

Konzepte von Betriebssystem-Komponenten

I/O: von der Platte zur Anwendung

SS 05

Igor Engel

Igor.Engel@informatik.stud.uni-erlangen.de

- 1 Einleitung
- 2 Übersicht
- 3 Systemaufrufe
 Beispiel in Unix
- 4 Dateien
 4.1 Dateisysteme
 4.2 Transport der Daten
- 5 Festplattentreiber
- 6 Festplattencontroller
- 7 Festplatte
- 8 Beispiel der I/O Programmierung
- 9 Quellen

1 Einleitung

Das Betriebssystem verbirgt die Einzelheiten der Festplatte und repräsentiert ein klares und abstraktes Modell geräteunabhängiger Dateien, auf die Anwendungen zugreifen und auf Daten arbeiten können. Die Festplatte liefert einen sequenziellen Bitstrom. Anwendungen kennen aber nur Dateien. Im Folgenden wird näher erläutert, wie das Betriebssystem hier die Rolle des Vermittlers spielt.

2 Übersicht

Der Weg von oben nach unten:

- Anwendungen greifen auf die Dateien über das Dateisystem zu,
- Dateien bestehen aus Blöcken,
- um Dateien zu lesen bzw. zu schreiben, werden Systemaufrufe aufgerufen,
- das Dateisystem gibt Lese- und Schreibbefehle für bestimmte Blöcke an den Plattentreiber weiter,
- der Plattentreiber stellt fest, welcher Befehl ausgeführt werden muss, übersetzt die Blockadressen in reale Adressen auf der Festplatte und schreibt hardwarenahe Anweisungen in die Steuerregister des Festplattencontrollers,
- der Controller führt dann die Anweisungen aus seinen Registern aus, d.h. greift lesend bzw. schreibend auf die Platte zu.

Der Weg von unten nach oben:

- Der Controller liest Bit für Bit von der Platte und konvertiert den Bitstrom in Byte-Blöcke,
- danach werden diese Blöcke Schicht für Schicht nach oben weitergegeben.

In den folgenden Abschnitten werden diese Schichten im Detail betrachtet.

3 Systemaufrufe

Die Schnittstelle zwischen dem Betriebssystem und den Benutzerprogrammen wird durch eine Menge von Systemaufrufen gebildet, die das Betriebssystem zur Verfügung stellt.

Um die Daten lesen bzw. schreiben zu können, müssen die Dateien geöffnet und dann wieder geschlossen werden, was alles über Systemaufrufe geschieht.

Beispiel in Unix

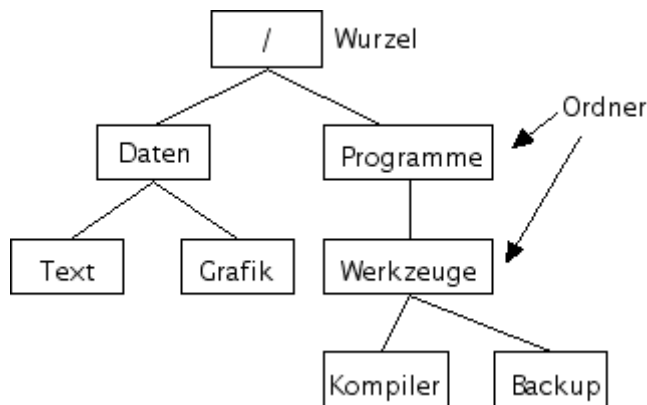
Wenn ein Prozess im Benutzermodus arbeitet und einen Systemaufruf nutzen will, wie etwa das Lesen einer Datei, dann muss er eine Unterbrechung oder eine spezielle System-Anweisung ausführen und dadurch die Kontrolle an das Betriebssystem übergeben. Danach wird der Systemaufruf ausgeführt und die Kontrolle an das Benutzerprogramm zurückgegeben.

4 Dateien

4.1 Dateisysteme

Dateien sind in einem Dateisystem organisiert. Es gibt verschiedene Dateisysteme, mit denen Betriebssysteme arbeiten können. Einige davon sind: FAT 32, NTFS, Ext, Der Zugriff auf die Dateien erfolgt über das Dateisystem.

Hierarchisches Dateisystem von Unix.



Das Dateisystem, das den UNIX Namensbaum verwaltet, in dem hierarchisch alle Dateien, Ordner und Geräte verzeichnet sind. Alle Einträge in diesen Namensbaum werden über die gleiche Schnittstelle angesprochen, so dass das Dateisystem erst einmal feststellen muss, welches Gerät bzw. welcher Gerätetreiber wirklich zuständig ist.

4.2 Transport der Daten

Dateien werden in Blöcke eingeteilt, die üblich 512 Byte groß sind. Die Belegung von Blöcken erfolgt fortlaufend, verkettet oder indiziert. Diese Blöcke werden zwischen der Festplatte und dem Hauptspeicher transportiert.

- Fortlaufende Belegung:

Die Daten werden in zusammenhängenden realen Blöcken gespeichert. Vorteile: schnelle Zugriffe. Nachteile: externe Fragmentierung.

- verkettete Belegung:

Die Blöcke einer Datei werden als verkettete Liste angelegt, wobei jeder Block die Nummer des Folgeblocks enthält. Vorteile: keine externe Fragmentierung. Nachteile: Overhead durch die Blockzeiger, Durchlaufen der Liste beim Zugriff in die Mitte.

- indexierte Belegung:

Der erste Block einer Datei ist Indexblock und enthält die Zeiger auf die Datenblöcke. Vorteile: schneller Zugriff, optimale Blockausnutzung. Nachteile: Overhead durch die Zeiger.

