

2. Übung Praktische Informatik

1. Lesen Sie (die angegebenen Teile von) *What Every Programmer Should Know About Memory* (URL: www.akkadia.org/drepper/cpumemory.pdf) und beantworten Sie die Fragen. Der Text ist sehr technisch, nicht verzweifeln.
 - (a) Lesen Sie Kapitel 2, 2.1, 2.2.4 und 2.2.5. Erklären sie die Abkürzungen SMP und NUMA.
 - (b) Lesen Sie Kapitel 3, 3.1, 3.2. Was ist die *von Neumann* Architektur? Was kostet ein Zugriff auf ein Register bzw. auf den Hauptspeicher? Was versteht man unter *eviction*?
 - (c) Lesen Sie Kapitel 4, 4.1, 4.2. Wofür braucht man virtuellen Speicher? Warum wird mehrstufige *address translation* verwendet?
 - (d) Lesen Sie Kapitel 6.2, 6.2.1 (bis Seite 51) und probieren Sie den angegebenen Code zur Matrix-Multiplikation aus. Messen Sie mit `gettimeofday()` und vergleichen Sie das Ergebnis mit dem Artikel. Was versteht man unter *locality*. Was versteht man unter *alignment*?
2. Implementieren zwei Programme, die Daten über ein Mittel der Interprozesskommunikation austauschen (wählen Sie: *shared memory* oder *pipes* oder *unix domain sockets*). Entweder unter MacOS, Linux, oder Windows. Die zwei Programme sollen eine einfache Aufgabe kooperativ lösen; blödes Beispiel: ein Prozess wählt eine Zufallszahl, der zweite berechnet die Wurzel und gibt das Ergebnis an den ersten Prozess zurück.
3. Wo wird Interprozesskommunikation praktisch eingesetzt? Und warum? Was sind die Vorteile? Am besten Sie beschreiben die Architektur eines Open-Source Projekt, dass Interprozesskommunikation verwendet...
4. Vorbereiten: Implementieren und testen Sie die Operationen *search()*, *insert()* der *skiplist* in C++. Ihre skiplist soll auch ihren gesamten Speicher freigeben können. Kümmern Sie sich zuerst darum, wie die Klasse aussehen kann bevor Sie die Details implementieren. Wie kann die Klasse getestet werden? Wie kann die Klasse möglichst wiederverwendbar gestaltet werden?