

Einführung in die Betriebssysteme

Wir brauchen ein Betriebssystem! Nur Zusammenspiel von Hard- und Software ergibt am Ende ein Hochleistungssystem, welches gleichzeitig flexibel und bedienbar ist. Ein Betriebssystem ist ein Programm, das dem Benutzer und Anwendungsprogrammen elementare Dienste bereitstellt. Das Betriebssystem steuert und überwacht die Abwicklung von Programmen und regelt den Betrieb des Rechnersystems.

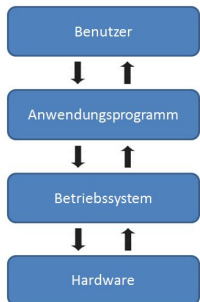
- Zentrale Aufgabe eines BS
- Zwischen Benutzer und Hardware
- Wichtige Fachbegriffe
- Betriebsmittel / Betriebssystemarten
- Batch-Job zum Multitasking
- Abarbeitung von Befehlen auf CPU
- Systemaufruf durch Programmierer

Zentrale Aufgabe eines BS

Die **zentrale Aufgabe** eines Betriebssystems ist die **Betriebsmittelverwaltung**.

- **Verwaltung, Aus- und Einlagern mehrerer Prozesse**
- **Verwaltung des Hauptspeichers und Versorgung Prozesse mit Teilen des Hauptspeichers**
- **Auswahl des jeweils nächsten Prozesses für die CPU**
- **Datei-/Verzeichnisverwaltung auf Datenträgern und Verwaltung Ein-/Ausgabegeräte**

Zwischen Benutzer und Hardware



- Der **Benutzer** bedient ein / mehrere **Anwendungsprogramme**
- **Anwendungsprogramme** geben Rückmeldungen an den **Anwender** oder nutzt **Betriebssystem**, um mit **Hardware** zu kommunizieren.
- **Betriebssystem** empfängt **Aufträge** von **Anwendungsprogrammen**.
- **AP** erhalten Rückmeldungen vom **BS** über abgelehnten und ausgeführten **Aufträge**.
- Die **Hardware** führt Befehle aus, die ihr vom **BS** in Auftrag gegeben wurden. Das **BS** erhält Rückmeldungen der **Hardware**. Auch bei Hardware-Fehlern wird das Betriebssystem informiert.

Wichtige Fachbegriffe

Fachbegriff	Bedeutung
<i>Anwendungsprogramm</i>	löst ein/mehrere Probleme, die Benutzer stellt (Textverarbeitung / Client / Browser)
<i>Systemprogramm</i>	Computerprogramm, welches der Verwaltung des Betriebs eines Computers dient
<i>Betriebssystem</i>	stellt Dienste; steuert Ablauf von Programmen; regelt den Rechnersystembetrieb.
<i>Betriebsmittel</i>	Hardware- oder Software-Ressource.
<i>Hardware-Ressource</i>	einzelne Hardware-Komponente des Rechners (CPU / RAM / Controller)
<i>Software-Ressource</i>	Prozess oder Datei auf einem beliebigen Datenträger
<i>Betriebsmittel Prozess</i>	Programmtext, Hauptspeicherdaten (Stack, Heap) und auf CPU (Registerinhalte).
<i>Betriebsmittel Datei</i>	Programmdatei oder Datendatei
<i>Entziehbares Betriebsmittel</i>	Betriebsmittel, das dem Prozess ohne negative Folgen entzogen könnte (z.B: CPU)
<i>Nicht entziehbares Betriebsmittel</i>	Betriebsmittel, das Prozess so lange zur Verfügung steht, wie dieser es benötigt.
<i>exklusiv nutzbaren Betriebsmittel</i>	zu beliebigen Zeitpunkt nur maximal einem Prozess zugeordnet (z.B: Drucker)
<i>gemeinsam nutzbar Betriebsmittel</i>	quasi -gleichzeitig von mehreren Prozessen genutzt werden kann (z.B: Festplatte)

Betriebsmittel / Betriebssystemarten

Ein Programm kann mehrfach gestartet werden, somit können aus einem Programm so mehrere Prozesse resultieren!

