

— Grundlagen der Betriebssysteme WiSe 2016/17 —

1. Übungsaufgabe

Ausgabe: 06.11.2017

Abgabe: 14.11.2016

1 C++11 Thread-Bibliothek

Punkte: 12

In der Vorlesung Grundlagen der Betriebssysteme vom 30.10.2017 wurde auf der Basis der POSIX-pthread-Bibliothek ein Programm vorgestellt, bei dem abhängig von der Konstanten `THREADS` leichtgewichtige Prozesse (Threads) erzeugt werden. Diese arbeiten auf einem Integer-Array der Länge `DATA_LENGTH`. Jeder Thread hat darin einen eigenen Abschnitt, in dem er die Zahl der gesetzten Bits zählt.

- Schreiben Sie dieses Verfahren in der Sprache C in eines der Sprache C++ Version 11 um. Benutzen Sie dabei die Klasse `std::thread`, die seit dieser Version Bestandteil der Sprache ist und erzeugen Threads, die in analoger Weise auf dem Array arbeiten. Als Vorlage nutzen Sie das unten aufgeführte Listing.
- Messen Sie die Laufzeit des Programms mit Hilfe von Funktionen aus der `<chrono>` Bibliothek.
- Prüfen Sie außerdem mit Hilfe einer bestimmten Funktion aus der `<thread>` Bibliothek wie viele Kerne ihr Rechner zur Verfügung stellt. Lässt sich anhand von Messungen feststellen, dass die Laufzeit des Programms mit der Anzahl der Threads in einem Zusammenhang steht?
- In das Programm hat sich ein kleiner Fehler eingeschlichen. Für bestimmte Belegungen der `THREADS` Variablen liefert das Programm falsche Ergebnisse. Finden und beheben Sie den Fehler.
- Finden Sie heraus, weshalb das Programm bei jeder Ausführung immer den selben Wert liefert, obwohl das Array mit einem scheinbar zufälligen Inhalt befüllt wird.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <pthread.h>
4
5 #define THREADS 1
6 #define DATA_LENGTH 5000000
7 #define LEN_PER_THREAD DATA_LENGTH / THREADS
8
9 unsigned int data[DATA_LENGTH];
10
11 unsigned int sum(unsigned int nr){
12     unsigned int start, sum, i, j, d;
13
14     sum = 0;
15     start = nr * LEN_PER_THREAD;
16     printf("Start of thread %d\n", nr);
17     for(i = start; i < start + LEN_PER_THREAD; i++){
```

```
18     d = data[i];
19     for(j = 0; j < sizeof(unsigned int); j++){
20         if ((d%2) == 1)
21             ++sum;
22         d >>= 1;
23     }
24 }
25 printf("End of thread %d\n", nr);
26 return sum;
27 }
28
29 void* work (void* arg){
30     return (void*) sum ((unsigned int) arg);
31 }
32
33 int main(){
34     unsigned int i, sum, ok, t;
35     pthread_t tid[THREADS];
36     void* ret;
37
38     for(i = 0; i < DATA_LENGTH; i++){
39         data[i] = random() % 10;
40     }
41
42     for(t = 0; t < THREADS; t++){
43         pthread_create(&tid[t], NULL, &work, (void*)t);
44     }
45
46     sum = 0;
47     for(t = 0; t < THREADS; t++){
48         pthread_join(tid[t], &ret);
49         sum += (unsigned int) ret;
50     }
51
52     printf("Die Berechnung ergibt: %d\n", sum);
53     return 0;
54 }
```