Technische Universität Berlin Fachgebiet Komplexe und Verteilte IT-Systeme Sommersemester 2022	**Praxisaufgabe 1** zu – Systemprogrammierung – Prof. Dr. Odej Kao
Abgabetermin:	**16.06.2023 23:59 Uhr**

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!

Aufgabe 1: Queue (2P)

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.

Aufgabe 2: Scheduler (8P)

Implementieren Sie anschließend die verschiedenen Scheduler, wie in den Vorgaben spezifiziert. Es sollen die Verfahren PRIO-NP, SJN, HRRN, RR und MLF als Beispiele für nicht verdrängende sowie LCFSPR-PR als Beispiel 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 $\tau = i!$ (Fakultät) beträgt. Das erste Level hat den Index $i = 1$, das nächste $i = 2$ 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.

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.
- **Vorgaben:** Bitte ändern Sie bestehende Datenstrukturen, Funktionsnamen, Headerfiles, ... 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 Hauptverzeichnis das *make* aus. *make* kompiliert das Projekt mit *clang*. Unter Ubuntu können *make* und *clang* mit dem Befehl *sudo apt install make clang* installiert werden. Durch den Befehl *make clean* werden kompilierte Dateien gelöscht.

Gerne können Sie das Makefile um weitere Flags erweitern oder anderweitig anpassen. Ihr Programm sollte aber mit dem vorgegebenen Makefile weiterhin compilierbar bleiben. Auch hier sei erneut auf die README hingewiesen.

- **Memoryleaks:** 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.

Abgabe:

- **Verpacken** sie den Ordner *src* mit den bearbeiteten **.c* Dateien (wie in der Vorgabe) zu einem zip-Archiv mit dem Namen *submission.zip*. Header-files (**.h*) sollen nicht verändert und damit auch nicht mitabgegeben werden. Hierfür können sie das make target *make pack* verwenden. Damit *make pack* funktioniert muss das tool *zip* auf ihrem System installiert sein! Das Archiv laden sie dann in ISIS hoch.
- **Wichtig:** Es ist nicht notwenig das Archiv und dessen Inhalt mit ihrem Namen bzw. Matrikelnummer zu personalisieren. Die Abgabe wird automatisch ihrem ISIS Account zugeordnet!

[1] http://valgrind.org/