



## Script generated by TTT

Title: Grundlagen\_Betriebssysteme (25.01.2016)

Date: Mon Jan 25 13:45:30 CET 2016

Duration: 59:55 min

Pages: 17

Schutz von gespeicherter Information vor Diebstahl, unerwünschter Manipulation und Verletzung der Vertraulichkeit ist ein zentrales Anliegen in allen Mehrbenutzersystemen.

[Anforderungen](#)

[Ebenen des Zugriffsschutzes](#)

[Schutzmatrix](#)

[Authentifizierung](#)

*Generated by Targeteam*



Schutz von gespeicherter Information vor Diebstahl, unerwünschter Manipulation und Verletzung der Vertraulichkeit ist ein zentrales Anliegen in allen Mehrbenutzersystemen.

[Anforderungen](#)

[Ebenen des Zugriffsschutzes](#)

[Schutzmatrix](#)

[Authentifizierung](#)

*Generated by Targeteam*



Applet wird als Byte-Code geladen. Jeder Befehl wird vor seiner Ausführung von der Java Virtual Machine (JVM) analysiert.

jeder Systembefehl wird abgefangen und untersucht.

für vertrauenswürdige Applets (z.B. geladen von lokaler Festplatte) werden Systembefehle ausgeführt.

nicht vertrauenswürdige Applets (z.B. geladen über Internet) werden innerhalb einer Sandbox ausgeführt.

*Generated by Targeteam*



Es werden nur Applets von **vertrauenswürdigen Quellen** geladen und ausgeführt. Der Applet-Code wird mit einer **digitalen Unterschrift** versehen, um zu garantieren, dass der Code der vertrauenswürdigen Quelle nicht verändert wurde.

digitale Unterschrift basiert auf public-key Verfahren.

Erzeugung der Unterschrift durch vertrauenswürdige Quelle

Hashfunktion erzeugt von Applet-Code eine 128/160 bit Zahl.

erzeugte Hashzahl wird mit privatem Schlüssel der Quelle verschlüsselt.

digitale Unterschrift wird mit Applet-Code verschickt.

Überprüfung der Unterschrift

Browser führt auf Applet-Code Hashfunktion aus und berechnet selbst Hashzahl.

Browser entschlüsselt Unterschrift mit öffentlichem Schlüssel der vertrauenswürdigen Quelle.

berechnete Hashzahl und Hashzahl in Unterschrift müssen übereinstimmen.

Generated by Targeteam



Das Internet führt zunehmend zu einer Verbreitung von mobilem Code. Beispiele sind

Web Seiten mit Applets

Postscript Dateien

mobile Software-Agenten (z.B. in Ecommerce Anwendungen).

Ausführung von heruntergeladenem Code birgt Risiken in sich. Methoden (anhand von Applets), um mit dieser Problematik umzugehen:

[Sandboxing](#)

[Interpretation](#)

[Signed Code](#)

Generated by Targeteam



- Prof. J. Schlichter
  - Lehrstuhl für Angewandte Informatik / Kooperative Systeme, Fakultät für Informatik, TU München
  - Boltzmannstr. 3, 85748 Garching
  - Email: [schlichter@in.tum.de](mailto:schlichter@in.tum.de)
  - Tel.: 089-289 18654
  - URL: <http://www11.informatik.tu-muenchen.de/>

[Übersicht](#)

[Einführung](#)

[Parallele Systeme - Modellierung, Strukturen](#)

[Prozess- und Prozessorverwaltung](#)

[Speicherverwaltung](#)

[Prozesskommunikation](#)

[Dateisysteme](#)

[Ein-/Ausgabe](#)

[Sicherheit in Rechensystemen](#)

[Entwurf von Betriebssystemen](#)

[Zusammenfassung](#)

Generated by Targeteam



Der Entwurf eines BS erfordert ein ingenieurmäßiges Vorgehen. Es gibt jedoch keine wohl-definierten Vorgehensweisen, sondern nur Erfahrungen von Entwicklern bzw. Nutzern.

**Einführung**

Die Ziele von Betriebssystemen können sich zwischen verschiedenen Systemen unterscheiden, für Server-Systeme, für Laptops, für Smartphones oder für eingebettete Systeme.

