

Script generated by TTT

Title: Baumgarten: GBS (23.10.2013)

Date: Wed Oct 23 13:17:34 CEST 2013

Duration: 35:49 min

Pages: 26

gbs_course-student.pdf

TUM-GBS-Fol-2013-ALLE-1-68.pdf — TUM-GBS-Fol-2013.ppt [Kompatibilitätsmodus]

1 (1 von 68)

Vorschaubilder

1

2

3

Vorlesungen
gruppen.pdf

Technische Universität München

Grundlagen:
Betriebssysteme und Systemsoftware
(GBS)

Uwe Baumgarten

Vorlesungsunterlagen
(Student Script)

Mi, 23. Okt, 13:17

gbs_course-student.pdf

TUM-GBS-Fol-2013-ALLE-1-68.pdf — TUM-GBS-Fol-2013.ppt [Kompatibilitätsmodus]

1 (1 von 68)

Vorschaubilder

1

2

3

Vorlesungen
gruppen.pdf

Technische Universität München

Grundlagen:
Betriebssysteme und Systemsoftware
(GBS)

Uwe Baumgarten

Vorlesungsunterlagen
(Student Script)

Mi, 23. Okt, 13:17

gbs_course-student.pdf

TUM-GBS-Fol-2013-ALLE-1-68.pdf — TUM-GBS-Fol-2013.ppt [Kompatibilitätsmodus]

37 (37 von 68)

Vorschaubilder

34

35

36

37

Vorlesungen
gruppen.pdf

Technische Universität München

Inhalte

| | |
|---|------|
| 1. Übersicht | [38] |
| 2. Einführung | [45] |
| 3. Parallele Systeme – Modellierung, Strukturen | □ |
| 4. Prozess- und Prozessorverwaltung | □ |
| 5. Speicherverwaltung | □ |
| 6. Prozesskommunikation | □ |
| 7. Dateisysteme | □ |
| 8. Ein/Ausgabe | □ |
| 9. Sicherheit in Rechensystemen | □ |
| 10. Entwurf von Betriebssystemen | □ |
| 11. Zusammenfassung | □ |

Vorlesungsunterlagen
(Student Script)

Mi, 23. Okt, 13:18

gbs_course-student.pdf

TUM-GBS-Fol-2013-ALLE-1-68.pdf — TUM-GBS-Fol-2013.ppt [Kompatibilitätsmodus]

Technische Universität München

2. Einführung

- Betriebssysteme – Überblick [48]
 - BS-Hauptaufgaben
 - Systemprogrammierung
 - Hardwarekomponenten
 - Betriebsarten
- *Historie* [60]
- Betriebssystem-Architektur [61]
 - Monolithischer Ansatz & Mikrokern-Ansatz
 - BS-Beispiele
 - Systemaufrufe
 - Virtuelle Maschine
- Hardwarenahe Programme [67]
 - Definitionen
 - Programmaufbereitung
 - Binder und Lader

Quelle: [JS12] Kap. 2

Vorlesungen
gruppen.pdf

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

Mi, 23. Okt, 13:18

Technische Universität München

Betriebssystem-Architektur

- Monolithischer Ansatz
- Mikrokern-Ansatz & Schichten
- BS-Beispiele
- Systemaufrufe
- Virtuelle Maschine
- Betriebssysteme sind große und komplexe Softwaresysteme
- Konzepte
 - VM: Virtualisierung, Virtuelle Maschinen
 - OOP: Objektorientierte Programmierung
 - Modularisierung

2. Einführung

Quelle: [JS12] Kap. 2

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

61

Technische Universität München

Monolithischer Ansatz

- BS als umfangreiche Menge von Funktionen
- Wechselwirkungen (Aufrufe) dazwischen
- Zusammenfassung in einem BS-Kern
- Betreten des Kerns durch Systemaufrufe
- Eigenschaften
 - BS-Kern arbeitet im Systemmodus
 - BS-Kern hat hohe Ablaufpriorität
 - BS-Kern ist permanent im Arbeitsspeicher
- [JS12, Kap. 2, p. 17]

