

Title: Baumgarten: GBS (08.01.2014)

Date: Wed Jan 08 13:15:08 CET 2014

Duration: 40:27 min

Pages: 15

# Grundlagen: Betriebssysteme und Systemsoftware (GBS)

Uwe Baumgarten

## Einführung (1)

- Differenzierung nach Betriebs-/Einsatzarten
  - Server-Systeme
  - Desktops und Laptops
  - Smartphones

} Linux
- Hauptaspekte
  - Definierte Abstraktionen
    - Z.B. Thread, Prozess, Datei
  - Bereitstellen einfacher Operationen
    - Zugang über Systemaufrufe
  - Sicherstellen der Abgrenzung
    - Eingrenzen von Fehlern durch Abgrenzung von Prozessen
  - Verwalten der Hardware

10. Entwurf von Betriebssystemen

Quelle: [JS12] Kap. 10

The screenshot shows a PDF viewer window titled 'gbs\_course-student.pdf'. The main content is a table of contents with the following items:

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	[168]
8	Fin-/Ausgabe	[177]

Below the table of contents, the section '9.3.4 Authentifizierung' is visible, with the text: 'Authentifizierung eines Nutzers erfolgt meist über Login-Name und dem zugehörigen Passwort. Generierung eines Hash Code aus dem Passwort und einem Streuwert fester'.

## Einführung (2)

- Probleme / Herausforderungen
  - Hardware kontinuierlich verbessert (Moore'sches Gesetz)
  - Betriebssysteme (teilweise eher) verschlechtert
- Herausforderungen
  - Betriebssysteme sind sehr komplex
  - Betriebssysteme müssen mit Parallelität umgehen
  - Betriebssysteme müssen mit „feindlichen“ Nutzern umgehen
  - Nutzer möchten Daten mit anderen Nutzern gemeinsam nutzen (sharing)
  - Betriebssysteme existieren eine lange Zeit
  - Entwickler wissen vorab oft nicht, wie die BSe genutzt werden
  - Betriebssysteme sind häufig portabel (unterschiedliche Hardware)
  - Rückwärtskompatibilität mit älteren BS-Versionen

10. Entwurf von Betriebssystemen

Quelle: [JS12] Kap. 10

## Schnittstellenentwurf – API für Dienste

- Leitlinien für den Entwurf
  - Prinzipien
    - Einfachheit
    - Vollständigkeit, aber Minimum an Mechanismen
    - Effizienz
- Nutzergruppen der APIs
  - Endbenutzer
    - Z.B. Fensteroberfläche, GUI
  - Programmierer
    - Systemaufrufchnittstelle
  - Systemadministration
    - Aufruf von Systemdiensten

10. Entwurf von Betriebssystemen

Quelle: [JS12] Kap. 10

## Systemaufrufchnittstellen

- Paradigmen
  - Ausführungsorientiert
  - Datenorientiert
- Algorithmischer Ansatz
  - Basislogik im Code festgelegt
  - Systemaufrufe mit Parameterübergabe
  - [JS12, Kap. 10, p. 211]
- Ereignisbasierter Ansatz
  - Anwendungen reagieren auf Ereignisse
  - Ereignisse werden durch BS erfasst und der Anwendung zugestellt
  - [JS12, Kap. 10, p. 212]
- Datenparadigma
  - „Alles“ ist eine Datei (UNIX-Ursprung)
  - „Alles“ ist ein Objekt (Microsoft, Windows (NT, 2000, usw.))

10. Entwurf von Betriebssystemen

Quelle: [JS12] Kap. 10

## Systemaufrufchnittstellen

- Paradigmen
  - Ausführungsorientiert
  - Datenorientiert
- Algorithmischer Ansatz
  - Basislogik im Code festgelegt
  - Systemaufrufe mit Parameterübergabe
  - [JS12, Kap. 10, p. 211]
- Ereignisbasierter Ansatz
  - Anwendungen reagieren auf Ereignisse
  - Ereignisse werden durch BS erfasst und der Anwendung zugestellt

10. Entwurf von Betriebssystemen

Kap. 10

### 9.3.4 Authentifizierung

Authentifizierung eines Nutzers erfolgt meist über Login-Name und dem zugehörigen Passwort.

Generierung eines Hash Code aus dem Passwort und einem Streuwert fester

gbs\_course-student.pdf

Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe

Vorherige Nächste 211 (218 von 228) 150%

Vorschaubilder

200

201

202

203

Vorlesungen

gruppen.pdf

Benutzereingaben zu erhalten  
BS-Dienste in Anspruch zu nehmen

**Beispiel**

```
main () {
    int ...;
    init();
    do_something();

    read(...); /* Systemaufruf */
    do_something_else();

    write(...); /* Systemaufruf */
    continue();
    exit(0);
}
```

211

Menu gbs\_course-s... TUM-GBS-Fol-... Mi. 08. Jan, 13:37

## Systemaufrufsschnittstellen

- Paradigmen
  - Ausführungsorientiert
  - Datenorientiert
- Algorithmischer Ansatz
  - Basislogik im Code festgelegt
  - Systemaufrufe mit Parameterübergabe
  - [JS12, Kap. 10, p. 211]
- Ereignisbasierter Ansatz
  - Anwendungen reagieren auf Ereignisse
  - Ereignisse werden durch BS erfasst und der Anwendung zugestellt
  - [JS12, Kap. 10, p. 212]
- Datenparadigma
  - „Alles“ ist eine Datei (UNIX-Ursprung)
  - „Alles“ ist ein Objekt (Microsoft, Windows (NT, 2000, usw.))

