* 1. Lokalität und Speicherhierachie

PowerPoint

描述已自动生成Ordnen Sie die Speicherhierachie nach Zugriffsgeschwindigkeit (oben - langsam, unten - schnell)

1.2

Das von-Neumann-Prinzip "Ein Programm ist eine Folge von Befehlen, die in der Regel in der gegebenen Reihenfolge ausgeführt werden" unterstützt das Prinzip der ...

Örtlichen Lokalität

1.3

Eine funktionale Programmiersprache (keine Schleifen, keine Variablen) unterstützt das Prinzip der ...

Örtlichen Lokalität

1.4

Eine Algorithmus, der mehrmals über ein Array iteriert, unterstützt das Prinzip der ...

Örtlichen Lokalität

Zeitlichen Lokalität

1.5

Ein Speicher soll durch ein Cache beschleunigt werden, der 0,5% der Größe des Hauptspeichers hat.

Ist der Cache-Zugriff ein Hit ist er 20mal schneller als ohne Cache. Jedoch wird durch die Zusätzliche Cache-Logik der Zugriff auf den Hauptspeicher im Miss-Fall um 2% langsamer als ohne Cache.

Das Verhältnis der mittleren Zugriffszeit **mit** Cache zur mittleren Zugriffzeit des gleichen Speichersystems **ohne** Cache ist

**1.01515**, wenn jeder Zugriff auf eine Speicherstelle gleichwahrscheinlich ist

**0.535**, wenn jeder Zugriff mit 50%iger Wahrscheinlichkeit ein Hit ist.

1.6

Ein Zugriff auf eine höhere Ebene der Speicherhierachie als die Adressierte ist **Caching**, auf eine tiefere Ebene ist **Virtualisierung**.

Caching dient zur **Beschleunigung des Zugriffs**, Virtualisierung dagegen zur Vergrößerung des Speicherraums.

Durch Caching und Virtualisierung können Annahmen über **die Speicherpersistenz** gebrochen werden, was u.a. ein **Sicherheits**-Risiko birgt.

2.1 Virtualisierung

Am häufigsten wird in modernen Systemen zur Lösung des Problems der Speicherknappheit eingesetzt:

Demand Paging

2.2

Das **Präsenzbit** gibt an, ob sich eine bestimmte Speicherseite in einem Speicherrahmen befindet. Es befindet sich **im erweiterten Eintrag in der Seitentabelle**. Erfolgt ein Zugriff bei nichtgesetztem Bit, wird durch **die MMU** **ein Seitenfehler** ausgelöst.

Nach jedem erfolgreichen schreibenden Zugriff auf eine Speicherseite ist ihr Präsenzbit auf **wahr** und ihr Modifikationsbit auf **wahr**

2.3

Ein System unterteilt den physischen Hauptspeicher in vier Kacheln, welche zu Beginn leer sind. Es finden Zugriffe auf Seiten in der folgenden Reihenfolge statt:

1-2-3-4-5-1-2-5-3-4-2-1-3

Ergänzen Sie für das Ersetzungverfahren LRU (Least Recently Used) die Belegung der Kacheln nach jedem Seitenzugriff. Tragen sie dafür die Seitennummern in die jeweiligen Kästchen ein. Markieren sie zusätzlich die Zeitpunkte, bei denen ein Seitenzugriffsfehler (page fault) auftritt.

**Hinweise**:

* Das Einlagern in eine leere Kachel wird **nicht** als Seitenzugriffsfehler gewertet.
* Die Speicherrahmen werden bei unechten Seitenzugriffsfehler in aufsteigender Nummerierung gefüllt

表格

描述已自动生成

2.4 Konkrete Systeme

**BSD**

* nutze ein Algorithmus mit **zwei** umlaufenden Zeigern.
* **Der Abstand der Zeiger** wird angepasst entsprechend **der Last**.

**Windows**

* Windows nutzt eine **lokale** Seitenersetzungstrategie.
* Bei Verkleinerung des Worksets erfolgt eine Auslagerung nach der **LRU**-Strategie