Jeder Prozess besitzt eigene Daten, auf die nur er Zugriff hat und mehrere Prozesse müssen sich auf der CPU abwechseln

Prozesse benötigen während ihrer **Abarbeitung verschiedene Betriebsmittel**, um alle anfallenden *Aufgaben erledigen* zu können. Damit sind jedem gestarteten **Prozess eine Menge von Betriebsmitteln** zugewiesen.

→ Ein Prozess benötigt (mindestens) die Betriebsmittel *Festplatte, ausführbare Datei X* sowie *RAM*, um aus der Datei X auf der Festplatte in den Arbeitsspeicher geladen zu werden.

→ Ein Prozess benötigt das **Betriebsmittel CPU** um befehlsweise abgearbeitet zu werden.

→ Ein Prozess benötigt das Betriebsmittel **Grafikkarte** um eine Ausgabe auf dem Bildschirm zu erzeugen.

→ sind nicht alle Betriebsmittel während der Laufzeit mit dem Prozess verbunden

→ effizienter, wenn Betriebsmittel nur dann angefordert werden, wenn sie tatsächlich **benötigt** werden

Betriebsmittel sind Prozessen zugeordnet

Betriebsmittel für Prozess	wird benötigt, um ...
<i>Festplatte, ausführbare Datei X sowie RAM</i>	aus der Datei X auf der Festplatte in den Arbeitsspeicher geladen zu werden
<i>CPU</i>	befehlsweise abgearbeitet zu werden
<i>Grafikkarte</i>	eine Ausgabe auf dem Bildschirm zu erzeugen

- **nicht** alle Betriebsmittel sind während der gesamten Laufzeit mit Prozess verbunden
 - effizienter, wenn Betriebsmittel nur dann angefordert werden, wenn benötigt und wieder freigegeben werden

Betriebssystemarten

Betriebssysteme für	Beschreibung
<i>Großrechner (z/OS von IBM)</i>	<ul style="list-style-type: none">• verarbeitet schnell viele Anfragen mit hoher E/A-Rate und Speicherkapazität• im Stapelbetrieb (Batch-Jobs) oder sind transaktionsorientiert
<i>Server (Unix/Linux)</i>	<ul style="list-style-type: none">• werden über Netzwerk von verschiedenen Clients angesprochen• Anwender an den Clients erwarten schnelle Reaktionszeit• Einsatzzwecke liegen beispielsweise bei: File-Server / Print-Server
<i>Laptops/Personal Computer (Windows 10)</i>	<ul style="list-style-type: none">• Anwender tätigt eine Eingabe und Betriebssystem reagiert darauf.

Batch-Job zum Multitasking

Batch-Jobs

- In den Anfangsjahren der Rechnerentwicklung wurden auszuführende Programme als sogenannte Batch-Jobs verarbeitet
- Jeder **Prozess** (*Batch-Job*) wurde nach dem Start komplett bis zu seinem Ende bearbeitet,
- erst im **Anschluß** startete der *nächste Prozess* (Batch-Job) und **belegte** die **CPU** ebenfalls bis zu seiner Terminierung.

Multitasking

- mehrere Prozesse können sich bei ihrer Abarbeitung auf der CPU abwechseln
- Diese geänderte Vorgehensweise markiert einen gewaltigen *Meilenstein in der Informatik*

Abarbeitung von Befehlen auf CPU

Kernel-Mode	<ul style="list-style-type: none">• Arbeitet CPU Kernel-Mode, ist jeder Befehl zur Ausführung zugelassen• kann auf Speicherbereiche für Daten- und Programmtext und auf Betriebsmittel zugreifen• Durch ein Steuer- oder Kontrollregister auf der CPU wird der Kernel-Mode angezeigt.• Betriebssystem arbeitet im Kernel-Mode
User-Mode	<ul style="list-style-type: none">• Arbeitet CPU im User-Mode, ist nur eingeschränkter Befehlssatz zugelassen• nicht alle Befehle erlaubt, kann nicht auf alle Speicherbereiche und Betriebsmittel zugreifen• Durch ein Steuer- oder Kontrollregister auf der CPU wird der User-Mode angezeigt.• Anwendungsprogramme arbeiten im User-Mode

