



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

Einführung in Dateisysteme

Proseminar Speicher- und Dateisysteme

Veranstaltung:
Proseminar Speicher- und Dateisysteme

Veranstalter: Dr. Michael Kuhn
Einrichtung: Universität Hamburg
Fachbereich: Informatik
Abgabesemester: Sommersemester 2015
Abgabedatum: 27.09.2015

Autor:
Kelbert, Raphael

E-Mail: Raphael.Kelbert@informatik.uni-hamburg.de
Studiengang: Wirtschaftsinformatik
Matr.-Nr.: 6525044
Fachsemester: 4

1 Inhaltsverzeichnis

2	Motivation.....	1
3	Allgemeines.....	1
3.1	Was ist ein Dateisystem?.....	1
3.2	Wofür benötigen wir Dateisysteme?	1
3.3	Historisches	2
4	Grundlegende Konzepte und Begrifflichkeiten.....	2
4.1	Massendatenspeicher	2
4.2	Dateien	3
4.2.1	Metadaten zu Dateien	4
4.2.2	Speicherung von Clusteradressen	4
4.3	Verzeichnisse	5
4.4	Speicherung von Daten auf der Festplatte (beispielhaftes Medium)	6
4.4.1	Master Boot Record (MBR).....	7
4.4.2	GUID Partition Table (GPT)	7
4.4.3	defragmentierte vs. fragmentierte Festplatten	8
5	Strukturelle Konzepte von Dateisystemen	9
5.1	Lineare Dateisysteme.....	9
5.2	Hierarchische Dateisysteme	9
5.2.1	Kommunikationsablauf eines Dateiaufrufs.....	10
5.3	Virtuelle Dateisysteme	10
6	Sicherheitsaspekte.....	10
6.1	Zugriffsschutz	10
6.2	Datensicherheit.....	11
6.2.1	Multitasking.....	11
6.2.2	Stromausfälle.....	11
7	Fazit.....	12
8	Abbildungsverzeichnis	13
9	Literaturverzeichnis	14

2 Motivation

In der heutigen, sehr stark technisierten Welt sind Dateisysteme nicht nur fast überall vertreten, sondern auch grundlegende Voraussetzung für die Funktionsfähigkeit der von uns verwendeten technischen Geräte.

In dieser Ausarbeitung gebe ich einen kurzen Überblick darüber, was ein Dateisystem ist, wofür es verwendet wird und welche grundlegenden Konzepte im Zusammenhang mit Dateisystemen bekannt sein sollten. Ferner zeige ich strukturell unterschiedliche Konzeptionen von Dateisystemen auf und gehe gezielt auf die Sicherheitsanforderungen gegenüber Dateisystemen ein.

3 Allgemeines

3.1 Was ist ein Dateisystem?

Die Internetplattform IT-Wissen definiert ein Dateisystem wie folgt:

„Das Dateisystem ist Bestandteil des Betriebssystems und bildet die Schnittstelle zwischen diesem und den Laufwerken. Es legt fest, wie der Computer Dateien auf den Datenträgern benennt, speichert, organisiert und verwaltet. Ein Dateisystem besteht aus Dateien, Verzeichnissen und Adressen, über die die Dateien lokalisiert werden.“ [IT-Wissen, 2015]

Grob gesagt lässt sich also das Dateisystem als eine Art „Vermittler“ für Dateien von technischen Geräten sehen, welcher dafür zuständig ist, die richtigen Daten zur richtigen Zeit zu speichern oder zu lesen. Die Art und Weise, wie dieser Vermittler die Daten für sich selber organisiert und verwaltet ist dabei für ein Gerät irrelevant, lediglich die Korrektheit der Daten wird dabei vorausgesetzt. Verschiedene Dateisysteme unterscheiden sich daher maßgeblich durch die unterschiedlichen Arten Daten zu organisieren und zu verwalten.

