kommandozeilenargument den Pfad und Dateinamen der Textdatei zu definieren, also z.B. `-processlistfile <dateiname>`.

- Jeder Prozess hat diese drei Attribute:
 - Name (oder alternativ Nummer)
 - CPU-Laufzeit
 - Ankunftszeit
- Die Ausgabe enthält die Ausführungsreihenfolge der Prozesse entweder als Gantt-Diagramm (Zeitleiste) oder in einer anderen geeigneten Form.
- Teil der Ausgabe sollen auch die Laufzeiten¹ und Wartezeiten² der einzelnen Prozesse sowie die durchschnittliche Laufzeit und durchschnittliche Wartezeit für jedes Scheduling-Verfahren sein, das ein Benutzer per Kommandozeilenargument definiert hat.
- Die Ausgabe des Simulators soll in eine Ausgabedatei (eine einfache Textdatei genügt) ausgegeben werden, deren Pfad und Dateiname die Benutzer frei definieren, also z.B. `-ausgabedatei <dateiname>`.

Aufgabe 3 Anforderungen an die Simulation des Multilevel-Feedback-Scheduling

Beim Multilevel-Feedback-Scheduling gibt es drei Warteschlangen. Innerhalb jeder dieser Warteschlangen wird **Round Robin** eingesetzt.

- Das Zeitquantum bei Warteschlange 1 ist 1 s.
- Das Zeitquantum bei Warteschlange 2 ist 2 s.
- Das Zeitquantum bei Warteschlange 3 ist 5 s.

Der Simulator implementiert keine Prozessprioritäten oder Zeitmultiplexe für die einzelnen Warteschlangen. Die Arbeitsweise ist wie folgt: Neue Prozesse werden nach ihrer Erzeugung in Warteschlange 1 eingereiht.

Wird ein Prozess von der CPU verdrängt, wird er in die nächst tiefere Warteschlange eingereiht. Tiefer als Warteschlange 3 geht es nicht.

Zuerst werden Prozesse in Warteschlange 1 gemäß ihrer Reihenfolge bearbeitet. Nur wenn Warteschlange 1 keine Prozesse enthält, werden die Prozesse in Warteschlange 2 gemäß ihrer Reihenfolge bearbeitet. Nur wenn die Warteschlangen 1 und 2 keine Prozesse enthalten, werden die Prozesse in Warteschlange 3 gemäß ihrer Reihenfolge bearbeitet.

¹Zeit [s] von der Prozesserzeugung bis zur Terminierung.

²Zeit [s], in der der Prozess im Zustand **bereit** ist, aber keinen Zugriff auf die CPU hat.

Aufgabe 4 Literatur

- Foliensatz 5 der Vorlesung **Betriebssysteme und Rechnernetze** im SS2022
- **Betriebssysteme kompakt**, *Christian Baun*, 2. Auflage, Springer Vieweg, S. 171-185
- **Betriebssysteme**, *William Stallings*, 4. Auflage, Pearson Studium (2003), S. 466-477
- **Betriebssysteme**, *Rüdiger Brause*, 4. Auflage, Springer (2017), S. 39-64
- **Betriebssysteme**, *Eduard Glatz*, 4. Auflage, dpunkt.verlag (2019), S. 160-170
- **Betriebssysteme**, *Carsten Vogt*, 1. Auflage, Spektrum Akademischer Verlag (2001), S. 65-67