Betriebssysteme

Tutorium 9

Philipp Kirchhofer
philipp.kirchhofer@student.kit.edu
http://www.stud.uni-karlsruhe.de/~uxbtt/

Lehrstuhl Systemarchitektur
Universität Karlsruhe (TH)

13. Januar 2010

Was machen wir heute?

1. Organisatorisches

2. Tutorien Übungsblatt
   - Caches
   - Paging Basics
   - HW- vs. SW-Walked Page Tables
   - Page Replacement Strategies
   - Comparison of Replacement Policies

Organisatorisches

Programmieraufgaben Vorstellung

Montag, den 18. Januar, 14:00 Uhr bis 15:30 Uhr

Poolraum G im Rechenzentrum (Gebäude 20.21)

Lösung der VM Aufgabe sowie das Design-Dokument vorstellen.

Wichtig: Beide Gruppenteilnehmer müssen vorstellen!

Tutorien Übungsblatt - Caches

Frage 9.1.a

Welche Arten von Cache Misses gibt es?

Tutorien Übungsblatt - Caches

Frage 9.1.a

Welche Arten von Cache Misses gibt es?

Antwort

- Compulsory miss
  Der Datenblock war noch nicht vorher im Cache
- Capacity miss
  Der Cache ist zu klein um alle benötigten Daten vorhalten zu können
- Conflict miss
  Verschiedene Datenblöcke werden auf die gleiche Cachezeile gelegt

Frage 9.1.b

Was ist der Unterschied zwischen der write-through und der write-back Strategie?

Tutorien Übungsblatt - Caches

Frage 9.1.b

Was ist der Unterschied zwischen der write-through und der write-back Strategie?

Antwort

Write-Through

Jeder Schreibzugriff auf einen Speicherbereich aktualisiert sowohl den Cache als auch den Hauptspeicher. Der Hauptspeicherinhalt ist dadurch immer aktuell, es gibt keine Inkonsistenzen zwischen dem Cache und dem Hauptspeicher.

Write-Back

Jeder Schreibzugriff aktualisiert nur den Cache. Bei einer Verdrängung des Cacheblocks aus dem Cache wird der Hauptspeicher aktualisiert. Hier kann es zu Inkonsistenzen zwischen dem Cache und dem Hauptspeicher kommen.

Tutorien Übungsblatt - Caches

Frage 9.1.c

Welche Strategien gibt es bei einem Schreibzugriff auf Daten, die nicht im Cache liegen?

Tutorien Übungsblatt - Caches

Frage 9.1.c

Welche Strategien gibt es bei einem Schreibzugriff auf Daten, die nicht im Cache liegen?

Antwort

Write-Allocate
Die zu schreibenden Daten werden im Cache gespeichert. Anschließend wird entweder die Write-Through oder die Write-Back Strategie angewendet.

Write-to-Memory
Die Daten werden direkt in den Hauptspeicher geschrieben, es wird keine Kopie im Cache angelegt.

Tutorien Übungsblatt - Caches

Frage 9.1.d

Welche Probleme können bei virtuell indizierten und virtuell getaggten Caches (VIVT) entstehen?

Tutorien Übungsblatt - Caches

Frage 9.1.d

Welche Probleme können bei virtuell indizierten und virtuell getaggten Caches (VIVT) entstehen?

Antwort

Ambiguity (Mehrdeutigkeit)
Gleiche virtuelle Adressen zeigen auf verschiedene physikalische Adressen

Aliases (Bedeutungsgleichheit)
Verschiedene virtuelle Adressen zeigen auf gleiche physikalische Adressen

Frage 9.1.e

Lösen virtuell indizierte und physikalisch getaggte Caches (VIPT) die vorher genannten Probleme?

Tutorien Übungsblatt - Caches

Frage 9.1.e

Lösen virtuell indizierte und physikalisch getaggte Caches (VIPT) die vorher genannten Probleme?

Antwort

VIPT Caches lösen das Mehrdeutigkeits-, aber nicht das Aliasproblem.