2. Einführung: Betriebssystem-Architektur

Quelle: [JS12] Kap. 2

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

62

Technische Universität München

Monolithischer Ansatz

- BS als umfangreiche Menge von Funktionen
- Wechselwirkungen (Aufrufe) dazwischen
- Zusammenfassung in einem BS-Kern
- Betreten des Kerns durch Systemaufrufe
- Eigenschaften
 - BS-Kern arbeitet im Systemmodus
 - BS-Kern hat hohe Ablaufpriorität

Betriebssystem-Architektur

Kap. 2

Vorlesungen
gruppen.pdf

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

Mi, 23. Okt, 13:30

gbs_course-student.pdf

1 (1 von 228)

Inhalt

- Übersicht 2
- Einführung 10
- Parallele Systeme - Modellierung... 32
- Prozess- und Prozessorverwaltung... 86
- Speicherverwaltung 113
- Prozesskommunikation 146
- Dateisysteme 175
- Ein-/Ausgabe 184
- Sicherheit in Rechensystemen 197
- Entwurf von Betriebssystemen 209
- Zusammenfassung 220

Grundlagen: Betriebssysteme und Systemsoftware (GBS)

Johann Schlichter
Institut für Informatik
TU München, Munich, Germany

Applied Informatics /
Cooperative Systems
AICOS

September 2012
Vorlesungsunterlagen
(Student Script)

Vorlesungen
gruppen.pdf

Mi, 23. Okt, 13:30

gbs_course-student.pdf

17 (24 von 228)

Inhalt

- Übersicht 2
- Einführung 10
- Betriebssystem - Überblick 10
- Betriebssystem-Architektur 16
- Monolithischer Ansatz 17
- Mikrokern-Ansatz 18
- Beispiel: BS-Architekturen 19
- Systemaufrufe 22
- Virtuelle Maschine 23
- Hardwarenahe Programme 24
- Parallele Systeme - Modell... 32
- Prozess- und Prozessorver... 86
- Speicherverwaltung 113
- Prozesskommunikation 146
- Dateisysteme 175
- Ein-/Ausgabe 184
- Sicherheit in Rechensyste... 197
- Entwurf von Betriebssystem... 209
- Zusammenfassung 220

Schlichter, TU München 2.2. BETRIEBSSYSTEM-ARCHITEKTUR

2.2.1 Monolithischer Ansatz

Das Betriebssystem besteht aus einer umfangreichen Menge an Funktionen, die sich bei Bedarf gegenseitig aufrufen können. Die Funktionen werden in einem großen BS-Kern zusammengefasst. Der BS-Kern wird durch Aufruf von Systemdiensten betreten.

BS-Kern arbeitet im Systemmodus.
Er hat hohe Ablaufpriorität.
Er ist permanent im Arbeitsspeicher.

```

graph TD
    subgraph User_Mode [Benutzer-Modus]
        A[Anwendung Benutzerprozess]
        B[Anwendung Benutzerprozess]
    end
    subgraph System_Mode [System-Modus Kernel-Mode]
        C[Systemdienste Hauptprozedur]
        D[Hilfsfunktionen]
    end
    subgraph Hardware
        E[Hardware]
    end
    A --> C
    B --> C
    C --> D
    D --> E
  
```

- komplexe, monolithische Betriebssysteme sind sehr schwierig zu warten und zu erweitern.
- Geschichtete Systeme**
Einen Ausweg aus der Problematik monolithischer Systeme bieten geschichtete Systeme:

Vorlesungen
gruppen.pdf

Mi, 23. Okt, 13:32

gbs_course-student.pdf

TUM-GBS-Fol-2013-ALLE-1-68.pdf — TUM-GBS-Fol-2013.ppt [Kompatibilitätsmodus]

62 (62 von 68)

