

## Aufgabe 4: Maximale Thread-Anzahl

Jeder Thread braucht seinen eigenen Stack. Der kostet je nach VM Settings typischerweise zwischen 64KB oder 1MB virtueller Speicher, wobei natürlich nicht der gesamte virtuelle Speicher des Threads auch in Realspeicher alloziert ist. Für die Java Hotspot JVM wird als Default 1MB virtueller Speicher für 64-Bit Systeme genommen, ansonsten die native Stackgrösse des Betriebssystems. Die Stackgrösse lässt sich in Java auch beim Thread-Konstruktor als Parameter angeben.

### 64-Bit JVM:

In einem 64-Bit JVM ist der reale Speicher die beschränkende Grösse: Für 8GB Main Memory lassen sich systemabhängig ca. 150'000 Threads starten. Dann wird das System sehr langsam und reagiert quasi nicht mehr. Die Speicheransicht im „Resource Monitor“ zeigt, dass z.B. bei 110'000 Threads ca. 7GB Realspeicher für das Java Programm alloziert (committed) sind, d.h. ungefähr 64KB Realspeicher pro Java Thread. Das bringt das System mit 8GB an die Speichergrenzen und macht es wegen des Page Swapping extrem langsam (sog. Trashing).

### Historisch: 32-Bit JVM:

In einer 32-Bit JVM ist der virtuelle Speicher die beschränkende Grösse (es sei denn es gibt weniger als 4GB Hauptspeicher). Die maximale Anzahl Threads ist bedeutend tiefer (Grössenordnung 2'300). Die JVM stoppt z.B. mit der Exception `OutOfMemoryError`: „unable to create new native threads“.