Tutorien Übungsblatt - Paging Basics

Frage 9.2.a

Was versteht man unter den Begriffen „demand paging“ und „pre-paging“?
Was sind die jeweiligen Vor- und Nachteile?

Tutorien Übungsblatt - Paging Basics

Demand Paging (Paging bei Bedarf)

Bei „demand paging“ werden Seiten nur bei Bedarf in den Hauptspeicher eingelagert. Der Bedarf entsteht z.B. bei einem Zugriff eines Programms auf eine ausgelagerte Seite (Pagefault).

- ⊕ Nur benötigte Daten werden in Speicher geladen
- ⊖ Hohe Anzahl an Pagefaults bei Programmstart

Tutorien Übungsblatt - Paging Basics

Demand Paging (Paging bei Bedarf)

Bei „demand paging“ werden Seiten nur bei Bedarf in den Hauptspeicher eingelagert. Der Bedarf entsteht z.B. bei einem Zugriff eines Programms auf eine ausgelagerte Seite (Pagefault).

- ⊕ Nur benötigte Daten werden in Speicher geladen
- ⊖ Hohe Anzahl an Pagefaults bei Programmstart

Pre-Paging

Es werden Seiten spekulativ in den Hauptspeicher geladen.

- ⊕ Anzahl an Pagefaults wird reduziert
- ⊕ Bandbreite von Festplatten wird besser ausgenutzt
- ⊖ Seiten, die nicht verwendet werden, wurden umsonst geladen
- ⊖ Speicherauslastung wird erhöht

Tutorien Übungsblatt - Paging Basics

Frage 9.2.b

Für welche Zwecke wird das Valid-Bit in einem Seitentabelleneintrag verwendet?

Tutorien Übungsblatt - Paging Basics

Frage 9.2.b

Für welche Zwecke wird das Valid-Bit in einem Seitentabelleneintrag verwendet?

Antwort

Das Valid-Bit zeigt an, ob eine Seite im Speicher eingelagert ist oder nicht.

Valid Bit gesetzt
Der Seitentabelleneintrag kann für die Umsetzung der virtuellen Seite in einen physikalischen Rahmen verwendet werden.

Valid Bit nicht gesetzt
Ein Seitenfehler wird beim Zugriff auf die Seite ausgelöst, die Seite wird anschließend wieder in den Hauptspeicher eingelagert.

Bei ausgelagerten Seiten (Nicht gesetztes Valid-Bit) kann das Feld für die physikalische Adresse u.a. für die Speicherung des Auslagerungsorts verwendet werden.

Tutorien Übungsblatt - Paging Basics

Frage 9.2.c

Was ist Copy-on-Write (COW)?

Tutorien Übungsblatt - Paging Basics

Frage 9.2.c

Was ist Copy-on-Write (COW)?

Antwort

Copy-on-Write vermeidet unnötiges Kopieren von Seiten beim Anlegen von Kopien von Bereichen des Adressraums.

Tutorien Übungsblatt - Paging Basics

Frage 9.2.c

Wie wird Copy-on-Write implementiert?

Tutorien Übungsblatt - Paging Basics

Frage 9.2.c

Wie wird Copy-on-Write implementiert?

Beispiel: Kopie des Adressraums eines Prozesses (z.B. bei *fork*)

1. Kopie der Seitentabelle wird erstellt
2. Alle Einträge in beiden Seitentabellen werden als schreibgeschützt und COW markiert

Beide Adressräume verweisen nun auf den gleichen physikalischen Frames.

Verfahren beim schreibenden Zugriff auf eine Seite:

1. Seitenfehler wird wegen als schreibgeschützt markierter Seite ausgelöst
2. Seite an neuen Ort im Hauptspeicher kopieren
3. Physikalische Adresse in einer Seitentabelle auf neuen Ort aktualisieren
4. Seite in beiden Seitentabellen als beschreibbar markieren und COW Markierung entfernen
5. Schreibende Instruktion neu starten

Tutorien Übungsblatt - HW- vs. SW-Walked Page Tables

Frage 9.3.a

Was ist der Unterschied zwischen Seitentabellen, die von der Software oder von der Hardware durchlaufen werden?

