

## Rechnerarchitektur

### 2. Übung

---

In dieser Übung untersuchen wir einfache Möglichkeiten die Leistung (Performance) von einem Programm zu untersuchen, ohne den Quellcode zu instrumentieren (ohne Code für Performance Messungen).

#### Aufgabe 1

Implementieren Sie ein C Programm zum Sortieren von `int` Werten. Dazu sind folgende Schritte notwendig:

1. Implementieren Sie eine Funktion `void bubbleSort(int* array, int numElements)` zum Sortieren eines Arrays in aufsteigender Reihenfolge von `numElements` `int` Werten. Nutzen Sie dazu das Sortierverfahren Bubble Sort. Hinweis: Im Internet gibt es bereits Implementierungen von Bubble Sort.
2. Implementieren Sie die Funktion `void initArray(int* array, int numElements)` zum Initialisieren eines Arrays mit zufälligen Werten. Verwenden Sie dazu die Funktion `lrand48()` im Header File `stdlib.h` (z.B. `array[i] = lrand48() % numElements`).
3. Implementieren Sie die Funktion `void printArray(int* array, int numElements)`, mit der Sie bei Bedarf das Array ausgeben können.
4. Implementieren Sie eine Funktion `bool isSorted(int* array, int numElements)`, die `true` zurück gibt, wenn das Array aufsteigend sortiert ist.
5. Implementieren Sie nun das komplette Programm, was den Speicher eines `int` Arrays dynamisch allokiert (`malloc()`), das Array zufällig initialisiert, sortiert und am Ende prüft, ob das Array tatsächlich korrekt sortiert ist.

#### Aufgabe 2

Führen Sie ihr Bubble Sort Programm mit dem Linux Performance Analyse Tool Perf aus, um folgende Metriken für die Ausführung ihres Programms zu bestimmen:

- Ausführungszeit (execution/elapsed time)
- Anzahl Takte (cycles)
- Anzahl Instruktionen
- Instruktionen pro Takt (instructions per cycle)
- Takte pro Instruktion (cycles per instruction)

Perf können Sie zum Beispiel wie folgt aufrufen:

```
perf stat -d -e cycles:u -e instructions:u <Programm Name>
```

Zum Mehrmaligen Ausführen ihres Programms können Sie den Parameter `-r <Anzahl Ausführungen>` nutzen. Sie erhalten dann zusätzlich für die Metriken den statistischen Parameter der Standardabweichung.

Wie verhalten sich Ausführungszeit, Anzahl Takte und Anzahl Instruktionen, wenn Sie ihr Programm mehrmals ausführen? Versuchen Sie die Beobachtung zu erklären.