

Script generated by TTT

Title: Baumgarten: GBS (29.11.2013)

Date: Fri Nov 29 08:29:53 CET 2013

Duration: 86:04 min

Pages: 46

Grundlagen: Betriebssysteme und Systemsoftware (GBS)

Uwe Baumgarten

gbs_course-student.pdf
TUM-GBS-Fol-2013-ALLE-1-161.pdf — TUM-GBS-Fol-2013.ppt [Kompatibilitätsmodus]
Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe
Vorherige Nächste 1 (1 von 161) Auf Seitenbreite einpassen
Vorschaubilder
Technische Universität München
Grundlagen:
Betriebssysteme und Systemsoftware
(GBS)
Uwe Baumgarten
Vorlesungen
gruppen.pdf
Fr, 29. Nov, 08:31

gbs_course-student.pdf
TUM-GBS-Fol-2013-ALLE-1-161.pdf — TUM-GBS-Fol-2013.ppt [Kompatibilitätsmodus]
Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe
Vorherige Nächste 37 (37 von 161) Auf Seitenbreite einpassen
Vorschaubilder
Technische Universität München
Inhalte
1. Übersicht [38]
2. Einführung [45]
3. Parallele Systeme – Modellierung, Strukturen [71]
4. Prozess- und Prozessorverwaltung [115]
5. Speicherverwaltung [131]
6. Prozesskommunikation [153]
7. Dateisysteme []
8. Ein/Ausgabe []
9. Sicherheit in Rechensystemen []
10. Entwurf von Betriebssystemen []
11. Zusammenfassung []
Vorlesungen
gruppen.pdf
Fr, 29. Nov, 08:32

gbs_course-student.pdf

TUM-GBS-Fol-2013-ALLE-1-161.pdf — TUM-GBS-Fol-2013.ppt [Kompatibilitätsmodus]

Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe

Vorherige Nächste 131 (131 von 161) Auf Seitenbreite einpassen

Vorschau

Technische Universität München TUM

5. Speicherverwaltung

- Fragestellungen
 - Verwaltungsaufgabe
 - Programadressraum und Maschinenadressraum
 - Virtueller Speicher
 - Adressierungsaufgabe
 - Direkte Adressierung, Basisadressierung, virtuelle Adressen
- Einführung
- Speicherabbildungen
- **Seitenadressierung** [140]
- Segment-Seitenadressierung
- Speicherhierarchie – Caches

Quelle: [JS12] Kap. 5

Vorlesungen gruppen.pdf

Menu TUM-GBS-Fol... gbs_course-s... Fr, 29. Nov, 08:33

Seitenadressierung – Seiten-Kacheltabelle

- **Seitendeskriptor**
 - Zugriffsrechte (rwx, lesend – schreibend – ausführend)
 - Existent-Attribut (e)
 - Geladen-Attribut (v)
 - Zugriffen-Attribut (r) | zeit
 - Verändert-Attribut (m)
 - Seitenadresse (s als Index)
 - Kacheladresse (ggf.) (Kachel k)
 - Hintergrundspeicheradresse (ggf.) (Block b)
- Große Seiten-Kacheltabellen
 - Mehrstufige Tabellen [JS12, Kap. 5, p. 131]
- Realisierungsvarianten
 - Prozess-spezifisch oder global, indiziert oder assoziativ

5. Speicherverwaltung

Quelle: [JS12] Kap. 5

Seitenverwaltungsstrategien (II)

- Seitenverdrängungsstrategie
 - LIFO (last in first out)
 - LRU (least recently used)
 - Optimale Strategie
 - Working-Set Model
 - Menge der Seiten der letzten k Zugriffe (k Fenstergröße)
 - Garantie: Mindestens der Working-Set eines aktiven Prozesses liegt im AS
 - Konsequenzen für die Verdrängung
- Ergänzungen
 - Wahl der Seitengröße
 - Overhead vs. Fragmentierung
 - Seitenflattern (thrashing)

5. Speicherverwaltung

Quelle: [JS12] Kap. 5

Virtuelle Adressierung in Linux

- Mehrstufige Seitentabellen
 - Page directory
 - Page middle directory
 - Page table
 - [JS12, Kap. 5, p. 137]
- Seitengruppierung
 - Realisierung in aufeinanderfolgenden Seitenrahmen
 - Bis zu 32 Seiten
- Seitenverdrängung
 - Modifizierter Clock-Algorithmus mit 8-bit-Zähler