5 Festplattentreiber

Ein Treiber stellt ganz allgemein einen Vermittler zwischen zwei verschiedenen Schichten oder Sichtweisen eines Systems dar, also z.B. zwischen einem Gerät und der Anwendung:
Anwendung >> Treiber >> Gerät

Dabei wandelt der Treiber die relativ abstrakten, geräteunabhängigen Anweisungen der Anwendung in konkrete Befehle an das Gerät um und stellt das Ergebnis der Aktion der Anwendung wieder in geeigneter Form zur Verfügung.

Da die Festplatte ein blockorientiertes Gerät ist, ist der Festplattentreiber ein blockorientierter Treiber, der als ein Teil des I/O – Subsystems gilt. Nachdem der Treiber festgestellt hat, welchen Befehl er ausführen muss, schreibt er diesen in die Steuerregister des Geräts. Nachdem der Befehl abgesetzt wurde, prüft der Treiber, ob der Gerätecontroller diesen Befehl akzeptiert hat und bereit für den nächsten Befehl ist.

6 Festplattencontroller

Nachdem der Treiber die Kommandos in die Register des Controllers geschrieben hat, beginnt der Zugriff des Controllers auf die Festplatte. Die Aufgabe des Controllers beim Lesen der Festplatte ist es, den seriellen Bitstrom in Byte-Blöcke zu konvertieren und gegebenenfalls Fehlerkorrekturen durchzuführen. Der Block wird typischerweise in einem Puffer innerhalb des Controllers Bit für Bit aufgesammelt. Nachdem die Prüfsumme getestet worden ist, wird der Block als fehlerfrei angesehen und weiterkopiert.

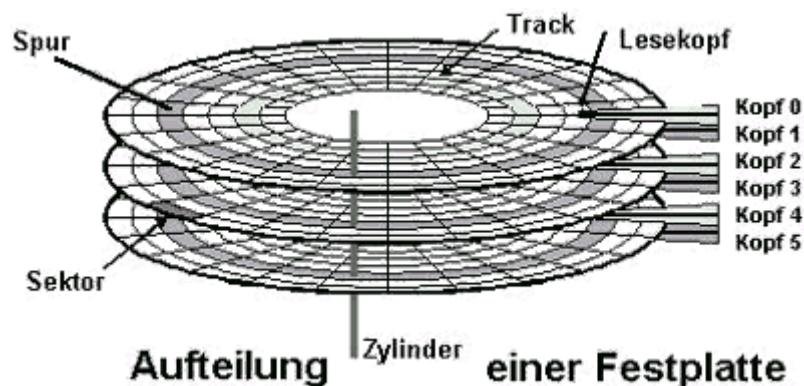
Ein-/Ausgabe können auf drei verschiedene Arten durchgeführt werden:

1. Zum einen gibt es die programmierte E/A (PIO), bei der die CPU eine E/A eines Zeichens oder Wortes ausführt und in einer Schleife wartet, bis das nächste Zeichen oder Wort empfangen oder gesendet werden kann.
2. Bei der unterbrechungsgesteuerten E/A stößt die CPU E/A eines Zeichens oder Wortes an und arbeitet weiter. Beim Beenden der E/A wird eine Unterbrechung ausgelöst, die das Ende der Aktion signalisiert.
3. Die dritte Art ist die DMA (Direct Memory Access) Übertragung, bei der ein eigener Chip den kompletten Transfer eines Datenblocks übernimmt und erst dann eine Unterbrechung erzeugt, wenn der gesamte Block übertragen wurde.

Es gibt verschiedene Scheduling-Verfahren, wie auf Blöcke zugegriffen werden muss:

- FCFS (First come, first served): die Kommandos werden so ausgeführt, wie sie ankommen,
- SSTF (Shortest seek time first): am nächsten liegende Kommandos werden zunächst ausgeführt,
- SCAN: läuft die Platte durch und bearbeitet alle Anfragen auf dem Weg,
- C-SCAN: gleich wie SCAN, bearbeitet aber keine Anfragen auf dem Rückweg,
- LOOK, C-LOOK: wie SCAN bzw. C-SCAN, bewegt den Kopf aber bis zur letzten Anfrage und dann bis zur ersten.

7 Festplatte



Festplatten bestehen aus einer oder mehreren Metallplatten, die beidseitig mit hochfeinpolierten Metalloxiden beschichtet sind. Ein Kamm von beweglichen Schreib/Leseköpfen greift seitlich in den rotierenden Plattenstapel hinein. Jede Platte ist in Tracks geteilt. Übereinander liegende Tracks (auf verschiedenen Platten) bilden Zylinder. Die Tracks selbst sind in Sektoren, in denen die Daten gespeichert werden, geteilt. Beim Zugriff auf einen bestimmten Block auf der Festplatte bekommt der Controller die Adresse dieses Blocks, bestehend aus Zylinder-, Track- und Sektornummer.

8 Beispiel der I/O Programmierung

```
1.  int main(int argc, char *argv[]){
2.      char c;
3.      FILE *fp;
4.      char *point = argv[1];
5.      fp = fopen(point, "r");
6.      while((c = getc(fp)) != EOF){
7.          putchar(c, stdout);
8.      }
9.      fclose(fp);
10.     return(0);
11. }
```

In diesem Beispiel wird eine Datei eingelesen und auf die Standardausgabe ausgegeben. In der Zeile 5 wird die Datei, deren Name über die Kommandozeile eingegeben wurde, geöffnet. In den Zeilen 6-8 wird die Datei zeichenweise eingelesen und gleich ausgegeben.

9 Quellen

Andrew S. Tannenbaum, Moderne Betriebssysteme 2., überarbeitete Auflage 2002
Abraham Silberschatz, Operating System Concepts, sixth Edition 2002