[Hauptaspekte](#)

[Probleme](#)

**Schnittstellenentwurf**

Ein BS bietet eine Reihe von Diensten, die von Benutzerprozessen in Anspruch genommen werden können.

Bereitstellen der Dienste über Schnittstellen.

[Leitlinien für den Entwurf](#)

**Paradigmen der Systemaufrufchnittstelle**

Für die Einbindung der Systemaufrufe in Nutzerprogramme kann man zwischen den Ausführungs- und den Datenparadigmen unterscheiden.

[Algorithmischer Ansatz](#)

[Ereignis-basierter Ansatz](#)

[Datenparadigma](#)

[Systemaufrufchnittstelle](#)

[Weitere Implementierungsaspekte](#)

[Trends beim Entwurf von Betriebssystemen](#)

Generated by Targeteam



Man kann die folgenden 4 Hauptaspekte unterscheiden

Definierte Abstraktion:

z.B. Thread, Prozess, Datei

Bereitstellen einfacher Operationen:

Operationen zum Manipulieren der zur Abstraktion gehörigen Datenstrukturen.

Ansprechbar über Systemaufrufe.

Sicherstellen der Abgrenzung:

Eingrenzung von Fehlern durch Abgrenzung von Prozessen.

Verwalten der Hardware.

Generated by Targeteam



Hardware hat sich gemäß des Moore'schen Gesetzes kontinuierlich verbessert; Betriebssysteme haben zwar mehr Funktionalität, jedoch bzgl. Verfügbarkeit sind sie teilweise schlechter als die alten Systeme.

Betriebssysteme sind sehr komplex.

Betriebssysteme müssen mit Parallelität umgehen.

Betriebssysteme müssen mit feindlichen Nutzern umgehen.

Nutzer möchten Daten mit anderen Nutzern gemeinsam nutzen.

Betriebssysteme existieren eine lange Zeit.

Entwickler wissen oft nicht vorab, wie ihre Systeme genutzt werden.

Betriebssysteme sind portabel für unterschiedliche Hardware entworfen.

Kompatibilität mit älteren Betriebssystem Versionen.

Generated by Targeteam



Der Entwurf eines BS erfordert ein ingenieurmäßiges Vorgehen. Es gibt jedoch keine wohl-definierten Vorgehensweisen, sondern nur Erfahrungen von Entwicklern bzw. Nutzern.

### Einführung

Die Ziele von Betriebssystemen können sich zwischen verschiedenen Systemen unterscheiden, für Server-Systeme, für Laptops, für Smartphones oder für eingebettete Systeme.

[Hauptaspekte](#)

[Probleme](#)

### Schnittstellentwurf

Ein BS bietet eine Reihe von Diensten, die von Benutzerprozessen in Anspruch genommen werden können.

Bereitstellen der Dienste über Schnittstellen.

[Leitlinien für den Entwurf](#)

### Paradigmen der Systemaufrufchnittstelle

Für die Einbindung der Systemaufrufe in Nutzerprogramme kann man zwischen den Ausführungs- und den Datenparadigmen unterscheiden.

[Algorithmischer Ansatz](#)

[Ereignis-basierter Ansatz](#)

[Datenparadigma](#)

[Systemaufrufchnittstelle](#)

[Weitere Implementierungsaspekte](#)

[Trends beim Entwurf von Betriebssystemen](#)

Generated by Targeteam



Es existieren eine Reihe von Prinzipien und Empfehlungen

Einfachheit.

Vollständigkeit.

Bereitstellen, was man benötigt und nicht mehr.

Minimum an Mechanismen.

Effizienz.

### Nutzergruppen von BS

Endbenutzer

Nutzung von Anwendungen  $\Rightarrow$  grafische Oberfläche des BS

Programmierer

Nutzung der Systemaufrufchnittstelle des BS

Systemadmin

Aufruf von Systemdiensten

Generated by Targeteam



Der Entwurf eines BS erfordert ein ingenieurmäßiges Vorgehen. Es gibt jedoch keine wohl-definierten Vorgehensweisen, sondern nur Erfahrungen von Entwicklern bzw. Nutzern.