3.2 Wofür benötigen wir Dateisysteme?

Angenommen, man hätte für ein Speichermedium kein Dateisystem: So hätte man zwar ein Medium zum Speichern von Daten, allerdings keinen oben beschriebenen Vermittler mehr, der die Organisation und Verwaltung übernimmt. Diese Aufgabe müsste dann eigenständig von Benutzer erledigt werden. Das heißt der Nutzer müsste sich für jede Datei merken in welchen Speicherblöcken eine Datei gespeichert ist und welche Blöcke zu welcher Datei gehören oder welche noch frei sind. Bei Speichermedien im Terabyte-Bereich, wie sie heute üblich sind, kann es schnell vorkommen, dass es mehrerer Milliarden verschiedene Speicherblöcke gibt. So wird die Zuordnung schnell zu einem unmöglichen Unterfangen für den Nutzer.

Dateisysteme nehmen dem Nutzer diese Organisation und Verwaltung ab und bilden damit die Schnittstelle zwischen dem Betriebssystem, welches auf Aktionen des Nutzers reagiert und der Firmware der unterschiedlichen Speichermedien. Dateisysteme sorgen durch Organisation und Strukturierung dafür, dass Dateien einen einfachen und leicht merkbaren Namen erhalten. Sie werden also benötigt für eine sinnvolle und automatisierte Verwaltung der Daten.

3.3 Historisches

Dateisysteme sind, obgleich vielfach anders vermutet, ein bereits relativ altes Konstrukt. So gab es bereits im 18. Jahrhundert sehr einfache Dateisysteme für Lochkarten und Lochstreifen, welche bei der Steuerung von automatisierten Webstühlen zum Einsatz kamen. Sie bilden genauso wie die um 1930 entwickelten Magnetbandspeicher lineare Dateisysteme. Erst durch immer komplexer werdenden Speichermedien und die Notwendigkeit der Massendatenspeicherung bei schnellem Datenzugriff mussten neue Dateisysteme entwickelt werden. Dies geschah im Zuge der Einführung von Trommelspeichern um 1950 und später von Festplatten. Im Gegensatz zu älteren Speichermedien, wie Lochkarten, Lochstreifen und Magnetbändern war nun nicht mehr nur das sequentielle Lesen der Medien möglich, sondern auch ein direkter Zugriff auf alle Speicherbereiche. Sie stellten daher neue Anforderungen an die zugehörigen Dateisysteme. Zwischenzeitlich wurde das Konzept der Dateisysteme soweit abstrahiert, dass auch Dateizugriffe im Netz oder auf anderen Geräten mit Hilfe von Dateisystemen durchgeführt werden können. Anwendungsprogramme können zu diesem Zweck werden diese anderen Geräte virtuell als Datei verwaltet. Anwendungsprogramme können so über einheitlich definierte Schnittstellen auf unterschiedliche Datenquellen zugreifen. [vgl. *Dateisystem, Wikipedia, 2015*]

4 Grundlegende Konzepte und Begrifflichkeiten

In diesem Abschnitt wird das grundlegende Konzept eines Massendatenspeichers erläutert und darauf aufbauend anhand von Beispielen aufgezeigt, wie die Organisation und Verwaltung der Daten in der Theorie funktioniert. Außerdem werden die dazu benötigten Grundbegriffe eingeführt.

4.1 Massendatenspeicher

Massendatenspeicher, wie Festplatten, DVDs, USB-Sticks, usw., sind rein technisch betrachtet nur physikalische Medien, welche Speicherplatz zur Verfügung stellen. Im Normalfall weisen diese eine Blockstruktur auf, heißt sie präsentieren sich gegenüber dem Betriebssystem vergleichbar einer großen linearen Fläche, auf der viele Blöcke nummeriert sind. Abbildung 1 (folgend) verdeutlicht die Struktur eines solchen Massendatenspeichers, die für die Nutzung mit einem Dateisystem ist.