5. Speicherverwaltung

Quelle: [JS12] Kap. 5

gbs_course-student.pdf

TUM-GBS-Fol-2013-ALLE-1-161.pdf — TUM-GBS-Fol-2013.ppt [Kompatibilitätsmodus]

Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe

Vorherige Nächste 149 (149 von 161) Auf Seitenbreite einpassen

Vorschaubilder

Technische Universität München

Virtuelle Adressierung in Linux

- Mehrstufige Seitentabellen
 - Page directory
 - Page middle directory
 - Page table
 - [JS12, Kap. 5, p. 137]
- Seitengruppierung
 - Realisierung in aufeinanderfolgenden Seitenrahmen
 - Bis zu 32 Seiten
- Seitenverdrängung
 - Modifizierter Clock-Algorithmus mit 8-bit-Zähler

5. Speicherverwaltung

Quelle: [JS12] Kap. 5

Vorlesungen
gruppen.pdf

Menu TUM-GBS-Fol... gbs_course-s... Fr, 29. Nov, 08:41

gbs_course-student.pdf

Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe

Vorherige Nächste 137 (144 von 228) 150%

Vorschaubilder

- **Seitenallokation**
 - aufeinanderfolgende Seiten werden auf zusammenhängende Kacheln abgebildet.
 - Behandlung von Gruppen mit 1, 2, 4, 8, 16, oder 32 Seiten.
- **Seitenverdrängung**
 - Anwendung eines modifizierten Clock-Algorithmus. Zugriffsbit r wird durch einen 8-bit Zähler ersetzt.
 - bei jedem Zugriff wird der Zähler inkrementiert.
 - Linux dekrementiert periodisch die Zähler aller Seiten im Arbeitsspeicher.

Vorlesungen
gruppen.pdf

Menu TUM-GBS-Fol... gbs_course-s... Fr, 29. Nov, 08:45

Virtuelle Adressierung in Linux

- Mehrstufige Seitentabellen
 - Page directory
 - Page middle directory
 - Page table
 - [JS12, Kap. 5, p. 137]
- Seitengruppierung
 - Realisierung in aufeinanderfolgenden Seitenrahmen
 - Bis zu 32 Seiten
- Seitenverdrängung
 - Modifizierter Clock-Algorithmus mit 8-bit-Zähler

5. Speicherverwaltung

Quelle: [JS12] Kap. 5

Segment-Seitenadressierung

- Prozessadressraum unterteilt in Segmente
 - Code-Segment, Daten-Segment, Stack-Segment
 - Segmenttabelle pro Prozess
 - Segmentzugriffsfehler
 - Angepasste Adressierung
 - Ggf. Hardware-Unterstützung
- [JS12, Kap. 5, p. 138]

5. Speicherverwaltung

Quelle: [JS12] Kap. 5

Segment-Seitenadressierung

- Prozessadressraum unterteilt in Segmente
 - Code-Segment, Daten-Segment, Stack-Segment
 - Segmenttabelle pro Prozess
 - Segmentzugriffsfehler
 - Angepasste Adressierung
 - Ggf. Hardware-Unterstützung

5. Speicherverwaltung • [JS12, Kap. 5, p. 138]

Quelle: [JS12] Kap. 5

Segment-Seitenadressierung

- Prozessadressraum unterteilt in Segmente
 - Code-Segment, Daten-Segment, Stack-Segment
 - Segmenttabelle pro Prozess
 - Segmentzugriffsfehler
 - Angepasste Adressierung
 - Ggf. Hardware-Unterstützung

• [JS12, Kap. 5, p. 138]

Kap. 5

Anwendung eines modifizierten Clock-Algorithmus. Zugriffsbit r wird durch einen 8-bit Zähler ersetzt.

bei jedem Zugriff wird der Zähler inkrementiert.
Linux dekrementiert periodisch die Zähler aller Seiten im Arbeitsspeicher.