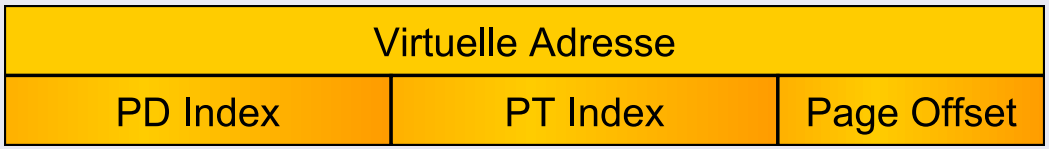
Tutorien Übungsblatt - HW- vs. SW-Walked Page Tables

Frage 9.3.a

Was ist der Unterschied zwischen Seitentabellen, die von der Software oder von der Hardware durchlaufen werden?

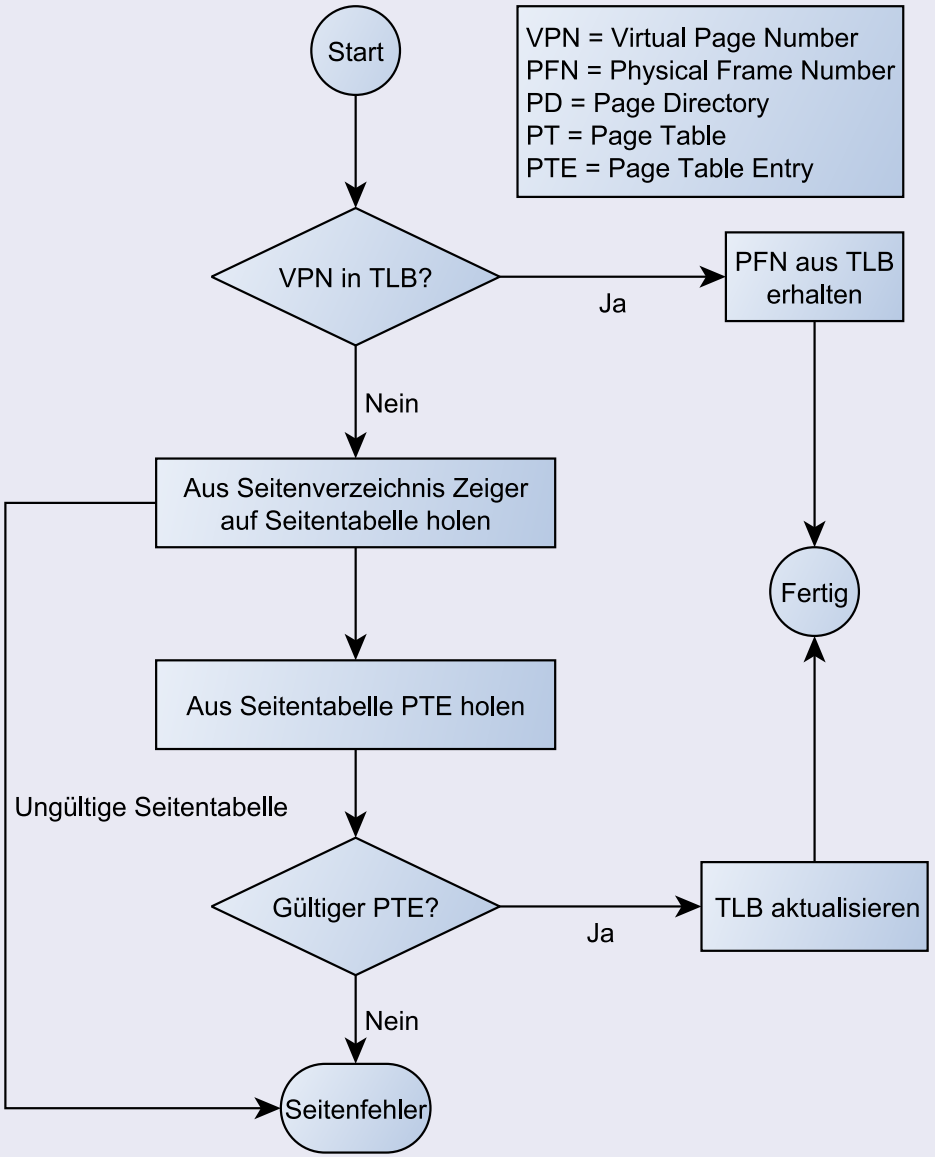
Antwort

Seitentabellen, die von der Hardware durchlaufen werden



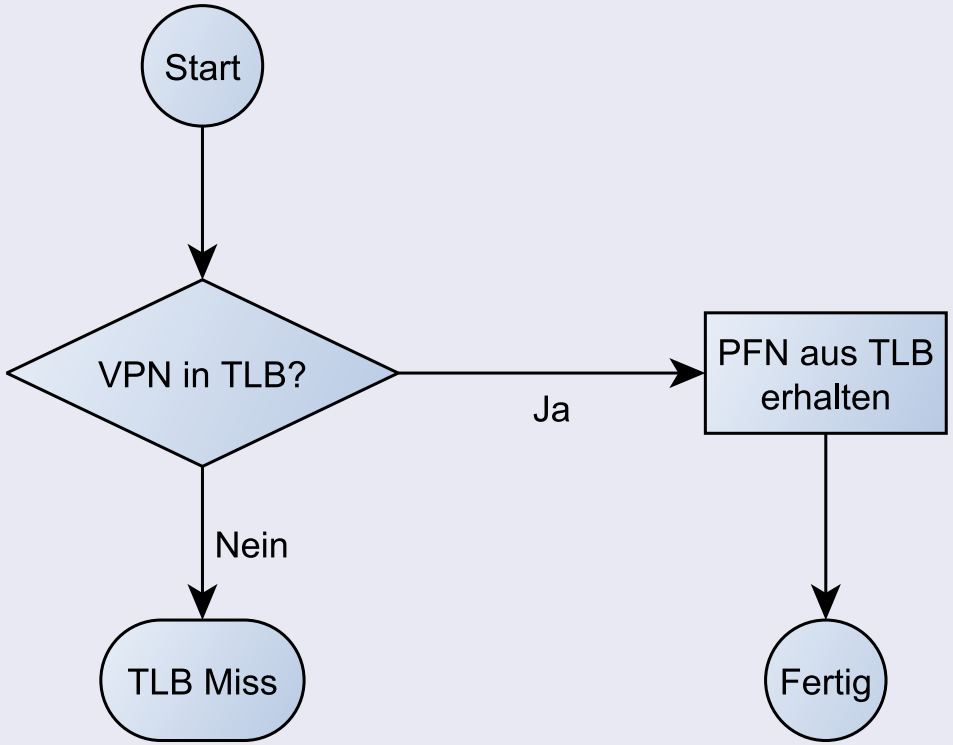
Tutorien Übungsblatt - HW- vs. SW-Walked Page Tables

HW-Walked Page Table



Tutorien Übungsblatt - HW- vs. SW-Walked Page Tables

SW-Walked Page Table



Tutorien Übungsblatt - HW- vs. SW-Walked Page Tables

Frage 9.3.b

Wie unterscheidet sich der Aufbau von Seitentabellen zwischen beiden Verfahren?

Tutorien Übungsblatt - HW- vs. SW-Walked Page Tables

Frage 9.3.b

Wie unterscheidet sich der Aufbau von Seitentabellen zwischen beiden Verfahren?

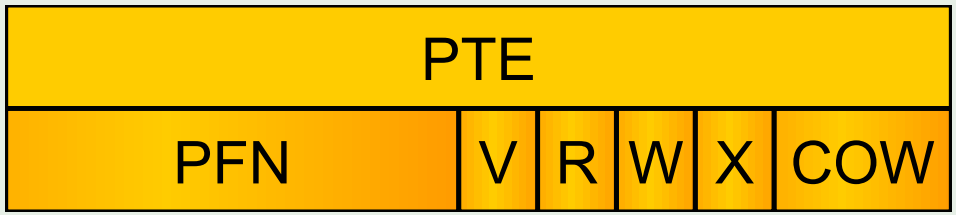
Antwort

Bei beiden Verfahren müssen die physikalische Adresse und weitere Verwaltungsdaten gespeichert werden.