Abbildung 1 - Struktur eines Massendatenspeichers

Der obere weiße Balken stellt den physikalischen Platz auf dem Medium dar, also den real existenten Speicherkörper. Dieser ist typischerweise in Blöcke mit einer Größe von 512 oder 4096 Bytes aufgeteilt (bei optischen Medien beträgt diese Blockgröße standardmäßig 2048 Bytes). Um die Zugriffsgeschwindigkeit zu erhöhen und den Aufwand des Dateisystems zu reduzieren, werden allerdings nicht diese einzelnen Blöcke adressiert, sondern immer mehrere Blöcke zu einem Cluster zusammengefasst. Heutzutage ist eine Clustergröße von 8 oder mehr Blöcken üblich wobei die Clustergröße im Allgemeinen einer Zweierpotenz entspricht.

Cluster sind die nach außen sichtbare Struktur eines Massendatenspeichers für Dateisysteme. So können einzelne Bereiche auf dem Medium adressiert werden. [vgl. *Massendatenspeicher*, Wikipedia, 2015]

4.2 Dateien

Die Internetplattform Wikipedia definiert den Begriff „Datei“ wie folgt:

„Eine Datei (englisch file) ist ein Bestand meist inhaltlich zusammengehöriger Daten, der auf einem Datenträger oder Speichermedium gespeichert ist.“ [Datei, Wikipedia, 2015]

Setzt man diese Definition in Zusammenhang zum oben erklärten Prinzip des Massendatenspeichers, so kommt man relativ schnell zu folgender Interpretation:

Eine Datei ist eine Menge von Clustern, die zusammengehörige Daten enthalten und nur als Ganzes interpretiert einen Sinn ergeben.

Es stellt sich aber die Frage, ob diese Interpretation ausreichend zur Definition des Datei-Begriffs ist, da die die zugehörigen Cluster einer Datei zwar zufälligerweise alle geordnet hintereinander auf dem Medium liegen können, sie können aber auch willkürlich an mehreren Orten auf dem Medium verstreut liegen. Die zweite Möglichkeit besitzt dabei eine höhere Wahrscheinlichkeit. Außerdem fehlt eine Angabe darüber, wie viele Cluster eine spezifische Datei beinhalten. Um zu ermitteln, welche Cluster zu welcher Datei gehören oder zu welcher Datei welche Cluster gehört ist also noch ein weiterer Bestandteil nötig: Die sogenannten Metadaten.

4.2.1 Metadaten zu Dateien

Die Metadaten werden auf einem Massendatenspeicher in eine dafür vorgesehene Tabelle eingetragen. Sie speichern Beschreibungen zu den einzelnen Dateien in einem eindeutig zuordbaren Datensatz je Datei. Minimale Angaben bei den Metadaten sind die eindeutige Datensatznummer (ID), sowie die Anzahl und die Orte der zur Datei gehörenden Cluster. Es gibt vier verschiedene Verfahren, um die Orte der zugehörigen Cluster zu hinterlegen. Diese werden im folgenden Abschnitt näher erläutert. Ferner können weitere beschreibende Eigenschaften gespeichert werden. Diese sind aber keinesfalls zwingend erforderlich. Es könnten zum Beispiel auch der Dateiname, der Dateityp, Zugriffsrechte, das Erstellungsdatum, das Datum der letzten Änderung und vieles mehr abgespeichert werden.

Erweitert man also die oben formulierte Interpretation, lässt sich eine, ausreichend detaillierte Definition des Begriffs Datei geben:

Eine Datei ist eine, über Metadaten eindeutig als verbunden definierte Menge von Clustern, die zusammengehörige Daten enthalten und nur als Ganzes interpretiert einen Sinn ergeben.

4.2.2 Speicherung von Clusteradressen

Wie bereits erwähnt, gibt es vier verschiedene Verfahren, die zur Speicherung der Clusteradressen angewendet werden.