Technische Universität München

Monolithischer Ansatz

- BS als umfangreiche Menge von Funktionen
- Wechselwirkungen (Aufrufe) dazwischen
- Zusammenfassung in einem BS-Kern
- Betreten des Kerns durch Systemaufrufe
- Eigenschaften
 - BS-Kern arbeitet im Systemmodus
 - BS-Kern hat hohe Ablaufpriorität
 - BS-Kern ist permanent im Arbeitsspeicher
- [JS12, Kap. 2, p. 17]

Quelle: [JS12] Kap. 2

© UB TUM GBS WS 2013/14 Grundlagen:

```

graph TD
    Hardware --> Systemdienste
    Systemdienste --> Hilfsfunktionen
  
```

- komplexe, monolithische Betriebssysteme sind sehr schwierig zu warten und zu erweitern.
- Geschichtete Systeme**
Einen Ausweg aus der Problematik monolithischer Systeme bieten geschichtete Systeme:

Vorlesungen
gruppen.pdf

Mi, 23. Okt, 13:32

gbs_course-student.pdf

TUM-GBS-Fol-2013-ALLE-1-68.pdf — TUM-GBS-Fol-2013.ppt [Kompatibilitätsmodus]

Technische Universität München

Mikrokern-Ansatz

- Grundkonzept: Geschichtete Systeme
- [JS12, Kap. 2, p. 18]
- Mikrokern
 - Wenige, grundlegende Funktionen
 - Kleiner Footprint (wenig Speicherplatzbedarf)
- [JS12, Kap. 2, p. 19]

Quelle: [JS12] Kap. 2

```

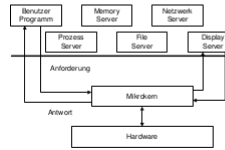
graph TD
    Hardware --> Systemdienste
    Systemdienste --> Hilfsfunktionen
  
```

- komplexe, monolithische Betriebssysteme sind sehr schwierig zu warten und zu erweitern.
- Geschichtete Systeme**
Einen Ausweg aus der Problematik monolithischer Systeme bieten geschichtete Systeme:

Vorlesungen
gruppen.pdf

Mi, 23. Okt, 13:32

werden in Serverprozessen im Benutzermodus realisiert, während der Mikrokernel wenige Basismechanismen umfasst.
Einfaches Austauschen von Subsystemen => ermöglicht die einfache Anpassung von Systemanforderungen.



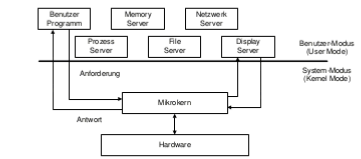
2.2.3 Beispiel: BS-Architekturen

Unix Betriebssystem

Die nachfolgende Abbildung skizziert die wesentlichen Komponenten des Unix Betriebssystems. Der Unix-BS-Kern enthält die Datei-, Prozess- und Prozessverwaltung, die Speicherverwaltung und die Geräte-Treiber.

| Inhalt | Seite |
|-------------------------------|-------|
| Übersicht | 2 |
| Einführung | 10 |
| Betriebssystem - Überblick | 10 |
| Betriebssystem-Architektur | 16 |
| Monolithischer Ansatz | 17 |
| Mikrokernel-Ansatz | 18 |
| Beispiel: BS-Architekturen | 19 |
| Systemaufrufe | 22 |
| Virtuelle Maschine | 23 |
| Hardwarenahe Programme | 24 |
| Parallele Systeme - Modell... | 32 |
| Prozess- und Prozessorver... | 86 |
| Speicherverwaltung | 113 |
| Prozesskommunikation | 146 |
| Dateisysteme | 175 |
| Ein-/Ausgabe | 184 |
| Sicherheit in Rechensyste... | 197 |
| Entwurf von Betriebssystem... | 209 |
| Zusammenfassung | 220 |

wenden in Serverprozessen im Benutzermodus realisiert, während der Mikrokernel wenige Basismechanismen umfasst.
Einfaches Austauschen von Subsystemen => ermöglicht die einfache Anpassung von Systemanforderungen.