**Einführung**

Die Ziele von Betriebssystemen können sich zwischen verschiedenen Systemen unterscheiden, für Server-Systeme, für Laptops, für Smartphones oder für eingebettete Systeme.

Hauptaspekte

Probleme

**Schnittstellentwurf**

Ein BS bietet eine Reihe von Diensten, die von Benutzerprozessen in Anspruch genommen werden können.

Bereitstellen der Dienste über Schnittstellen.

Leitlinien für den Entwurf

**Paradigmen der Systemaufrufchnittstelle**

Für die Einbindung der Systemaufrufe in Nutzerprogramme kann man zwischen den Ausführungs- und den Datenparadigmen unterscheiden.

Algorithmischer Ansatz

Ereignis-basierter Ansatz

Datenparadigma

Systemaufrufchnittstelle

Weitere Implementierungsaspekte

Trends beim Entwurf von Betriebssystemen

*Generated by Targateam*



basiert auf der Idee, dass ein Programm eine bestimmte Funktion erfüllen soll, die es im Voraus oder durch Parameter kennt.

Basislogik ist im Code festgelegt

Gelegentliche Systemaufrufe, um

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);
}
```

*Generated by Targateam*



nach einer Initialisierung warten Programme dieses Ansatzes auf Ereignisse des Betriebssystems

Anwendungen reagieren auf Ereignisse

Ereignisse werden durch BS erfasst und der Anwendung zugestellt

Ereignisse sind z.B.

Tastendruck

Mausbewegung

**Beispiel**

```
main () {
    mess_t msg;
    init();
    while (get_message(&msg)) {
        case 1: ...;
        case 2: ...;
        case 3: ...;
        .....
    }
}
```

*Generated by Targateam*



Wie werden Systemstrukturen und Geräte im Betriebssystem gegenüber dem Programmierer präsentiert?

**Beispiel Unix**

"Alles" ist eine Datei

Vereinheitlichung von Dateien, E/A-Geräte und Pipes

Zugriff über Dateioperationen

```
fd1 = open("file1", O_RDWR)
```

```
fd2 = open("/dev/tty", O_RDWR)
```

**Beispiel Windows 2000**

"Alles" ist ein Objekt.

*Generated by Targateam*



Entscheidung bzgl. der Bereitstellung unterschiedlicher Varianten von Systemaufrufen

Z.B. Lese-Systemaufruf: `read_file`, `read_process`, `read_tty`, ...

Besser: nur Bereitstellung des allgemeinen Falls

Z.B. Lese-Systemaufruf: `read`

Restliche Varianten über Bibliotheksfunktionen auf den allgemeinen Systemaufruf abbilden

Sichtbarkeit von Systemaufrufen

Trennung von Systemaufrufen und Bibliotheksaufrufen

⇒ Ermöglicht übersichtliche Darstellung der vorhandenen Systemaufrufe

*Generated by Targemam*

Der Entwurf eines BS erfordert ein ingenieurmäßiges Vorgehen. Es gibt jedoch keine wohl-definierten Vorgehensweisen, sondern nur Erfahrungen von Entwicklern bzw. Nutzern.

### Einführung

Die Ziele von Betriebssystemen können sich zwischen verschiedenen Systemen unterscheiden, für Server-Systeme, für Laptops, für Smartphones oder für eingebettete Systeme.

### Hauptaspekte

### Probleme

### Schnittstellenentwurf

Ein BS bietet eine Reihe von Diensten, die von Benutzerprozessen in Anspruch genommen werden können.

Bereitstellen der Dienste über Schnittstellen.

### Leitlinien für den Entwurf

### Paradigmen der Systemaufrufchnittstelle

Für die Einbindung der Systemaufrufe in Nutzerprogramme kann man zwischen den Ausführungs- und den Datenparadigmen unterscheiden.

### Algorithmischer Ansatz

### Ereignis-basierter Ansatz

### Datenparadigma

### Systemaufrufchnittstelle

### Weitere Implementierungsaspekte

### Trends beim Entwurf von Betriebssystemen

*Generated by Targemam*