- **Übergang vom Kernel-Mode in User-Mode** ist unproblematisch, da hierbei die Rechte eingeschränkt werden
 - Das Betriebssystem veranlasst Übergang, wenn es dem Prozess eines Anwendungsprogramms die CPU zuteilt.
- **Übergang vom User-Mode in den Kernel-Mode** funktioniert nur indirekt über einen **Systemaufruf**
 - denn Prozess eines Anwendungsprogramms darf nicht über die umfassenden Rechte des Kernel-Modus verfügen

Unter einem **Systemaufruf** versteht man den von einem im User-Mode ablaufenden Prozess getätigten Aufruf einer vom Betriebssystem zur Verfügung gestellten Funktion, welche nur im Kernel-Mode ausgeführt werden kann.

- Durch **Systemaufruf** gibt der im **User-Mode Prozess Kontrolle** zurück an **BS**, welches die Funktion stellvertretend ausführt.
- **BS** führt Sicherheitsüberprüfungen durch und prüft, ob alle Gegebenheiten für die Ausführung sprechen.
- Ist **Funktion ausgeführt**, wird zur Beendigung des Systemaufrufs wieder **in den User-Mode zurückgeschaltet**

- **BS** gibt **Kontrolle** zurück an aufrufenden Prozess, der ein Ergebnis seines Systemaufrufs zum Reagieren empfängt
-

Beispiel: Ein Anwendungsprogramm möchte auf eine Datei zugreifen.

- im **User-Mode** ablaufender **Prozess** eines **Anwendungsprogramms** möchte auf eine Datei zugreifen
 - Diese **E/A-Handlung** ist nur im **Kernel-Mode** zugelassen.
 - **Betriebssystem** stellt **Anwendungsprogramm** diese **Systemaufrufen** zur Verfügung: **open** | **read** | **write** | **close**
- Wird jetzt ein **Systemaufruf** getätigt, so entspricht dies der Auslösung eines **Interrupts**
 - der im **User-Mode** laufende **Prozess** wird angehalten
 - die **Interrupt-Service-Routine** wird ausgeführt (dadurch wird **Betriebssystem**-Code ausgeführt)
 - es wird in den **Kernel-Mode** geschaltet und Überprüfungen werden durchgeführt
 - der Dateizugriff wird entweder erlaubt und durchgeführt oder verweigert,
 - es wird in den **User-Mode** zurückgeschaltet,
 - angehaltene **Prozess** wird wieder gestartet und ihm wird Rückgabewert mit Ergebnis des Systemaufrufs gegeben

Systemaufruf durch Programmierer

```
import java.io.FileWriter;
import java.io.IOException;

public class Beispiel_Systemaufrufe {
    public static void main(String[] args) {
        FileWriter fw;
        try { // Hier folgen drei Systemaufrufe: Anwendung wird angehalten, nach Überprüfung
            // sendet BS Befehle richtung Festplattencontroller
            fw = new FileWriter("hallo.txt"); // Datei wird zum Schreiben geöffnet
            fw.write("Hallo Systemaufruf!"); // Hallo Systemaufruf! wird in geöffnete Datei
            // geschrieben
            fw.close(); //Datei wird wieder geschlossen
        } catch (IOException e) { // BS sorgt dafür, dass Fehlermeldung generiert wird
            System.out.println("Es ist ein Fehler aufgetreten: \r\n" + e.toString());
        }
        // CPU von Kernelmode wieder auf Usermode, Programm läuft weiter
    }
}
```

Der Anwender muss über Systemaufrufe nichts wissen, er nutzt nur Methoden, alles weitere geschieht im Hintergrund.