**Übungszettel 1**

**Aufgabe 1:**

1. Nennen Sie drei unterschiedliche Einsatzbereiche für Parallelisierung.
2. Was sind die Voraussetzungen für „Parallel Processing“?
3. Erklären Sie den Begriff „Speedup“.

**Aufgabe 2:**

Erläutern Sie zwei Gemeinsamkeiten und Unterschiede der TM, RAM und des v. Neumann Konzepts.

**Aufgabe 3:**

Betrachte das folgende Programm 1.

Programm 1

Read 0

JGTZ gr\_0

Write =1

Halt

gr\_0 Store 1

Sub =1

Store 2

JGTZ loop

Write =1

Halt

loop Load 1

Mult 2

Store 1

Load 2

Sub =1

Store 2

JGTZ loop

Write 1

Halt

1. Welche Berechnung wird mit dem Programm durchgeführt und welches Konzept steckt dahinter?
2. Wie sieht der Verlauf der Register aus, wenn das Programm aufgerufen wird?

**Übungszettel 2**

**Aufgabe 1:**

1. Welche Thesen zum Speedup wurden in der Vorlesung vorgestellt?
2. Wie sind die Thesen bzgl. des Speedup Faktors einzuordnen?
3. Geben Sie Möglichkeiten an, die Thesen zu erweitern.

**Aufgabe 2:**

Das Gesetz von *Amdahl* beschäftigt sich mit Aussagen über die parallelen und sequenziellen Anteile eines Programms. Hieraus lässt sich der bereits besprochene Speedup herleiten, den ein Programm durch eine mögliche Parallelisierung maximal erzielen kann.

1. Warum kann die Ausführungszeit eines Programms durch Hinzufügen von *N* Prozessoren nicht beliebig verringert werden?
2. Welcher ist momentan der wesentlichste Faktor, der die lineare Skalierung bezüglich der Ausführungszeit verhindert?
3. Nennen Sie drei unterschiedliche Interconnect-Varianten bei Parallelrechnern. (Die Antworten können sich auf die Topologie als auch die verwendete Technik beziehen.)

**Aufgabe 3:**

Die Taxonomie von *Flynn* teilt Rechnerarchitekturen in vier Klassen ein:

**SISD / MISD / SIMD / MIMD**

1. Was bedeuten die vier Akronyme?
2. Falls möglich, finden Sie zu jeder Klasse zwei real existierende Beispiele.
3. Gibt es noch andere Taxonomien für Parallelrechner? Wenn ja, welche?

**Übungszettel 3**

In dieser Übung sollen einige Grundlagen zur parallelen Programmierung in der Programmiersprache Java kennengelernt und ausprobiert werden.

**Aufgabe 1:**

1. Moderne Betriebssysteme unterstützen die Nebenläufigkeit in Prozessen und Threads. Beschreiben Sie den Unterschied, Vorteil und Nachteil von Prozessen und Threads im Kontext der Programmiersprache Java.
2. Programmieren Sie eine kleine Java-Anwendung mit folgenden Schritten:  
   - 10 Threads werden in einer „for“-Schleife mit der „Thread“-Klasse gestartet  
   - jeder Thread gibt zu Beginn der Ausführung den Text „Start“ + Threadname aus  
   - jeder Thread wartet anschließend für 5 Sekunden  
   - jeder Thread gibt zum Schluss den Text „End“ + Threadname aus
3. Starten Sie das Programm und beschreiben Sie die Ausgabe. Was fällt Ihnen auf?

**Aufgabe 2:**

1. Neben den einfachen Threads bietet die Java Concurrency-API weitere Möglichkeiten zur parallelen Programmierung. Beschreiben Sie was ein „Thread-Pool“ ist und welche Vorteile er gegenüber normalen Threads bietet.
2. Machen Sie sich mich dem „ExecutorService“ in Java vertraut und programmieren Sie das Beispiel aus Aufgabe 1 mit einem ExecutorService Thread-Pool der Größe 10.
3. Welchen Unterschied bemerken Sie, wenn der Thread-Pool mit einer Größe von 2 erstellt wird und warum?

**Aufgabe 3:**

Die bisherigen Methoden führen den Code lediglich einmal im Thread aus. Manchmal ist es allerdings sinnvoll den gleichen Code in periodischen Abständen auszuführen.

1. Geben Sie Beispiele bei denen diese periodische Ausführung sinnvoll wäre.
2. Auch hierfür gibt es in Java eine Implementierung: „ScheduledExecutorService“.
3. Programmieren Sie eine kleine Java-Anwendung, welche mit Hilfe des ScheduledExecutorService alle 5 Sekunden die aktuelle Zeit in der Form „hh-mm-ss“ ausgibt.

**Übungszettel 4**

**Aufgabe 1:**

**Übungszettel 5**

**Aufgabe 1:**

**Übungszettel 6**

**Aufgabe 1:**

**Übungszettel 7**

**Aufgabe 1:**

**Übungszettel 8**

**Aufgabe 1:**

**Übungszettel 9**

**Aufgabe 1:**

**Übungszettel 10**

**Aufgabe 1:**