147



148



149



150

Technische Universität München

Segment-Seitenadressierung

- Prozessadressraum unterteilt in Segmente
 - Code-Segment, Daten-Segment, Stack-Segment
 - Segmenttabelle pro Prozess
 - Segmentzugriffsfehler
 - Angepasste Adressierung
 - Ggf. Hardware-Unterstützung

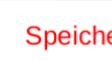
5. Speicherverwaltung • [JS12, Kap. 5, p. 138]

Quelle: [JS12] Kap. 5

© UB TUM GBS WS 2013/14 Grundlagen:



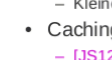
151



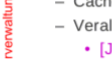
152



153



154



155

Technische Universität München

Speicherhierarchie / Caches

- Beispiel
 - [JS12, Kap. 5, p. 139]
- Cache-Speicher
 - Kleine, schnell zugreifbare Speichereinheit
- Caching in der Speicherhierarchie
 - [JS12, Kap. 5, p. 140]
 - Cache-Treffen (hit) und Cache-Fehler (miss)
 - Verallgemeinerte Caches
 - [JS12, Kap. 5, p. 141]

5. Speicherverwaltung

Quelle: [JS12] Kap. 5

© UB TUM GBS WS 2013/14 Grundlagen:

Speicherhierarchie / Caches

- Beispiel
 - [JS12, Kap. 5, p. 139]
- Cache-Speicher
 - Kleine, schnell zugreifbare Speichereinheit
- Caching in der Speicherhierarchie
 - [JS12, Kap. 5, p. 140]
 - Cache-Treffen (hit) und Cache-Fehler (miss)
 - Verallgemeinerte Caches
 - [JS12, Kap. 5, p. 141]

5. Speicherverwaltung

Quelle: [JS12] Kap. 5

Realisierung von Caches

- Allgemeine Cache Organisation
 - Mengen von Cache-Reihen (Lines)
 - Cache-Reihe mit fester Byteanzahl
 - [JS12, Kap. 5, p. 142]
- Adressierung von Caches (Menge, Reihe, Wort)
 - [JS12, Kap. 5, p. 143]
 - Direct Mapped Cache
 - [JS12, Kap. 5, p. 143]
 - Mengen-assoziativer Cache
 - [JS12, Kap. 5, p. 144]
- Cache-freundlicher Code
 - Lokalitätsverhalten (gemessen in Cache-Fehlerrate)

5. Speicherverwaltung

Quelle: [JS12] Kap. 5

gbs_course-student.pdf

140 (147 von 228) 150%

5.6.3 Caching in der Speicherhierarchie

kleinere, schnellere, teurere Einheiten auf Schicht k cachen Daten von größeren, langsameren, preiswerteren Einheiten der Schicht k + 1.

Programme braucht Objekt d in Block b

- Cache-Treffer (hit): Programm findet b im Cache der Schicht k, z.B. in Block 10
- Cache-Fehler (miss): b ist nicht auf Schicht k; Cache muss b von Schicht k+1 holen (z.B. Block 8).

Vorlesungen gruppen.pdf

Fr. 29. Nov. 09:00

gbs_course-student.pdf

TUM-GBS-Fol-2013-ALLE-1-161.pdf -- TUM-GBS-Fol-2013.ppt [Kompatibilitätsmodus]

151 (151 von 161) Auf Seitenbreite einpassen

Speicherhierarchie / Caches

- Beispiel
 - [JS12, Kap. 5, p. 139]
- Cache-Speicher
 - Kleine, schnell zugreifbare Speichereinheit
- Caching in der Speicherhierarchie
 - [JS12, Kap. 5, p. 140]
 - Cache-Treffen (hit) und Cache-Fehler (miss)
 - Verallgemeinerte Caches
 - [JS12, Kap. 5, p. 141]

Caches werden auf unterschiedlichen Ebenen in modernen Rechnersystemen genutzt.

Cache Typ	was wird gecached	wo wird gecached	kontrolliert
Register	4 Byte Wort	CPU Register	Compiler
TI R	Address Transla..	On-Chip TI R	Hardware

Vorlesungen gruppen.pdf

Fr. 29. Nov. 09:01

Realisierung von Caches

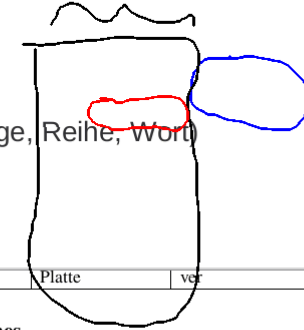
- Allgemeine Cache Organisation
 - Mengen von Cache-Reihen (Lines)
 - Cache-Reihe mit fester Byteanzahl
 - [JS12, Kap. 5, p. 142]
- Adressierung von Caches (Menge, Reihe, Wort)
 - Direct Mapped Cache
 - [JS12, Kap. 5, p. 143]
 - Mengen-assoziativer Cache
 - [JS12, Kap. 5, p. 144]
- Cache-freundlicher Code
 - Lokalitätsverhalten (gemessen in Cache-Fehlerrate)