2.2.3 Beispiel: BS-Architekturen

Unix Betriebssystem

Die nachfolgende Abbildung skizziert die wesentlichen Komponenten des Unix Betriebssystems. Der Unix-BS-Kern enthält die Datei-, Prozess- und Prozessverwaltung, die Speicherverwaltung und die Geräte-Treiber.

BS-Beispiele

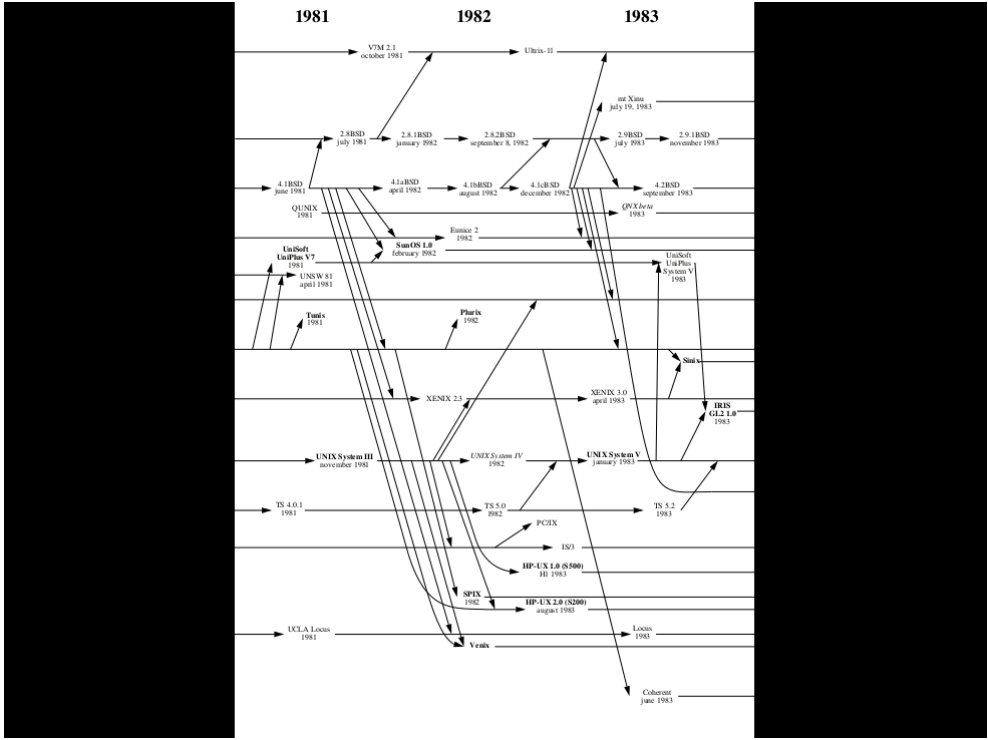