Von Hardware durchlaufene Seitentabellen
Anordnung der Informationen fest vorgegeben

Von Software durchlaufene Seitentabellen
Der Betriebssystemarchitekt kann die Anordnung der Informationen frei variieren und auch einfach weitere Informationen hinzufügen.

Beispiel: PTE



Tutorien Übungsblatt - HW- vs. SW-Walked Page Tables

Frage 9.3.c

Wann werden TLB Miss oder Seitenfehler ausgelöst?

Tutorien Übungsblatt - HW- vs. SW-Walked Page Tables

Frage 9.3.c

Wann werden TLB Miss oder Seitenfehler ausgelöst?

Antwort

TLB Miss (Nur bei SW Seitentabelle)
Bei einem Zugriff auf eine Seite, deren Adressumsetzung nicht im TLB gespeichert ist, wird ein TLB Miss ausgelöst.

Bei einem System mit SW Seitentabellen können auch noch TLB Write, TLB Read oder TLB Execute Ausnahmen ausgelöst werden.

Seitenfehler (Nur bei HW Seitentabelle)
Bei einem Zugriff auf eine Seite, deren Eintrag in der Seitentabelle nicht gültig ist oder deren Zugriffsmodus nicht erlaubt ist.

Tutorien Übungsblatt - HW- vs. SW-Walked Page Tables

Frage 9.3.d

Wie können mehrstufige HW Seitentabellen auf einem komplett mit virtuellen Adressen arbeitenden System benutzt werden?

Tutorien Übungsblatt - HW- vs. SW-Walked Page Tables

Frage 9.3.d

Wie können mehrstufige HW Seitentabellen auf einem komplett mit virtuellen Adressen arbeitenden System benutzt werden?

Antwort

Das grundlegende Problem ist, dass alle Ebenen bis auf die letzte Ebene physikalische Adressen als Zeiger auf die tieferen Ebenen verwenden.

Eine mögliche Lösung ist das Abbilden eines zusammenhängenden physikalischen Speicherbereichs mit einem festen Offset in den Kern-Adressraum.