1. Angabe von Startcluster und Länge:

Das einfachste Verfahren für die Speicherung von Clusteradressen ist es lediglich den Startclusters der Datei sowie ihre Länge zu speichern. Dazu müssen allerdings alle zur Datei gehörenden Cluster sequentiell hintereinander gespeichert werden. Startcluster 1234 und Länge 7, sagt zum Beispiel aus, dass die Datei im Cluster 1234 beginnt und sieben Cluster lang ist, also im Cluster 1241 letzten Informationen der Datei liegen. Nachteil dieses Verfahrens ist, dass alle Cluster sequentiell hintereinander liegen müssen. Wenn beispielsweise Cluster 1242 bereits belegt ist und die Datei vergrößert werden soll geht dies nicht, da nicht einfach die Längenangabe um die Anzahl der zusätzlichen Cluster erhöht werden kann. Die komplette Datei muss dann an einen neuen Ort geschrieben werden. Die Abspeicherung der Änderungen wird deutlich verlangsamt. Außerdem kann es passieren, dass viel freier, aber leider unbenutzbarer Speicherplatz existiert, wenn dieser so verteilt ist, dass für eine große Datei keine ausreichende Menge aufeinanderfolgender, freier Cluster gefunden werden kann.

2. Angabe vom Startcluster und Folgecluster am Ende jeden Clusters:

Beim zweiten Verfahren werden nur die Speicheradresse des Startclusters und die Gesamtclusterzahl einer Datei in den Metadaten gespeichert, erst am Ende jeden Clusters steht dann die Speicheradresse des darauf folgenden Clusters. So kann

freier Speicherplatz beliebig genutzt werden und das Hauptproblem der ersten Variante ist behoben. Aber auch dieses Verfahren ist nicht ohne Nachteile:

Durch diese Art der Speicherung müssen bei Zugriff auf das Dateiende immer erst alle vorhergehenden Cluster gelesen werden, damit der gewünschte Cluster überhaupt definiert werden kann. Zudem hätte die Beschädigung eines Clusters zur Folge, dass die gesamte Datei möglicherweise nicht mehr gelesen werden kann.

3. alle Clusteradressen der Datei einzeln speichern:

Die bei den ersten beiden Varianten auftretenden Nachteile können durch Anwendung des dritten Verfahrens vermieden werden: Bei diesem werden die Speicheradressen aller zur Datei gehörenden Cluster einzeln in den Metadaten hinterlegt. So kann schnell und unabhängig auf jeden Cluster zugegriffen werden. Allerdings zeichnet sich auch bei diesem Verfahren ein bedeutendes Problem ab: Die große Menge an Metadaten, die gespeichert werden müssen. Dies ist durch den Eintrag in den Metadaten je Cluster bedingt.

4. Speicherung in Extents, ein Mix aus 1 und 3:

Das heute am häufigsten verbreitete Verfahren ist die Speicherung in Extents. Es handelt sich um eine Kombination aus dem Verfahren der Speicherung eines Startclusters mit Clusteranzahl (sequenziell) und einer Speicherung jeder einzelnen Clusteradresse. Ein Extent stellt dabei einen Dateiabschnitt dar, der mehrere sequentiell hintereinander beschriebene Cluster einer Datei umfasst. Für jedes Extent einer Datei werden der Startcluster und die Länge des Dateiabschnitts gespeichert. So muss nicht jeder Cluster einzeln gespeichert werden, es kann aber dennoch der Speicherplatz optimal belegt werden und relativ schnell auf alle Dateibereiche zugegriffen werden. Ein Metadaten-Eintrag einer Datei könnte dann zum Beispiel folgendermaßen aussehen:

Extent 1: Startcluster: 4895; Länge: 11

Extent 2: Startcluster: 9875 Länge: 3

Extent 3: Startcluster: 9998 Länge: 7

[vgl. Dateisystem, Wikipedia, 2015]