5. Speicherverwaltung

Quelle: [JS12] Kap. 5

Realisierung von Caches

- Allgemeine Cache Organisation
 - Mengen von Cache-Reihen (Lines)
 - Cache-Reihe mit fester Byteanzahl
 - [JS12, Kap. 5, p. 142]
- Adressierung von Caches (Menge, Reihe, Wort)
 - Direct Mapped Cache
 - [JS12, Kap. 5, p. 143]
 - Mengen-assoziativer Cache



Speicherverwaltung

Kap. 5

5.6.4 Realisierung von Caches

Obwohl je nach der Ebene der Cache Hierarchie unterschiedliche Varianten der Umsetzung existieren, folgen diese doch gemeinsamen allgemeinen Prinzipien.

gbs_course-student.pdf

142 (149 von 228) 150%

Schlichter, TU München 5.6. SPEICHERHIERARCHIE / CACHES

Cache ist ein Array von Mengen ($S = \text{Anzahl von Mengen}$),
 jede Menge enthält eine oder mehrere Reihen ($E = \text{Anzahl von Reihen pro Menge}$),
 jede Reihe enthält einen Datenblock (B ist Blockgröße pro Reihe).
 Cachegröße = $B * E * S$ Bytes.

1-gültig Bit pro Reihe, t tagbits pro Reihe, $B = 2^b$ Bytes pro Cache Block

set 0	valid	tag	0	1	---	B-1
	valid	tag	0	1	---	B-1
set 1	valid	tag	0	1	---	B-1
	valid	tag	0	1	---	B-1
set s-1	valid	tag	0	1	---	B-1
	valid	tag	0	1	---	B-1

S Mengen, E Reihen pro Menge

- Adressierung von Caches

Die Cache Organisation bedingt eine Strukturierung der Speicheradresse

gbs_course-student.pdf

143 (150 von 228) 150%

Adresse A

tag	set Index	block offset
t bits	s bits	b bits

set 0	valid	tag	0	1	---	B-1
	valid	tag	0	1	---	B-1
set 1	valid	tag	0	1	---	B-1
	valid	tag	0	1	---	B-1
set s-1	valid	tag	0	1	---	B-1
	valid	tag	0	1	---	B-1

- Direct-Mapped Cache

Einfachste Art des Caches mit genau einer Reihe pro Menge; Zugriff erfolgt in der Reihenfolge

Mengenselektion: Benutzung der "set Index" bits zum Bestimmen der relevanten Menge.

Reihenabgleich: Finden der gültigen Reihe in der selektierten Menge mit dem richtigen Tag.

gbs_course-student.pdf

Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe

Vorherige Nächste 143 (150 von 228) 150%

Vorschaubilder

149

150

151

152

Vorlesungen

gruppen.pdf

• **Direct-Mapped Cache**

Einfachste Art des Caches mit genau einer Reihe pro Menge; Zugriff erfolgt in der Reihenfolge

Mengenselektion: Benutzung der "set Index" bits zum Bestimmen der relevanten Menge.

Reihenabgleich: Finden der gültigen Reihe in der selektierten Menge mit dem richtigen Tag.

Wortselektion: Benutzung des block offsets zum Bestimmen des relevanten Wortes.

• **Mengen Assoziativer Cache**

143

© UB TUM GBS WS 2013/14 Grundlagen: Betriebssysteme und Systemsoftware (IN0009) Fr, 29. Nov, 09:11

gbs_course-student.pdf

TUM-GBS-Fol-2013-ALLE-1-161.pdf — TUM-GBS-Fol-2013.ppt [Kompatibilitätsmodus]

Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe

Vorherige Nächste 152 (152 von 161) Auf Seitenbreite einpassen

Vorschaubilder

149

150

151

152

Vorlesungen

gruppen.pdf

Technische Universität München

TUM

Realisierung von Caches

- Allgemeine Cache Organisation
 - Mengen von Cache-Reihen (Lines)
 - Cache-Reihe mit fester Byteanzahl
 - [JS12, Kap. 5, p. 142]
- Adressierung von Caches (Menge, Reihe, Wort)
 - [JS12, Kap. 5, p. 143]
 - Direct Mapped Cache
 - [JS12, Kap. 5, p. 142]
 - Mengen-assoziativer Cache
 - [JS12, Kap. 5, p. 144]
- Cache-freundlicher Code
 - Lokalitätsverhalten (gemessen in Cache-Fehlerrate)

5. Speicherverwaltung

Quelle: [JS12] Kap. 5

© UB TUM GBS WS 2013/14 Grundlagen:

• **5.6.5 Cache freundlicher Code**

Die Performanz eines Caches wird anhand der auftretenden Cache Misses und Cache Hits gemessen.

