

Algorithmen und Datenstrukturen in der Bioinformatik

Vierte Praktikumsaufgabe WS 12/13

Abgabe Montag, 10.12., 15:00 Uhr

Paralleles Pattern-Matching

Implementieren Sie den Shift-And Algorithmus (aus Aufgabe P-2) parallelisiert mit OpenMP. Schreiben Sie dazu ein Program, das:

- den zu durchsuchenden Text T aus einer Datei einliest,
- ein Pattern mittels parallelisiertem Shift-And Algorithmus in T sucht (**4 Punkte**),
- jeweils Suchlaufzeit in Millisekunden und Anzahl der gefundenen Treffer durch Kommas getrennt ausgibt (**1 Punkt**) und
- die Suchergebnisse aufsteigend sortiert nach ihrer Startposition in T und zeilenweise in der Datei *results.txt* speichert (**1 Punkt**).

Als drittes Argument wird nun die Zahl der zu benutzenden Threads und als viertes Argument der Dateiname für die Suchergebnisse übergeben. Es soll die reine Suchlaufzeit (ohne Dateizugriffe und ohne Sortierung, siehe unten) mit `omp_get_wtime` gemessen werden und in Millisekunden zusammen mit der Zahl der Treffer ausgegeben werden. Beispiel:

```
rmaerker@peking:~$ ./aufgabe4 english.50MB whatever 1 results.txt
1000, 687
rmaerker@peking:~$ ./aufgabe4 english.50MB whatever 2 results.txt
500, 687
rmaerker@peking:~$ ./aufgabe4 english.50MB whatever 4 results.txt
250, 687
rmaerker@peking:~$
```

Hinweise: Für die Suchergebnisse soll ein `std::vector<unsigned long>` verwendet werden, der die Startpositionen (nullbasiert) der Matches im Text speichert. Am Ende der Suche sollen diese Textpositionen sortiert und untereinander in die Ergebnisdatei geschrieben werden (siehe `std::sort` und `std::ofstream`). Bsp:

```
rmaerker@peking:~$ head results.txt
46909
60671
71667
74688
96565
248812
251671
252341
260710
260784
rmaerker@peking:~$
```

Beachten Sie die Praktikumshinweise unter <https://www.mi.fu-berlin.de/w/ABI/AlDaBiWS12>.