4.3 Verzeichnisse

Kernaufgabe von Dateisystemen ist und bleibt aber die automatisierte Organisation von Dateien. Da der Dateiname für den Nutzer zur Identifikation einer Datei genutzt wird, jedoch nicht zwangsläufig zu den Metadaten einer Datei gehört, braucht das Dateisystem zur sinnvollen Nutzung einen Ort, an dem dieser Dateiname abgelegt wird. Dies geschieht in sogenannten Verzeichnissen. Da es sich hierbei auch um Speicherflächen handelt, werden als Verzeichnisse von den meisten Dateisystemen speziell gekennzeichnete Dateien verwendet. In einem Verzeichnis werden immer paarweise der Dateiname und die Datensatznummer (ID) oder eine Referenz auf den zugehörigen Metadatensatz der Datei gespeichert. So können über Verzeichnisse mittels des Dateinamen die abgelegten

Daten genau identifiziert werden. Nachfolgend wird ein Beispiel dargestellt, wie die in einem beliebigen Verzeichnis abgelegte Datei „Hallowelt.exe“ gelesen wird:

In dem Verzeichnis steht der Eintrag „Hallowelt.exe“. Dieser hält eine Referenz auf den zugehörigen Datensatz in der Metadaten-tabelle. In diesem Datensatz steht ganz zu Beginn die Gesamtzahl der zur Datei gehörigen Cluster [12]. Im Folgenden die Extents mit Startcluster und Länge, hier z.B.: [1254, 4 und 4895, 8] die Datei besteht also aus zwei Extents mit einer Länge von 4 und 8 Clustern. Außerdem finden sich in der Metadaten-tabelle noch weitere beschreibende Informationen der Datei. Über die Extents kann auf der Festplatte auf einen Cluster und die darin enthaltenen Daten zugegriffen werden. [vgl. Verzeichnisse, Wikipedia, 2015]

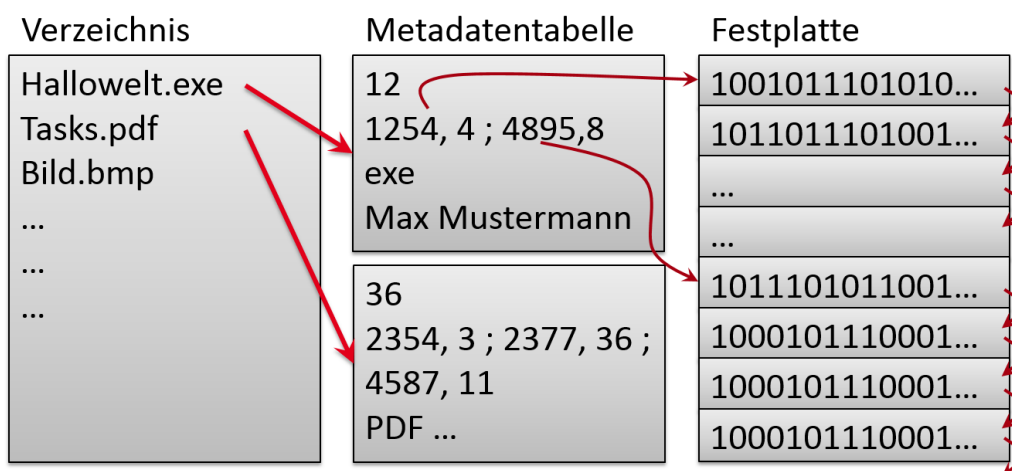


Abbildung 2 - Ablauf bei Dateiaufruf

4.4 Speicherung von Daten auf der Festplatte (beispielhaftes Medium)

Die Festplatte ist letztlich der physikalische Ort, an dem die Dateien, Verzeichnisse, Metadaten und noch einige weitere Bereiche gespeichert werden. Neben den bereits eingeführten Elementen kommen noch die Block Availability Map (BAM), sowie ein Bootblock. Die BAM dient der Buchführung darüber, welche Blöcke oder Cluster eines Mediums noch frei, oder bereits belegt sind. Dazu wird für jeden Speicherblock ein Bit angelegt, welches mittels 0-1-Codierung anzeigt, ob der Block frei oder belegt ist (0 = frei, 1 = belegt). Im Grunde dient die BAM der Effizienz der Verwaltung, da die Informationen nur redundant gehalten werden. Sollten Informationen verloren gehen, lässt sich die BAM mit Hilfe eines Scans des Speichermediums neu erzeugen.