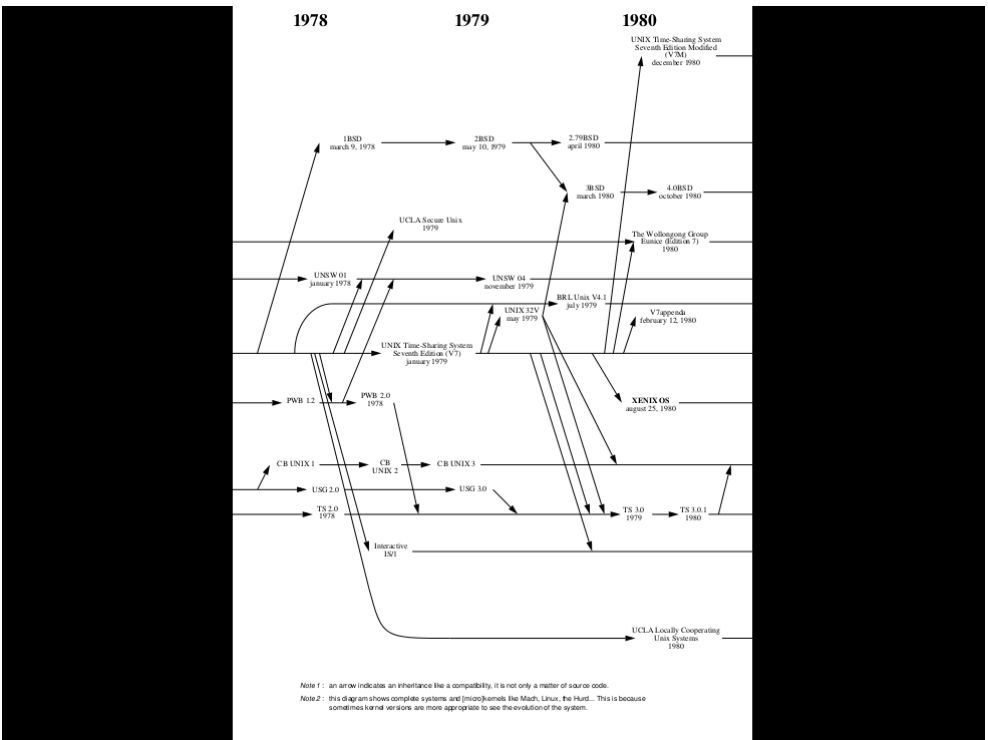
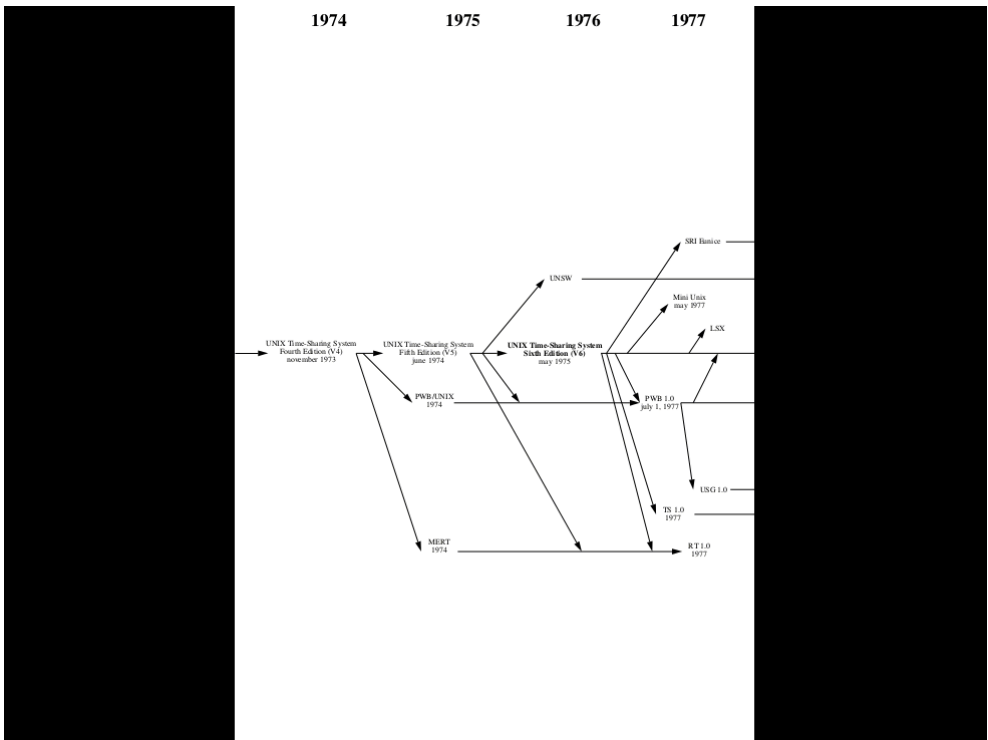
- UNIX Betriebssystem
 - BSD (Berkeley Systems Distribution)
 - [JS12, Kap. 2, p. 20]
- Windows NT
 - Und Nachfolger 2000, Vista, 7, 8
 - [JS12, Kap. 2, p. 21]
- Linux Betriebssystem
 - Modulare Struktur, dynamisches Binden
 - [JS12, Kap. 2, p. 22]

