

# Aufgabe 1

## 1. Programm

Ein *Programm* ist eine Datei oder ein Satz von Dateien, der Quellcode und Anweisungen enthält, die eine bestimmte Funktion oder Anwendung darstellen. Programme werden geschrieben, aber sie sind im Ruhezustand, bis sie ausgeführt werden. Solange ein Programm nicht gestartet wird, bleibt es eine passive Ressource auf der Festplatte. Beispiele für Programme sind Textverarbeitungen, Webbrowser oder auch systemnahe Programme wie der Dateimanager.

Zusammengefasst:

- Ein Programm ist eine inaktives Set von Anweisungen.
- Es wird auf der Festplatte gespeichert und ist noch nicht im Arbeitsspeicher geladen.

## 2. Prozess

Ein *Prozess* entsteht, wenn ein Programm ausgeführt wird. Beim Start eines Programms lädt das Betriebssystem den Programmcode und die benötigten Ressourcen (z. B. Speicherplatz) in den Arbeitsspeicher. Ab diesem Moment wird das Programm zu einem Prozess, der aktiv vom Betriebssystem verwaltet wird. Jeder Prozess hat seinen eigenen Speicherbereich, in dem er Daten und Anweisungen verarbeitet, und läuft in einer isolierten Umgebung, was bedeutet, dass er nicht direkt auf die Ressourcen anderer Prozesse zugreifen kann. Diese Isolation dient der Sicherheit und Stabilität des Systems.

Zusammengefasst:

- Ein Prozess ist eine aktive Instanz eines Programms, das vom Betriebssystem verwaltet wird.
- Jeder Prozess hat seinen eigenen Speicherbereich und ist isoliert von anderen Prozessen.
- Prozesse können unabhängig voneinander arbeiten, aber sie können auch miteinander kommunizieren (z. B. über Interprozesskommunikation, IPC).

## 3. Thread

Ein *Thread* ist die kleinste Ausführungseinheit innerhalb eines Prozesses. Threads teilen sich den gleichen Speicherbereich und Ressourcen des Prozesses, in dem sie laufen. Ein Prozess kann einen oder mehrere Threads enthalten. Der Vorteil mehrerer Threads innerhalb eines Prozesses ist, dass diese parallel arbeiten können, um Aufgaben effizienter zu verarbeiten. Beispielsweise könnte ein Webbrowser einen Thread für die Benutzeroberfläche, einen anderen für die Netzwerkkommunikation und einen dritten für die Verarbeitung von Seiteninhalten verwenden.

Zusammengefasst:

- Ein Thread ist eine Ausführungseinheit innerhalb eines Prozesses.
- Threads innerhalb desselben Prozesses teilen sich Speicher und Ressourcen, wodurch die Kommunikation zwischen ihnen schneller ist als zwischen getrennten Prozessen.
- Multithreading ermöglicht parallele Verarbeitung innerhalb eines einzelnen Prozesses.

## Aufgabe 2

Das Amdahlsche Gesetz beschreibt den Speedup  $S$  eines Programms, das auf mehreren Kernen parallel ausgeführt werden kann, wenn nur ein Teil des Programms parallelisierbar ist. Der Speedup kann mit folgender Formel berechnet werden:

$$S = \frac{1}{(1 - p) + \frac{p}{N}}$$

Dabei ist:

- $p$  der parallelisierbare Anteil des Programms,
- $N$  die Anzahl der verwendeten Kerne.

In dieser Aufgabe variieren wir  $p$  und  $N$  und berechnen den Speedup für verschiedene Kombinationen.

### Gegebene Werte

#### 1. Parallelisierbarer Anteil $p$ :

- $p = 0,25$  (25%)
- $p = 0,50$  (50%)
- $p = 0,75$  (75%)

#### 2. Anzahl der Kerne $N$ :

- Die Anzahl der Kerne wird als  $N = 2^x$  berechnet mit  $x = 0$  bis  $x = 3$ .
- Das bedeutet, dass die möglichen Werte für  $N$  sind: 1, 2, 4 und 8.

Im Folgenden sind die berechneten Speedups für verschiedene Werte des parallelisierbaren Anteils  $p$  und der Anzahl der Kerne  $N$  dargestellt:

- Für  $p = 0,25$  (25% parallelisierbar):

- $N = 1: S = 1,0$
- $N = 2: S \approx 1,14$
- $N = 4: S \approx 1,23$
- $N = 8: S \approx 1,28$

- Für  $p = 0,50$  (50% parallelisierbar):

- $N = 1: S = 1,0$
- $N = 2: S \approx 1,33$
- $N = 4: S \approx 1,6$
- $N = 8: S \approx 1,78$

- Für  $p = 0,75$  (75% parallelisierbar):

- $N = 1: S = 1,0$
- $N = 2: S \approx 1,6$
- $N = 4: S \approx 2,29$
- $N = 8: S \approx 2,91$

Diese Werte zeigen, dass der Speedup mit einem höheren parallelisierbaren Anteil und einer größeren Anzahl von Kernen zunimmt, jedoch nur bis zu einem gewissen Punkt, wie durch das Amdahlsche Gesetz vorhergesagt.

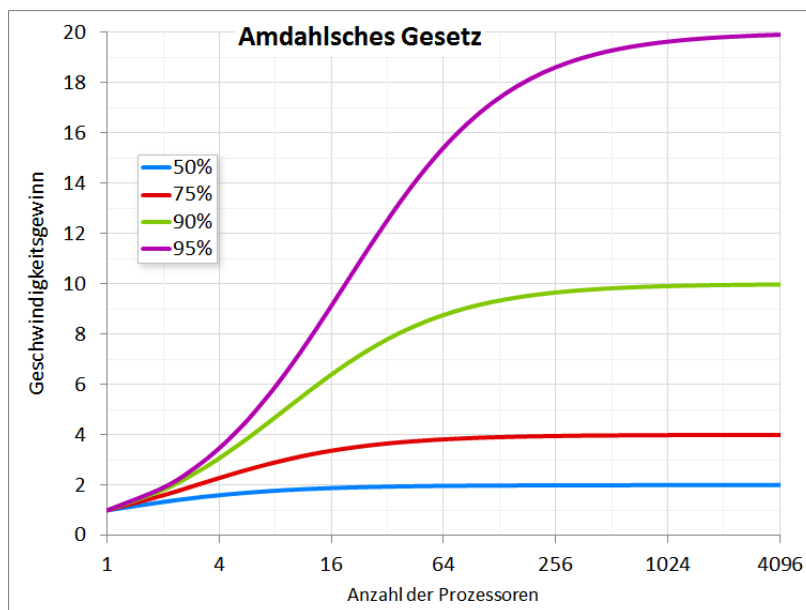


Abbildung 1: Amdahlsches Gesetz für verschiedene Parallelisierbarkeiten

## Aufgabe 3

Branch: s4mrlang