Der Bootblock oder auch Bootsektor ist normalerweise der erste Sektor eines startfähigen Mediums und enthält Informationen oder ein Programm, welches zum Starten dieses Mediums benötigt wird. Besonders wichtig ist die im Bootsektor enthaltene Partitionstabelle. Sie wird von den gängigen Betriebssystemen verwendet, um einen Zugriff auf die einzelnen Partitionen zu erhalten. Heute lassen sich im Wesentlichen die folgenden beiden Varianten für den Aufbau des Bootblocks unterscheiden.

4.4.1 Master Boot Record (MBR)

Der Master Boot Record (MBR) ist ein Programm mit Partitionstabelle zum Starten von BIOS-basierten Computern. Angeordnet wird der MBR strukturell im ersten Sektor von partitionierbaren Speichermedien, wie zum Beispiel Festplatten und besteht aus Bootloader und maximal vier Partitionseinträgen. Ursprünglich wurde das Konzept aus Bootloader und Partitionstabellen 1983 mit MS-DOS/PC-DOS 2.0 auf dem IBM-PC XT eingeführt. Bis ca. 2000 galt der MBR als De-facto-Standard für Partitionstabellen verschiedenartiger Speichermedien, wie z. B. USB-Sticks, Speicherkarten oder externe Festplatten, etabliert. Aufgrund der immer größeren Speichermedien (häufig > 2TB) und der Nachfrage nach mehr als vier Partitionen, wird immer häufiger der seit 2010 eingeführte GUID Partition Table (GPT) angewendet. [vgl. MBR, Wikipedia, 2015]

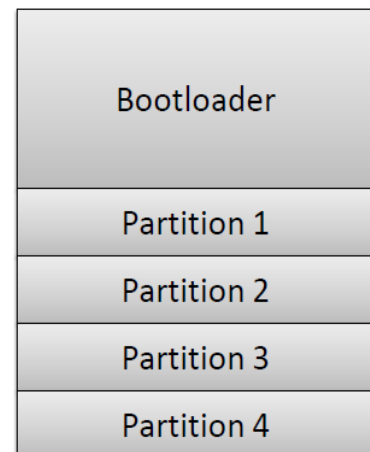


Abbildung 3 – schematisch vereinfachte Darstellung eines MBR

4.4.2 GUID Partition Table (GPT)

GPT ist der „neue“ Standard für das Format von Partitionstabellen auf Datenträgern und ist Teil des UEFI-Standard, der etwa ab dem Jahr 2000 ausgehend von Großrechnern und seit ca. 2010 auch vermehrt auf Heimcomputern das BIOS ersetzt. GPT ist sozusagen der Nachfolger des MBR und beinhaltet einen MBR Bootloader, den Primary-GPT-Haeder und im Anschluss, wie beim MBR vorerst vier Partitionseinträge. Diese und der Bootloader zusammen ergeben einen klassischen MBR, wodurch der Standard auch abwärtskompatibel ist. Im Folgenden kommen noch mögliche Plätze für bis zu insgesamt 128 Partitionen hinzu. Am Ende des Mediums werden redundant ein zweiter Secondary-GPT-Haeder sowie noch einmal alle 128 Partitionseinträge gehalten. Somit ist eine Wiederherstellung im Fehlerfall möglich. [vgl. GPT, Wikipedia, 2015]