Tutorien Übungsblatt - Page Replacement Strategies

Frage 9.4.a

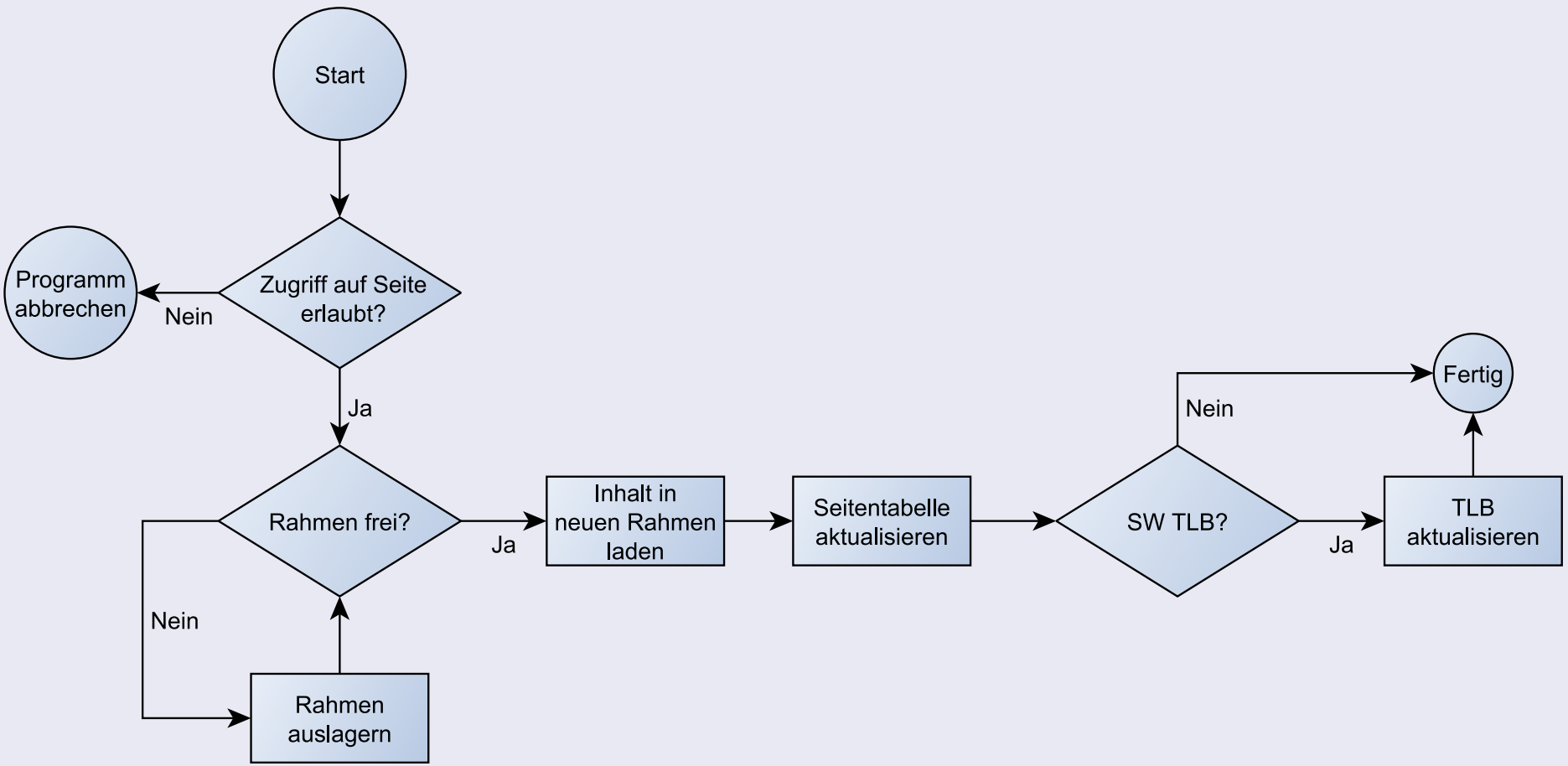
Was passiert bei einem Seitenfehler im Adressraum einer Anwendung?

Tutorien Übungsblatt - Page Replacement Strategies

Frage 9.4.a

Was passiert bei einem Seitenfehler im Adressraum einer Anwendung?

Antwort



Tutorien Übungsblatt - Page Replacement Strategies

Frage 9.4.b

Manche Betriebssysteme versuchen immer eine kleine Anzahl an freien Rahmen bereitzustellen. Welche Vorteile bietet dies?

Tutorien Übungsblatt - Page Replacement Strategies

Frage 9.4.b

Manche Betriebssysteme versuchen immer eine kleine Anzahl an freien Rahmen bereitzustellen. Welche Vorteile bietet dies?

Antwort

Wenn neue Rahmen angefordert werden kann sofort ein freier Rahmen verwendet werden. Falls keine Rahmen ohne Modifikationen verfügbar wären müsste sonst erst ein Rahmen auf die Festplatte zurückgeschrieben werden.

Tutorien Übungsblatt - Page Replacement Strategies

Frage 9.4.c

Was ist der Unterschied zwischen globalen und lokalen Seitenersetzungsalgorithmen? Welche Vor- und Nachteile bestehen?

Tutorien Übungsblatt - Page Replacement Strategies

Frage 9.4.c

Was ist der Unterschied zwischen globalen und lokalen Seitenersetzungsalgorithmen? Welche Vor- und Nachteile bestehen?

