

Dateien

- ▶ Prozesse erzeugen/löschen etc. Dateien und öffnen sie zum Lesen/Schreiben
- ▶ Dateisystem verwaltet Plattenplatz und teilt ihn unter den Dateien auf
- ▶ Dateisystem pflegt Liste der verfügbaren Dateien und bietet Suchfunktionen

Funktionen des Dateisystems

... aus Sicht des Benutzers:

- ▶ create: Datei erzeugen
- ▶ open: Datei zum Lesen/Schreiben öffnen
- ▶ seek: In geöffneter Datei an Position x springen
- ▶ read: An aktueller Dateiposition Daten lesen
- ▶ write: ... Daten schreiben
- ▶ rename, delete, ...
- ▶ Verzeichnisverwaltung

Funktionen des Dateisystems

... aus Sicht des Betriebssystems:

- ▶ Freien Plattenplatz verwalten (create, delete, Datei vergrößern: write)
- ▶ Zugriff auf vorhandene Dateien erlauben (open, seek, read, write)
- ▶ Verzeichnisse verwalten (rename, move, ...)
- ▶ Zugriff für Prozesse transparent machen, u. a. unabhängig von tatsächlicher Formatierung (etwa: FAT, NTFS, Linux Ext3)

Beispiel: Dateisystem von CP/M (1/2)

- ▶ Flache Struktur (keine Verzeichnisse)
- ▶ Dateien: 8+3-Konvention (Name + Extension)
- ▶ Extension beschreibt Dateityp
 - ▶ Programme: *.com
 - ▶ Textdateien: *.txt etc.
- ▶ Etwas Struktur durch „User“ (16 mögliche Dateibesitzer)
- ▶ „Laufwerksbuchstaben“ für verschiedene Datenträger

Beispiel: Dateisystem von CP/M (2/2)

- ▶ Am Anfang eines Datenträgers: Inhaltsverzeichnis (für 64 Dateien)
- ▶ Für jeden Eintrag 32 Byte reservieren

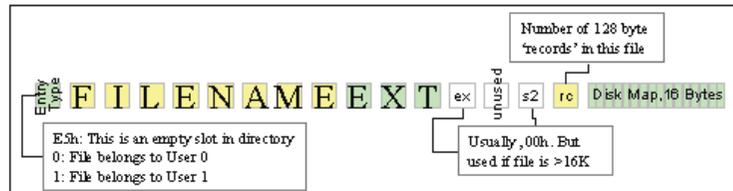


Foto: <http://www.dcast.vbox.co.uk/cpm.html> (modifiziert)

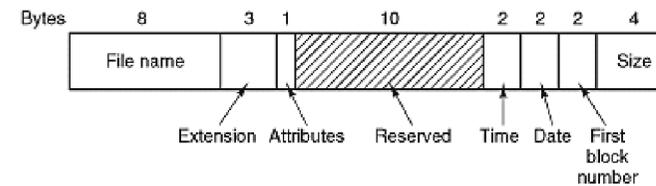
- ▶ obere Bits der Erweiterungen enthalten drei Attribute (read-only, hidden, archived)
- ▶ Dateigröße >16 KByte → mehrere Einträge

Beispiel: Dateisystem von MS-DOS (1/2)

- ▶ Hierarchische Struktur (Unterverzeichnisse)
- ▶ Dateien: 8+3-Konvention (wie CP/M)
- ▶ Extensions ähnlich CP/M (Programme: *.com, *.exe, ...)
- ▶ „Laufwerksbuchstaben“ (wie CP/M)
- ▶ keine Liste der Dateiblöcke, sondern Index in File Allocation Table (FAT, mehr dazu später)

Beispiel: Dateisystem von MS-DOS (2/2)

Jeder Verzeichniseintrag ist 32 Byte groß:



Byte	Bedeutung
0-10	Dateiname (8 Byte) und Dateierweiterung (3 Byte)
11	Datei-Attribute: Bit 0: read-only (R), Bit 1: hidden(H), Bit 2: system file (S), Bit 3: volume label, Bit 4: Verzeichnis, Bit 5: archiviert (A), Bits 6-7: ungenutzt, attrib +RHS c:\io.sys
12-21	reserviert
22-23	Zeit: Stunde (5 Bit), Minute (6 Bit), Doppel-Sekunde (5 Bit)
24-25	Datum: Jahr seit 1980 (7 Bit), Monat (4 Bit), Tag (5 Bit)
26-27	Erster Block (0 = leere Datei)
28-31	Dateigröße in Byte (32 Bit: theoretisch bis zu 4 GByte)

Bild: <http://www.phptr.com/articles/article.asp?p=25878>

Probleme klassischer Dateisysteme

BS implementiert ein konkretes Dateisystem

- ▶ Systemfunktionen (öffnen, lesen, schreiben etc.) auf dieses spezielle Dateisystem zugeschnitten
- ▶ Support für fremde / neue Dateisysteme schwierig; oft ganz unmöglich
- ▶ Software (von Drittanbietern) für Zugriff auf fremde Dateisysteme schlecht integriert

Lösung: Virtuelles Dateisystem

Hauptspeicher

- ▶ Prozesse benötigen Hauptspeicher für ihren Programmcode, die Daten und den Stack
- ▶ Sie sprechen den Speicher auf zwei Arten an:
 - ▶ Sprungbefehle – zu einer anderen Anweisung im Programmcode
 - ▶ Ladebefehle – Daten aus dem Speicher in ein Register schreiben (oder umgekehrt)

Funktionen der Speicherverwaltung

... aus Benutzersicht:

- ▶ Sprung zu anderer Code-Adresse (`jmp`, `call`)
- ▶ Auslesen einer Speicheradresse
- ▶ Schreiben einer Speicheradresse
- ▶ evtl.: dynamisch mehr Speicher (für Daten) anfordern

Funktionen der Speicherverwaltung

... aus Sicht des Betriebssystems:

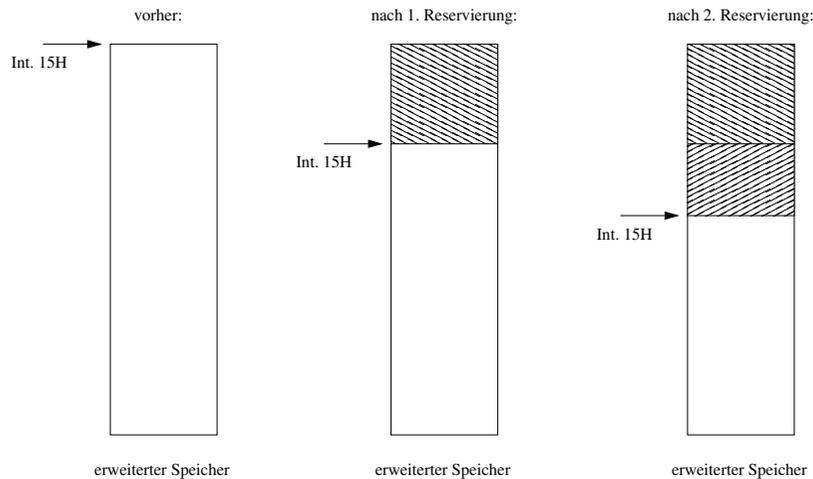
- ▶ Prozess beim Start mit Speicher versorgen
- ▶ evtl.: dynamisch die zugeteilte Speichermenge anpassen
- ▶ Sicher stellen, dass Prozesse exklusiv auf ihren Speicher zugreifen
- ▶ evtl.: Verwaltung von „Swap Space“
- ▶ Verwaltung des freien Speichers (bei allen Änderungen, inkl. Prozess-Start und -Ende)

Beispiel: Speicher unter MS-DOS (1/2)

