

Aufgabe 1: Scheduler

In dieser Aufgabe sollen verschiedene Schedulevarianten implementiert werden. Für die Verwaltung der Jobs wird außerdem eine Queue benötigt, die von Ihnen ebenso erstellt werden soll. Ein einfaches FCFS Beispiel ist bereits vorgegeben. Beachten Sie auch die README innerhalb der Vorgabe!

- Erstellen Sie eine Queue. In den Dateien `queue.c` und `queue.h` sind dafür bereits die nötigen Funktionen und Structs vorgegeben. Beachten Sie die Dokumentation in der Headerdatei.
- Implementieren Sie anschließend die verschiedenen Scheduler, wie in den Vorgaben spezifiziert. Es sollen die Verfahren LCFS, HRRN, RR und MLF als Beispiele für nicht verdrängende und PRIO-P sowie SRTN-P als Beispiele für verdrängende Algorithmen erstellt werden. Studieren Sie dazu auch die `scheduler.c` und `scheduler.h` Dateien um sich einen Überblick zu verschaffen. Hinweis zum MLF-Verfahren: Es soll hier das MLF-Verfahren mit insgesamt 4 Leveln implementiert werden, wobei die jeweilige Zeitscheibe $\tau = 3^i$ beträgt. Das erste Level hat den Index $i = 0$, das nächste $i = 1$ und so weiter. Das letzte Level soll als mit FCFS behandelt werden, das heißt, dass Prozesse, die dieses Level erreichen und dann gescheduled werden, ihre gesamte Restlaufzeit abarbeiten können.
- Überprüfen Sie abschließend Ihr Programm auf Memory leaks, um zu evaluieren ob der gesamte vom Scheduler allokierte Speicher wieder freigegeben wurde. Hier empfiehlt sich das Kommandozeilenwerkzeug `valgrind`¹. Memory leaks führen zu Punktabzug.
- optional:** Erstellen Sie weitere Testcases. Auch hier können Sie sich an den Vorgaben orientieren.

Hinweise:

- Beschreibungen der Funktionen:** Nähere Beschreibungen der einzelnen Funktionen finden Sie auch in den `.h`-Dateien im Ordner `include`.
- Das Abgabearchiv** muss aus den Ordnern `src` und `lib` bestehen.
- Vorgaben:** Bitte ändern Sie bestehende Datenstrukturen, Funktionsnamen, ... nicht. Eine Missachtung kann zu Punktabzug führen. Gerne können Sie weitere Hilfsfunktionen oder Datenstrukturen definieren, die Ihnen die Implementierung leichter gestalten.
- Makefile:** Bitte verwenden Sie für diese Aufgabe das Makefile aus der Vorgabe. Führen Sie hierzu im Unterordner `src` der Vorgabe `make` aus. Durch den Befehl `make clean` werden kompilierte Dateien gelöscht. Gerne können Sie das Makefile um weitere Flags

¹<http://valgrind.org/>

erweitern oder anderweitig anpassen. Ihr Programm sollte aber mit dem vorgegebenen Makefile weiterhin compilierbar bleiben. Auch hier sei erneut auf die README hingewiesen.