10. Entwurf von Betriebssystemen

Quelle: [JS12] Kap. 10

## Weitere Implementierungsaspekte (1)

- Architektur
  - Geschichtete Systeme
  - [JS12, Kap. 10, p. 214]
  - Alternativen
    - Modular
    - Objektorientiert
- Trennung von Mechanismen und Strategien (Policies)
  - Z.B.: Prioritätenbasierter Scheduler
    - Mechanismus: Realisierung als sortiertes Feld
    - Strategie: Festlegung der Prioritäten
  - Z.B.: Paging beim virtuellen Speicher
    - Mechanismus: Verwaltung der Seitentabelle, MMU, etc.
    - Strategie: Auslagerung nach FIFO, LRU, etc.

10. Entwurf von Betriebssystemen

Quelle: [JS12] Kap. 10

gbs\_course-student.pdf

TUM-GBS-Fol-2013-ALLE-1-207.pdf — TUM-GBS-Fol-2013.ppt [Kompatibilitätsmodus]

Technische Universität München

## Weitere Implementierungsaspekte (1)

- Architektur
  - Geschichtete Systeme
  - [JS12, Kap. 10, p. 214]
  - Alternativen
    - Modular
    - Objektorientiert
- Trennung von Mechanismen und Strategien (Policies)
  - Z.B.: Prioritätenbasierter Scheduler
    - Mechanismus: Realisierung als sortiertes Feld

case 2: ...;  
case 3: ...;  
.....  
}

Vorlesungen

gruppen.pdf

Menu gbs\_course-s... TUM-GBS-Fol-... Mi. 08. Jan, 13:43

10. Entwurf von Betriebssystemen

Kap. 10

## Weitere Implementierungsaspekte (1)

10. Entwurf von Betriebssystemen

- Architektur
  - Geschichtete Systeme
  - [JS12, Kap. 10, p. 214]
  - Alternativen
    - Modular
    - Objektorientiert
- Trennung von Mechanismen und Strategien (Policies)
  - Z.B.: Prioritätenbasierter Scheduler
    - Mechanismus: Realisierung als sortiertes Feld
    - Strategie: Festlegung der Prioritäten
  - Z.B.: Paging beim virtuellen Speicher
    - Mechanismus: Verwaltung der Seitentabelle, MMU, etc.
    - Strategie: Auslagerung nach FIFO, LRU, etc.

Quelle: [JS12] Kap. 10

## Weitere Implementierungsaspekte (2)

10. Entwurf von Betriebssystemen

- Namensräume
  - Namen für langlebige „Datenstrukturen“ (Objekte)
  - Ebenen-Organisation
    - externe Namen (z.B. Dateiname)
    - interne Namen (z.B. inode Nummer)
  - Vergleiche: „Adresse“ vs. „Bezeichner“ vs. „Identifikator“
- Statische vs. Dynamische Datenstrukturen
  - Statisch
    - Vorteile: einfacher, schneller zugreifbar
  - Dynamisch
    - Vorteile: flexibler
  - Abwägung nach Einsatzzweck

Quelle: [JS12] Kap. 10

## Weitere Implementierungsaspekte (3)

10. Entwurf von Betriebssystemen

- Verbergen der Hardware
  - Z.B.: HAL (Hardware Abstraction Layer)
  - Abfrage der AS-Größe zum Bootzeitpunkt
  - CPU-abhängige Übersetzung
    - [JS12, Kap. 10, p. 216]
  - Architektur-abhängige Übersetzung
    - [JS12, Kap. 10, p. 216]
- Speicherplatz vs. Laufzeit
  - Nutzung von Prozeduren (Funktionen) [JS12, Kap. 10, p. 216]
  - Nutzung von Makros (Shiften und Addieren) [JS12, Kap. 10, p. 216]
  - Nutzung von Makros (Shiften und Addieren) [JS12, Kap. 10, p. 217]

Quelle: [JS12] Kap. 10

## Weitere Implementierungsaspekte (3)

10. Entwurf von Betriebssystemen

- Verbergen der Hardware
  - Z.B.: HAL (Hardware Abstraction Layer)
  - Abfrage der AS-Größe zum Bootzeitpunkt
  - CPU-abhängige Übersetzung
    - [JS12, Kap. 10, p. 216]
  - Architektur-abhängige Übersetzung
    - [JS12, Kap. 10, p. 216]
- Speicherplatz vs. Laufzeit
  - Nutzung von Prozeduren (Funktionen) [JS12, Kap. 10, p. 216]

Kap. 10

Abstraktion der Hardware (z.B. HAL von Windows)

Vorlesungen

gruppen.pdf

### 10.3.2 Mechanismen vs. Policies