Extended Memory und Interrupt 15H

- ▶ Interrupt 15H meldet den verfügbaren erweiterten Speicher (Extended Memory)
- ▶ Programme, die erweiterten Speicher nutzen wollen, ...
 - ▶ zweigen einen Teil des Speichers für sich ab,
 - ▶ „verbiegen“ Interrupt 15H auf eine eigene Funktion, die nun weniger verfügbaren Speicher meldet
- ▶ Unter DOS nutzten u. a. TSR-Programme (Terminate and Stay Resident) diese Funktion

Beispiel: Speicher unter MS-DOS (2/2)



Einleitung

- ▶ Zusammenhängende Speicherzuordnung, engl.: contiguous (memory) allocation
- ▶ Prozesse: erhalten ein zus.hgd. Stück Speicher
- ▶ Dateien: verwenden einen zus.hgd. Bereich der Platte

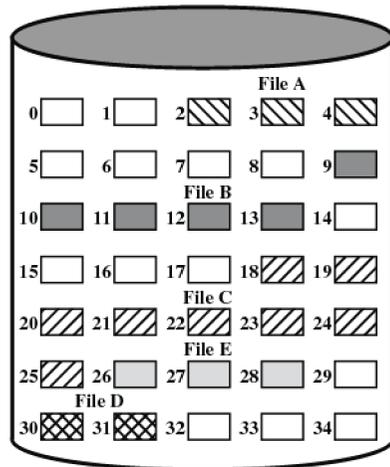
Methoden der Speicher-/Plattenplatzverwaltung

- ▶ Zusammenhängende Speicherzuordnung
 - ▶ Dateien / Prozess-Speicher „am Stück“
 - ▶ sehr übersichtliche Verfahren möglich
 - ▶ schlechte Ressourcennutzung, Fragmentierung
- ▶ Nicht-zusammenhängende Speicherzuordnung
 - ▶ Dateien / Prozess-Speicher „gestückelt“
 - ▶ komplexere Verfahren
 - ▶ bessere Ressourcennutzung, evtl. immer noch Fragmentierung

... für Dateien (1/3)

- ▶ Festplatte in Blöcke gleicher Größe unterteilen, z. B. 1 GByte Platte, Blockgröße 1 KByte → 1 M Blöcke
- ▶ Beim Erzeugen einer Datei wird eine bestimmte Menge Blöcke alloziert
- ▶ Ein einziger Eintrag in der Dateizuordnungstabelle: Anfangsblock und Länge der Datei

... für Dateien (2/3)



File Allocation Table

File Name	Start Block	Length
File A	2	3
File B	9	5
File C	18	8
File D	30	2
File E	26	3

Bild: Stallings, Kapitel 12

Fragmentierung

- ▶ Es bleiben Lücken zwischen zugeordneten Bereichen
- ▶ Lücken sind evtl. zu klein, um noch benutzbar zu sein
- ▶ → **externe Fragmentierung**
(extern, weil unbenutzbare Bereiche *außerhalb* der reservierten Bereiche liegen)
- ▶ lässt sich beheben mit Defragmentierung

... für Dateien (3/3)

Vorteile:

- ▶ Schneller Zugriff (sequentielles Lesen/Schreiben)
- ▶ übersichtliche Verwaltung (für jede Datei nur Anfang, Länge)
- ▶ freie Bereiche leicht erkennbar

Probleme:

- ▶ Was tun, wenn Datei größer wird?
- ▶ Was tun, wenn Datei kleiner / gelöscht wird?

... für Speicher

Arbeitsauftrag:

Übertragen Sie die Konzepte auf die Speicherverwaltung

- ▶ Zuordnung Prozess / Speicherbereich entspricht ... ?
- ▶ Was entspricht dem Löschen einer Datei, dem Vergrößern einer Datei?
- ▶ Gibt es hier ein Fragmentierungsproblem?