© UB TUM GBS WS 2013/14 Grundlagen: Betriebssysteme und Systemsoftware (IN0009) Fr, 29. Nov, 09:14

6. Prozesskommunikation

- Fragestellungen
 - Kommunikationsarten
 - Nachrichten-basierte Kommunikation, insbs. Client-Server
 - Sockets und Ports
- Einführung
 - Ziel: lokal und entfernt (verteiltes System) in ähnlicher Art
- Nachrichten-basierte Kommunikation
- Client-Server-Modell Sockets **UNIX**
- Netzwerkprogrammierung

Quelle: [JS12] Kap. 6

Einführung

- Kommunikationsarten
 - Schmalbandige Kommunikation
 - Ereignisse, Flags, Unterbrechungen
 - Breitbandige Kommunikation
 - Implizit oder indirekt/mittelbar
 - Nutzung gemeinsamer Betriebsmittel (Dateien, Speicher, ...)
 - Explizit oder direkt/unmittelbar
 - SEND und RECEIVE als nachrichtenbasierte Kommunikation
- Kommunikationsbeziehungen
 - 1:1 unicast, anycast
 - 1:m multicast
 - 1:alle broadcast
 - n:1

6. Prozesskommunikation

Quelle: [JS12] Kap. 6

Einführung

- Implizite Kommunikationsformen
 - 1:1 ein Puffer für Sender/Empfänger-Paar
 - n:1 n Sender senden an eine Empfänger
 - 1:m Mitteilung an eine Menge von Prozessen (Multicast)
 - 1:alle Mitteilung an alle Prozesse (Broadcast)
 - n:m ein Puffer für n Erzeuger und m Verbraucher
- [JS12, Kap. 5, p. 149]
- Explizite Kommunikationsformen
 - Nachrichtenbasierte Kommunikation (message passing)
 - SEND und RECEIVE
 - Nachrichtenaufbau
 - Nachrichtenkopf
 - Nachrichteninhalt = Nutzlast (payload)

6. Prozesskommunikation

Quelle: [JS12] Kap. 6

gbs_course-student.pdf

TUM-GBS-Fol-2013-ALLE-1-161.pdf — TUM-GBS-Fol-2013.ppt [Kompatibilitätsmodus]

Technische Universität München

Einführung

- Implizite Kommunikationsformen
 - 1:1 ein Puffer für Sender/Empfänger-Paar
 - n:1 n Sender senden an eine Empfänger
 - 1:m Mitteilung an eine Menge von Prozessen (Multicast)
 - 1:alle Mitteilung an alle Prozesse (Broadcast)
 - n:m ein Puffer für n Erzeuger und m Verbraucher
- [JS12, Kap. 5, p. 149]
- Explizite Kommunikationsformen
 - Nachrichtenbasierte Kommunikation (message passing)
 - SEND und RECEIVE

5.6.5 Cache freundlicher Code

Die Performanz eines Caches wird anhand der auftretenden Cache Misses und Cache Hits gemessen.

Fr. 29. Nov. 09:32

Kommunikation

Kap. 6

gbs_course-student.pdf

TUM-GBS-Fol-2013-ALLE-1-161.pdf — TUM-GBS-Fol-2013.ppt [Kompatibilitätsmodus]

Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe

Vorherige Nächste 155 (155 von 161) Auf Seitenbreite einpassen

Vorschaubilder

152

153

154

155

Technische Universität München

Einführung

- Implizite Kommunikationsformen
 - 1:1 ein Puffer für Sender/Empfänger-Paar
 - n:1 n Sender senden an eine Empfänger
 - 1:m Mitteilung an eine Menge von Prozessen (Multicast)
 - 1:alle Mitteilung an alle Prozesse (Broadcast)
 - n:m ein Puffer für n Erzeuger und m Verbraucher
- [JS12, Kap. 5, p. 149]
- Explizite Kommunikationsformen
 - Nachrichtenbasierte Kommunikation (message passing)
 - SEND und RECEIVE
 - Nachrichtenaufbau
 - Nachrichtenkopf
 - Nachrichteninhalt = Nutzlast (payload)

6. Prozesskommunikation

Quelle: [JS12] Kap. 6

© UB TUM GBS WS 2013/14 Grundlagen:

E1 E1 E1 Em

1:1 Kommunikation n:1 Kommunikation n:m Kommunikation

Si i-ter Senderprozess
Ei i-ter Empfängerprozess

Fr. 29. Nov. 09:32

Einführung

- Übergang zu Verteilten Systemen
- Multi-CPU-Systeme
 - Multiprozessoren/Multi-Core-Systeme enggekoppelt
 - Multicomputer
 - Verteiltes System lose gekoppelt
- Klassifikation
 - [JS12, Kap. 5, p. 151]

6. Prozesskommunikation

Quelle: [JS12] Kap. 6

gbs_course-student.pdf
TUM-GBS-Fol-2013-ALLE-1-161.pdf — TUM-GBS-Fol-2013.ppt [Kompatibilitätsmodus]

Technische Universität München

Einführung

- Übergang zu Verteilten Systemen
- Multi-CPU-Systeme
 - Multiprozessoren/Multi-Core-Systeme
 - Multicomputer
 - Verteiltes System
- Klassifikation
 - [JS12, Kap. 5, p. 151]

enggekoppelt
lose gekoppelt

Empfängerfunktion
Nachrichteninhalt: Nutzlast (payload)

- explizite Kommunikation ist besonders geeignet in verteilten, vernetzten Systemen.

6.2.2 Verteilte Systeme

Kommunikation

Vorlesungen
gruppen.pdf

Fr, 29. Nov, 09:38

Technische Universität München

Nachrichten-basierte Kommunikation

- „send(E: process, m: message)“ „receive(S: process, m: message)“
- Meldung i.d.R. unidirektionaler Nachrichtenaustausch
 - asynchron
 - [JS12, Kap. 6, p. 153]
 - synchron (z.B. Rendezvous)
 - [JS12, Kap. 6, p. 153]
- Auftrag i.d.R. bidirektionaler Nachrichtenaustausch
 - synchron
 - [JS12, Kap. 6, p. 154]
 - asynchron
 - [JS12, Kap. 6, p. 155]

6. Prozesskommunikation

Quelle: [JS12] Kap. 6

© UB TUM GBS WS 2013/14 Grundlagen:
Betriebssysteme und Systemsoftware (IN0009)

157

gbs_course-student.pdf
TUM-GBS-Fol-2013-ALLE-1-161.pdf — TUM-GBS-Fol-2013.ppt [Kompatibilitätsmodus]

Technische Universität München

Nachrichten-basierte Kommunikation

- „send(E: process, m: message)“ „receive(S: process, m: message)“
- Meldung i.d.R. unidirektionaler Nachrichtenaustausch
 - asynchron
 - [JS12, Kap. 6, p. 153]
 - synchron (z.B. Rendezvous)
 - [JS12, Kap. 6, p. 153]
- Auftrag i.d.R. bidirektionaler Nachrichtenaustausch
 - synchron
 - [JS12, Kap. 6, p. 154]
 - asynchron
 - [JS12, Kap. 6, p. 155]

Verwaltung eine Organisati- eine Organisati- viele Organisati-
on on onen

Die Einteilung in Multicomputer oder verteiltes System hängt etwas von der verwendeten Betrachtungsweise ab.

Kommunikation

Vorlesungen
gruppen.pdf

Fr, 29. Nov, 09:43

gbs_course-student.pdf

Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe

Vorherige Nächste 153 (160 von 228) 150%

Vorschaubilder

154
155
156
157

Sender S Nachrichtendienst ND Empfänger E

send → Meldung → receive

Zeit

- Nachrichtendienst des Betriebssystems puffert Nachricht; Sender S kann seine Ausführung fortsetzen, sobald Nachricht N in den Nachrichtenpuffer des ND eingetragen ist. S wartet *nicht*, bis E die Nachricht empfangen hat.
- Empfänger E zeigt durch receive an, dass er am Empfang der Nachricht N interessiert ist. Empfänger wird blockiert, bis Sender Nachricht bereit stellt.

- Synchrone Meldung**
Sender und Empfänger von Meldungen sind zeitlich gekoppelt.

Vorlesungen
gruppen.pdf

Fr, 29. Nov, 09:45

gbs_course-student.pdf

153 (160 von 228) 150%

Zeit

- Nachrichtendienst des Betriebssystems puffert Nachricht; Sender S kann seine Ausführung fortsetzen, sobald Nachricht N in den Nachrichtenpuffer des ND eingetragen ist. S wartet *nicht*, bis E die Nachricht empfangen hat.
- Empfänger E zeigt durch receive an, dass er am Empfang der Nachricht N interessiert ist. Empfänger wird blockiert, bis Sender Nachricht bereit stellt.