Quelle: [JS12] Kap. 2
<http://www.levenez.com/unix/>

1969 1970 1971 1972 1973

UNIX september 1969 → UNIX Time-Sharing System First Edition (V1) november 3, 1971 → UNIX Time-Sharing System Second Edition (V2) June 12, 1972 → UNIX Time-Sharing System Third Edition (V3) February 1973

Open System
Oktober 22, 2013
© Eric Levenez 1993/2013
<http://www.levenez.com/unix/>



File Edit View Go Search Help

Vorherige Nächste 30 (30 von 30) Auf Seitenbreite einpassen

Vorschaubilder

61

62

63

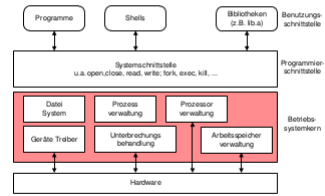
64

Vorlesungen

gruppen.pdf

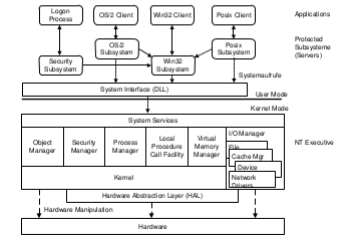
Menu TUM-G... gbs_c... Leven...

Mi, 23. Okt, 13:42



Windows NT Betriebssystem

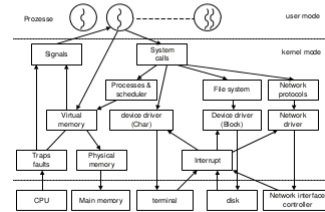
Mit Hilfe von HAL wird versucht, die meisten Maschinenabhängigkeiten zu verbergen. HAL präsentiert dem restlichen BS abstrakte Hardwaregeräte (z.B. Systembus, Arbeitsspeicher etc).



Linux Betriebssystem

Linux begann Anfang der 1990er Jahre als eine Unix Variante für den IBM PC; erste Version durch Linus Torvalds (1991). Linux ist frei und Quellcode ist verfügbar. kollaborative Weiterentwicklung durch Open Source Community.

- kein Mikrokern-Ansatz, jedoch modulare Struktur. dynamic linking. Module sind hierarchisch organisiert.
- jeder Modul ist durch 2 Datenstrukturen beschrieben. Modulbeschreibung, u.a Modulname, Größe, Zahl der exportierten Symbole und referenzierte Module. Symbol-Tabelle.



2.2.4 Systemaufrufe

Die Schnittstelle zwischen dem Betriebssystem und den Benutzerprogrammen wird durch eine Menge von Systemaufrufen (engl. system calls) gebildet. Sie stellen für Benutzerprogramme die einzige Schicht zum Zugriff auf Dienste des Betriebssystems und damit zur Nutzung der Hardware dar.

- in Benutzerprogrammen werden Systemaufrufe nicht direkt verwendet, sondern dies erfolgt über die Einbindung von Systembibliotheken, z.B. C-Library.

gbs_course-student.pdf

File Edit View

← Vorherige

Vorschaubilder

61

62

63

64

Vorlesungen

gruppen.pdf

Schlichter, TU München 2.2. BETRIEBSSYSTEM-ARCHITEKTUR

2.2.4 Systemaufrufe

Die Schnittstelle zwischen dem Betriebssystem und den Benutzerprogrammen wird durch eine Menge von Systemaufrufen (engl. system calls) gebildet. Sie stellen für Benutzerprogramme die einzige Schicht zum Zugriff auf Dienste des Betriebssystems und damit zur Nutzung der Hardware dar.

- in Benutzerprogrammen werden Systemaufrufe nicht direkt verwendet, sondern dies erfolgt über die Einbindung von Systembibliotheken, z.B. C-Library.