Antwort

Globaler Seitenersetzungsalgorithmus

- Verwendet alle Seiten in allen Adressräumen
- Beim Anfordern einer neuen Seite kann ein Rahmen von einer anderen Anwendung verwendet werden

Lokaler Seitenersetzungsalgorithmus

- Verwendet nur Seiten aus dem Adressraum der Anwendung
- Jede Anwendung hat feste Anzahl an Frames zur Verfügung

Tutorien Übungsblatt - Page Replacement Strategies

Vor- und Nachteile eines lokalen Seitenersetzungsalgorithmus

- ⊕ Anwendung hat garantierte Anzahl an Frames zur Verfügung
- ⊕ Kleine Anzahl an Frames erlaubt schnelle Seitenersetzungsalgorithmen
- ⊖ Schwierige Anpassung an wechselnden Speicherbedarf der Anwendung
- ⊖ Wahl der optimalen Anzahl an Frames schwierig

Tutorien Übungsblatt - Page Replacement Strategies

Frage 9.4.d

Was ist Seitenflattern (Thrashing)?

Tutorien Übungsblatt - Page Replacement Strategies

Frage 9.4.d

Was ist Seitenflattern (Thrashing)?

Antwort

Thrashing bezeichnet einen Zustand sehr hoher Paging-Aktivität. Das System ist zum größten Teil der Zeit nur noch damit beschäftigt Seiten ein- und auszulagern.

Thrashing kann auftreten, wenn der Hauptspeicher nicht mehr für die Ausführung aller Anwendungen ausreicht.

Beispiel: Trashing

Eine Anwendung besteht aus zwei Seiten, einer Code- und einer Stack-Seite. Der Anwendung wird ein Rahmen zugewiesen und ein lokaler Seitenersetzungsalgorithmus eingesetzt. Bei jedem Zugriff auf den Stack muss nun die Code-Seite auf die Festplatte ausgelagert und die Stack-Seite von der Festplatte eingelagert werden. Nach dem Stackzugriff muss anschließend die Stack-Seite aus- und die Code-Seite eingelagert werden.

Tutorien Übungsblatt - Page Replacement Strategies

Frage 9.4.e

Was ist die Arbeitsmenge (Working Set) einer Anwendung?
Wie kann damit Trashing verhindern werden?

Tutorien Übungsblatt - Page Replacement Strategies

Frage 9.4.e

Was ist die Arbeitsmenge (Working Set) einer Anwendung?
Wie kann damit Trashing verhindern werden?

Antwort

Die Arbeitsmenge ist die Menge an Seiten, die von den letzten t Seitenreferenzen verwendet wurden. Aufgrund der örtlichen und zeitlichen Lokalität von Anwendungen kann davon ausgegangen werden, dass diese Seiten von der Anwendung häufig benötigt werden.

Das Betriebssystem kann nun sicherstellen, dass die Summe der Arbeitsmengen aller Anwendungen im System kleiner oder gleich der Größe des Hauptspeichers ist. Bei Bedarf kann das Betriebssystem eine komplette Anwendung pausieren und auf die Festplatte auslagern.

Tut. Übungsblatt - Comparison of Replacement Policies

Eine Anwendung hat 4 Rahmen zur Verfügung. Der Zustand der Seitentabelle ist in der folgenden Tabelle dargestellt:

Rahmen	Seite	Wann geladen?	Letzter Zugriff	Referenziert	Verändert
0	2	60	161	0	1
1	1	130	160	0	0
2	0	26	162	1	0
3	3	20	163	1	1

Frage 9.5

Es wird ein Zugriff auf die Seite 4 durchgeführt, dabei wird ein Seitenfehler ausgelöst. Welcher Rahmen wird ausgetauscht?

- First In First Out (FIFO)
- Least recently used (LRU)
- Clock (Aufsteigend nach Ladezeit geordnet, nächster Rahmen Zeiger = 3)
- Optimaler Algorithmus (Folgende Zugriffe: 4,0,0,0,2,4,2,1,0,3,2)

Tut. Übungsblatt - Comparison of Replacement Policies

Ersetzungsstrategie: FIFO

Rahmen 3 wird ausgetauscht, da er am längsten im Speicher war

Tut. Übungsblatt - Comparison of Replacement Policies

Ersetzungsstrategie: FIFO

Rahmen 3 wird ausgetauscht, da er am längsten im Speicher war

Ersetzungsstrategie: LRU

Rahmen 1 wird ausgetauscht, da auf ihn am längsten nicht zugegriffen wurde

Tut. Übungsblatt - Comparison of Replacement Policies

Ersetzungsstrategie: FIFO

Rahmen 3 wird ausgetauscht, da er am längsten im Speicher war

Ersetzungsstrategie: LRU

Rahmen 1 wird ausgetauscht, da auf ihn am längsten nicht zugegriffen wurde

Ersetzungsstrategie: Clock

- R Bit bei Rahmen 3 entfernen
- R Bit bei Rahmen 2 entfernen
- Rahmen 0 wählen, da $R = 0$