• **Synchrone Meldung**
 Sender und Empfänger von Meldungen sind zeitlich gekoppelt.

Sender S Nachrichtendienst ND Empfänger E

send Meldung receive

Quittung

154

155

156

157

Vorlesungen

gruppen.pdf

Fr, 29. Nov, 09:46

gbs_course-student.pdf

153 (160 von 228) 150%

Zeit

- Nachrichtendienst des Betriebssystems puffert Nachricht; Sender S kann seine Ausführung fortsetzen, sobald Nachricht N in den Nachrichtenpuffer des ND eingetragen ist. S wartet *nicht*, bis E die Nachricht empfangen hat.
- Empfänger E zeigt durch receive an, dass er am Empfang der Nachricht N interessiert ist. Empfänger wird blockiert, bis Sender Nachricht bereit stellt.

• **Synchrone Meldung**
 Sender und Empfänger von Meldungen sind zeitlich gekoppelt.

Sender S Nachrichtendienst ND Empfänger E

send Meldung receive

154

155

156

157

Vorlesungen

gruppen.pdf

Fr, 29. Nov, 09:46

gbs_course-student.pdf

153 (160 von 228) 150%

- Nachrichtendienst des Betriebssystems puffert Nachricht; Sender S kann seine Ausführung fortsetzen, sobald Nachricht N in den Nachrichtenpuffer des ND eingetragen ist. S wartet *nicht*, bis E die Nachricht empfangen hat.
- Empfänger E zeigt durch receive an, dass er am Empfang der Nachricht N interessiert ist. Empfänger wird blockiert, bis Sender Nachricht bereit stellt.

• **Synchrone Meldung**
 Sender und Empfänger von Meldungen sind zeitlich gekoppelt.

Sender S Nachrichtendienst ND Empfänger E

send Meldung receive

Quittung

Zeit

154

155

156

157

Vorlesungen

gruppen.pdf

Fr, 29. Nov, 09:48

gbs_course-student.pdf

155 (162 von 228) 150%

Schlichter, TU München 6.3. NACHRICHTENBASIERTE KOMMUNIKATION

Sender S Nachrichtendienst ND Empfänger E

send Auftrag receive

reply Resultat

Zeit

154

156

157

Vorlesungen

gruppen.pdf

Fr, 29. Nov, 09:49

Vorteile/Nachteile synchrone Sender

gbs_course-student.pdf

TUM-GBS-Fol-2013-ALLE-1-161.pdf — TUM-GBS-Fol-2013.ppt [Kompatibilitätsmodus]

Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe

Vorherige Nächste 157 (157 von 161) Auf Seitenbreite einpassen

Vorschaubilder

154

155

156

157

Technische Universität München TUM

Nachrichten-basierte Kommunikation

- „send(E: process, m: message)“, „receive(S: process, m: message)“
- Meldung i.d.R. unidirektionaler Nachrichtenaustausch
 - asynchron
 - synchron (z.B. Rendezvous)
- Auftrag i.d.R. bidirektionaler Nachrichtenaustausch
 - synchron
 - asynchron

Quelle: [JS12] Kap. 6

6. Prozesskommunikation

Zeit

Vorteile/Nachteile asynchrones Senden

Vorlesungen gruppen.pdf

Menu TUM-GBS-Fol... gbs_course-s... Fr, 29. Nov, 09:50

gbs_course-student.pdf

TUM-GBS-Fol-2013-ALLE-1-161.pdf — TUM-GBS-Fol-2013.ppt [Kompatibilitätsmodus]

Technische Universität München TUM

Nachrichten-basierte Kommunikation

- Vorteile von asynchronem Senden
 - Nützlich für Realzeitanwendungen
 - Ermöglicht parallele Abarbeitung durch Sender und Empfänger
 - Anwendbar zum Signalisieren von Ereignissen
- Nachteile von asynchronem Senden
 - Verwaltung des Nachrichtenpuffers durch BS erforderlich
 - Benachrichtigung des Senders S im Fehlerfall und Behandlung von Fehlern ist problematisch
 - Entwurf und Nachweis der Korrektheit ist schwierig
- Erzeuger-Verbraucher-Problem

Quelle: [JS12] Kap. 6

6. Prozesskommunikation

Zeit

Vorteile/Nachteile asynchrones Senden

Vorlesungen gruppen.pdf

Menu TUM-GBS-Fol... gbs_course-s... Fr, 29. Nov, 09:52

gbs_course-student.pdf

TUM-GBS-Fol-2013-ALLE-1-161.pdf — TUM-GBS-Fol-2013.ppt [Kompatibilitätsmodus]

Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe

Vorherige Nächste 158 (158 von 161) Auf Seitenbreite einpassen

Vorschaubilder

155

156

157

158

Technische Universität München TUM

Nachrichten-basierte Kommunikation

- Vorteile von asynchronem Senden
 - Nützlich für Realzeitanwendungen
 - Ermöglicht parallele Abarbeitung durch Sender und Empfänger
 - Anwendbar zum Signalisieren von Ereignissen
- Nachteile von asynchronem Senden
 - Verwaltung des Nachrichtenpuffers durch BS erforderlich
 - Benachrichtigung des Senders S im Fehlerfall und Behandlung von Fehlern ist problematisch
 - Entwurf und Nachweis der Korrektheit ist schwierig
- Erzeuger-Verbraucher-Problem
 - Programm, Petri-Netz [JS12, Kap. 6, p. 156]
 - Timeouts zur Lösung von unendlichem Warten

Quelle: [JS12] Kap. 6

6. Prozesskommunikation

ack. received

ack. sent

Vorlesungen gruppen.pdf

Menu TUM-GBS-Fol... gbs_course-s... Fr, 29. Nov, 09:53

Technische Universität München TUM

Nachrichten-basierte Kommunikation

- Ports (verbindungslos, Paket-orientiert)
 - Loslösung von Prozessnamen
- Operationen
 - „portID = createPort()“
 - „send(E.portID, message)“
 - „receive(portID, message)“
 - „deletePort(portID)“
- Eigenschaften
 - Port ist mit dem Adressraum des zugehörigen Prozesses verbunden
 - Ein Empfängerprozess kann Sender-spezifische Ports einrichten
- Einsatz
 - TCP-/IP-Kommunikationsprotokolle
 - Vordefinierte Ports [JS12, Kap. 6, p. 158]

Quelle: [JS12] Kap. 6

6. Prozesskommunikation

© UB TUM GBS WS 2013/14 Grundlagen: Betriebssysteme und Systemsoftware (IN0009) 159

Nachrichten-basierte Kommunikation

- Kanäle (verbindungsorientiert)
 - Einrichtung eines Kanals (Socket) zwischen Ports
 - Bidirektionale Übertragung über Kanäle
 - Operationen (SEND, RECEIVE) bezogen auf lokale Ports
 - Übertragung einzelner Pakete
 - Einsatz: TCP mit verbindungsorientierter Kommunikation
- Ströme
 - Übertragung von Datenströmen
 - Verdecken die Nachrichtengrenzen von Kanälen
 - [JS12, Kap. 6, p. 159]
 - Anwendung für I/O in Java: `java.io.OutputStream`

6. Prozesskommunikation

Quelle: [JS12] Kap. 6

Nachrichten-basierte Kommunikation

- Vorteile von asynchronem Senden
 - Nützlich für Realzeitanwendungen
 - Ermöglicht parallele Abarbeitung durch Sender und Empfänger
 - Anwendbar zum Signalisieren von Ereignissen
- Nachteile von asynchronem Senden
 - Verwaltung des Nachrichtenpuffers durch BS erforderlich
 - Benachrichtigung des Senders S im Fehlerfall und Behandlung von Fehlern ist problematisch
 - Entwurf und Nachweis der Korrektheit ist schwierig
- Erzeuger-Verbraucher-Problem
 - Programm, Petri-Netz [JS12, Kap. 6, p. 156]
 - Timeouts zur Lösung von unendlichem Warten

6. Prozesskommunikation

Quelle: [JS12] Kap. 6

The screenshot shows a PDF viewer window titled 'gbs_course-student.pdf'. The main content is a slide from a presentation, which is a duplicate of the slide on the left. The slide title is 'Nachrichten-basierte Kommunikation'. The slide content includes a list of advantages and disadvantages of asynchronous sending, and the producer-consumer problem. The slide also features the TUM logo and a source reference. At the bottom of the slide, there are two small diagrams: one labeled 'ack. received' and another labeled 'ack. sent'. The PDF viewer interface includes a navigation bar with 'Vorherige' and 'Nächste' buttons, a page number '158' out of '161', and a search bar. The TUM logo is visible at the top of the slide. At the bottom of the slide, there are two small diagrams: one labeled 'ack. received' and another labeled 'ack. sent'.