
Multi-Threading und Synchronisation (10 Punkte)

Sie sollen in dieser Aufgabe ein Programm entwickeln, das mehrere einzelne Dateien einliest, komprimiert und als neue Datei speichert. Lesen Sie sich das beiliegende Hilfsdokument durch, das nähere Erklärungen und Beispiele zu Threads (Seite 1-8), Dateikomprimierung (Seite 9-10) und der beiliegenden Queue (Seite 11) enthält.

Aufgabenstellung

- Dem Programm wird per Kommandozeile ein Ordnerpfad übergeben. (1 Punkt)
- Arbeiten Sie mit mehreren Threads. Ein Thread (Leser-Thread) liest die Dateien in dem übergebenen Ordner ein und hängt den Inhalt sowie den Dateipfad an eine Queue an. Nutzen Sie dafür eine `struct Job`. (2 Punkte)

In den beigelegten Dateien befindet sich bereits eine Queue-Implementierung, die sie benutzen können.

- Eine zur Compilezeit konfigurierbare Anzahl von Threads (Komprimierungs-Thread) liest jeweils einen Job aus der Queue, komprimiert seinen Inhalt und speichert diesen im Format `<alter Dateiname>.compr`. Dies soll solange wiederholt werden, bis die Queue leer und der Leser-Thread beendet ist. (2 Punkte)
- Der Leser-Thread soll Dateien, die mit `.compr` enden, ignorieren. (1 Punkt)
- Ein Komprimierungs-Thread bekommt bei seiner Erstellung als Parameter eine Instanznummer zugeordnet, die ihn z.B. bei Debug-Ausgaben eindeutig identifiziert. (1 Punkt)
- Die Zugriffe auf die Queue müssen synchronisiert, d.h. gegeneinander geschützt sein. (1 Punkt)
- Fügen Sie im Leser-Thread nach dem Einlesen einer Datei ein `sleep(1)` und in den Kompressions-Threads nach dem Komprimieren ein `sleep(3)` ein, um eine langsame Festplatte und einen komplexen Kompressionsalgorithmus zu simulieren. Beobachten Sie, wie ihr Programm mit und ohne die `sleep`-Anweisungen arbeitet. (1 Punkt)
- Bestimmen Sie näherungsweise die Laufzeit bei unterschiedlicher Anzahl von Kompressions-Threads, z. B. mit der Funktion `difftime()` in `main()`. (1 Punkt)

Hinweis: Gesucht ist eine Lösung, die mit POSIX-konformen Threads arbeitet. Eine Lösung auf Basis von Shellskripts kann nicht gewertet werden. Die Dateien `miniz.c`, `miniz.h`, `queue.h` müssen sich in Ihrem Projekt befinden. Inkludieren Sie die beiden Header-Dateien.