Tut. Übungsblatt - Comparison of Replacement Policies

Ersetzungsstrategie: FIFO

Rahmen 3 wird ausgetauscht, da er am längsten im Speicher war

Ersetzungsstrategie: LRU

Rahmen 1 wird ausgetauscht, da auf ihn am längsten nicht zugegriffen wurde

Ersetzungsstrategie: Clock

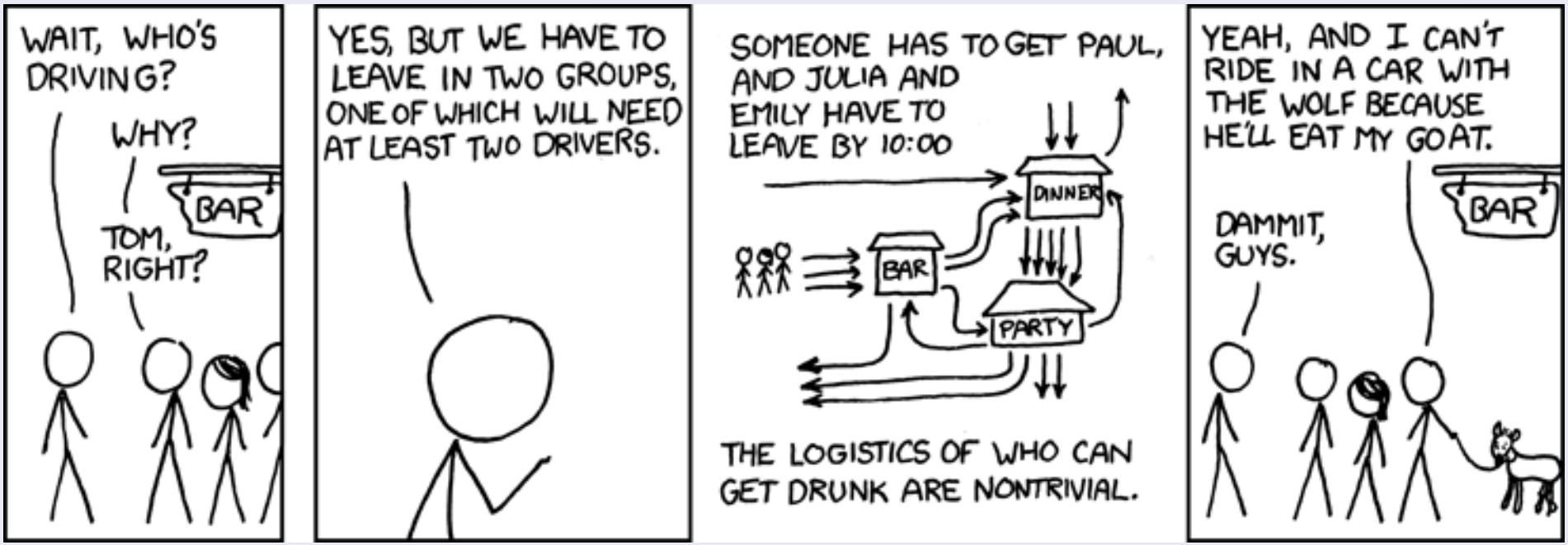
- R Bit bei Rahmen 3 entfernen
- R Bit bei Rahmen 2 entfernen
- Rahmen 0 wählen, da $R = 0$

Ersetzungsstrategie: Optimaler Algorithmus

Rahmen 3 wird ausgetauscht, da auf ihn am längsten nicht zugegriffen wird

Fragen & Kommentare?

Ende



xkcd: Designated Drivers