

THREADS + SPEEDUP

In dieser Aufgabe geht es um die Verbesserung einer Suche nach Wörtern. Aus dem Buch **Lés Miserables** wurden alle Wörter extrahiert und dann jeweils miteinander kombiniert. Um bei der Suche nach solchen Zweier-Kombinationen nicht Strings vergleichen zu müssen wurden diese Kombinationen mit der **Hashfunktion MD5** abgespeichert. Leider wurde dies nur in eine Richtung gemacht und wir würden gerne wissen welche beiden Wörter zu dem Hash **e65493ccdee9c4514fe20e0404f3bcb8** gehören. Zum Schluss soll diese Suche durch die Verwendung von parallelen Threads noch weiter beschleunigt werden.

Folgende Schritte sollen für diese Aufgabe erledigt werden:

1. Einlesen des Buches Lés Miserables (kann von der Webseite herunter geladen werden oder von Projekt Gutenberg).
2. Alle Wörter die mindestens einmal im Text vorkommen abspeichern. Um dies zu vereinfachen kann man `strtok()` mit folgendem delimiter: " . , ; - : 0123456789 ? ! \ " * + () | & [] # \$ / % ' " verwendet werden. Achtet darauf dass ihr keine Steuerzeichen wie newlines (`\n`) oder Tabulatoren (`\t`) mit euren Wörtern abspeichert. Diese verändern den resultierenden Hash.
3. Testen alle möglicher Zweier-Wortkombinationen (Kreuzprodukt) mittels Generierung des MD5 hashes. Jede Kombination besteht aus den **beiden Wörtern und einem Leerzeichen** dazwischen. Zur Berechnung des Hashes gibt es auf der Website eine MD5 Bibliothek die eingebunden werden kann. Alle drei Funktionen aus `md5.h` werden benötigt. Zuerst wird MD5 initialisiert (`MD5_Init`), danach werden die Daten miteinbezogen (`MD5_Update`) und zum Schluss das Ergebnis ermittelt (`MD5_Final`). Zum Beispiel ergibt der Hash von "Les Miserables" den hash `1c0143c6ac7505c8866d10270a480dec`.
4. Wenn nun die Suche erfolgreich ist soll die Zeit des Programmes gemessen werden, dafür bietet sich die `time.h` Bibliothek mit der Funktion `time()` an.
5. Um diese Laufzeit zu verkürzen kann man nun auf OpenMP zurückgreifen. Versucht euren Code mittels `#pragma` Anweisungen zu beschleunigen.

Auf der Website findet ihr ein Code-Skelet auf das ihr eure Lösung aufbauen könnt.

Da das parsen und verarbeiten des ganze Buches sehr lange dauern kann, könnt ihr euer Programm mit dem Vorwort testen. Das Finden der Wörter "Les Miserables" anhand ihres Hashes (siehe 3.) sollte nahezu ohne Verzögerung laufen.