



Parallele Programmierung mit C++ (SS 2019)

Abgabe bis zum 17. Mai 2019, 14:00 Uhr

Lernziele:

- Aufteilung und Parallelisierung eines Problems mit Hilfe des Fork-and-Join-Patterns

Aufgabe 4: Parallele Suche

Die Suche einer Zeichenkette in einer größeren Datei lässt sich gut parallelisieren, indem der Dateiinhalt auf n Threads aufgeteilt wird, die jeweils ihren Bereich durchsuchen.

Zu erstellen ist ein Werkzeug namens *pgrep*, das drei Parameter erhält: Die Zahl der zu verwendenden Threads, einen regulären Ausdruck und die Eingabedatei. Auszugeben sind in sortierter Reihenfolge alle Zeilen, für die der reguläre Ausdruck zutrifft, jeweils mit der zugehörigen Zeilennummer.

Hinweise: Um die Suche einfach und effizient zu gestalten, empfiehlt es sich, den Dateiinhalt mit *mmap* in den eigenen Adressraum abzubilden. Wenn Sie möchten, können Sie hierfür die zur Verfügung gestellte Klasse *MappedFile* aus *mapped_file.hpp* verwenden.

Für die regulären Ausdrücke empfiehlt sich die Verwendung der entsprechenden Bibliothek, die mit **#include** <regex> zur Verfügung steht.

Wenn Sie den Dateiinhalt auf n Threads aufteilen, sollten Sie darauf achten, dass Sie den einzelnen Threads jeweils ganze Zeilen geben. Beachten Sie auch den Sonderfall, dass Sie weniger Zeilen als Threads haben.

Da Sie nicht von vorneherein wissen, wieviele Zeilen die einzelnen Teilbereiche umfassen, sollte jeder Thread zunächst mit relativen Zeilennummern arbeiten. Erst bei der Aggregation können Sie diese in absolute Zeilennummern konvertieren.

Organisieren Sie Ihr Programm so, dass jeder Thread seine Ergebnisse in einer eigenen Datenstruktur erfasst. Erst nach der Join-Phase sollten die Ergebnisse vom Hauptthread ausgewertet und aggregiert werden.

Packen Sie all Ihre Quellen mitsamt Ihrem *Makefile* in ein kleines *tar*-Archiv ein, das Sie dann einreichen können:

```
tar cvf pgrep.tar *.*pp [mM]akefile  
submit pp 4 pgrep.tar
```

Viel Erfolg!