22

Menu TUM-G... gbs_c... Leven... Mi, 23. Okt, 13:48

gbs_course-student.pdf

TUM-GBS-Fol-2013-ALLE-1-68.pdf — TUM-GBS-Fol-2013.ppt [Kompatibilitätsmodus]

Technische Universität München

Systemaufrufe

- Schnittstelle zwischen Anwendungen (Benutzerprogrammen) und Betriebssystem
- Einbindung über Systembibliotheken (z.B. C-Library)
- Übergang in den Systemmodus
- Beispiele:
 - Prozess-V.: fork, waitpid, exit
 - Datei-V.: open, close, read, write
 - Verzeichnis-V.: mkdir, link, mount

Betriebssystem-Architektur

Kap. 2

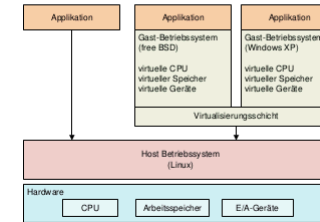
• in Benutzerprogrammen werden Systemaufrufe nicht direkt verwendet, sondern dies erfolgt über die Einbindung von Systembibliotheken, z.B. C-Library.

Vorlesungen

gruppen.pdf

TUM-G... TUM-G... Fol-2013.ppt [Kompatibilitätsmodus]

Mi, 23. Okt, 13:49



Die Java Virtual Machine (JVM) spezifiziert einen abstrakten Computer, auf dem Java Programme ausgeführt werden. JVM ist ein Softwaresystem, das auf dem Host-Betriebssystem ausgeführt wird.

2.3 Hardwarenahe Programme

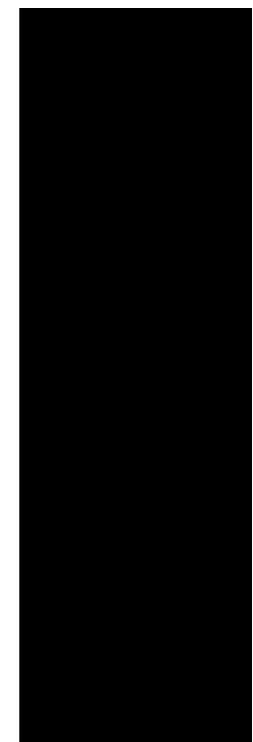
Zur Erleichterung werden hardwarenahe Programme für ein Rechnerystem nicht als ausführbare Maschinenprogramme (Folge von Befehlswörtern) realisiert, sondern mit Hilfe von Assemblerprogrammen.

2.3.1 Definitionen

Maschinenschnittstelle

Als Maschinenschnittstelle bezeichnet man die Gesamtheit aller Datenobjekte und Operationen der reinen Hardwarearchitektur (auch Programmierschnittstellen der Maschine).

- Folge von Maschinenbefehlen ist auf dieser Ebene eine Folge von Binärzeichen. Auf dieser Ebene müsste man insbesondere die Befehle der Maschine als reine



gbs_course-student.pdf

Schlichter, TU München 2.3 HARDWARENAHE PROGRAMME

Die Java Virtual Machine (JVM) spezifiziert einen abstrakten Computer, auf dem Java Programme ausgeführt werden. JVM ist ein Softwaresystem, das auf dem Host-Betriebssystem ausgeführt wird.

2.3 Hardwarenahe Programme

Zur Erleichterung werden hardwarenahe Programme für ein Rechnerystem nicht als ausführbare Maschinenprogramme (Folge von Befehlswörtern) realisiert, sondern mit Hilfe von Assemblerprogrammen.

2.3.1 Definitionen

Maschinenschnittstelle

Als Maschinenschnittstelle bezeichnet man die Gesamtheit aller Datenobjekte und Operationen der reinen Hardwarearchitektur (auch Programmierschnittstellen der Maschine).

- Folge von Maschinenbefehlen ist auf dieser Ebene eine Folge von Binärzeichen. Auf dieser Ebene müsste man insbesondere die Befehle der Maschine als reine

24

Vorlesungen

gruppen.pdf

TUM-G... gbs_c... Leven...

Mi, 23. Okt, 13:52