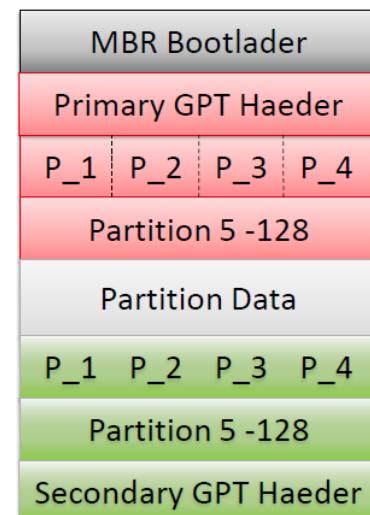


Abbildung 4 – schematisch vereinfachte Darstellung eines GPT

4.4.3 defragmentierte vs. fragmentierte Festplatten

Nachfolgende Grafiken zeigen den Unterschied zwischen einer defragmentierten und einer fragmentierten Partition:

fragmentierte Partition:



defragmentierte Partition:



(1) Bootblock; (2) BAM; (3) Metadaten; (4) Dateien und Verzeichnisse

Abbildung 5 - vereinfachte Darstellung einer Partition im fragmentierten und defragmentierten Zustand

Zu erkennen ist, dass der Bootblock und die Block Availability Map vorhanden sind. Sie liegen jeweils an derselben Stelle (vorne an) und weisen die gleiche Größe auf. Dies ist unabhängig davon, ob eine Festplatte fragmentiert oder defragmentiert ist. Die beiden Zustände unterscheiden sich dadurch, wie die Daten und Metadaten strukturiert sind.

Wird auf einer Festplatte ein Datensatz abgespeichert, so wird eine Metadaten-tabelle erstellt, die Metadaten eingetragen und der Datensatz abgelegt. Werden weitere Datensätze hinzugefügt, wird geschaut, ob noch ausreichend Platz in einer bereits angelegten Metadaten-tabelle existiert um die neuen Metadaten zu speichern. Ist dies der Fall, so wird der Metadaten-satz gespeichert und die Daten abgelegt. Ist nicht mehr ausreichend Platz vorhanden, wird eine neue Metadaten-tabelle angelegt und die Daten werden dort abgelegt. Dabei kommt es dazu, dass im Laufe der Zeit immer mehr kleinere Metadaten-tabellen auf der Festplatte angelegt werden und sich, bildlich gesprochen, die Festplatte in viele kleine Elemente zerstückelt (siehe Abbildung 5).

Durch das Defragmentieren einer Festplatte kann dieses Durcheinander „aufgeräumt“ werden und die Metadaten-tabellen werden in einer Tabelle vereint und Datensätze so zusammengelegt, dass zusammengehörige Daten möglichst hintereinander abgelegt sind. Ins besondere für ältere Dateisysteme, die die Clusteradressen nach Angabe von Startcluster und Länge abspeichern war ein regelmäßiges Defragmentieren äußerst wichtig, da so Speicherlücken reduziert werden konnten. Dennoch gilt auch für die anderen Varianten, dass eine „aufgeräumte“ Festplatte übersichtlicher ist und das Dateisystem auf dieser schneller arbeiten kann. [vgl. Fragmentierung, Wikipedia, 2015]

5 Strukturelle Konzepte von Dateisystemen

In diesem Kapitel werden die drei typischen Konzepte von linearen, hierarchischen und virtuellen Dateisystemen erläutert und Gemeinsamkeiten, sowie Unterschiede aufgezeigt.

5.1 Lineare Dateisysteme

Für normale Heimcomputer spielen lineare Dateisysteme de facto keine Rolle mehr. Früher wurden sie häufig für Lochkarten und auch Lochbänder eingesetzt und sind heute fast ausnahmslos noch in der professionellen Datensicherung zu finden, wo häufig noch mit Magnetbändern gearbeitet wird. Ein lineares Dateisystem unterscheidet im Gegensatz zur bisher beschriebenen Theorie nicht zwischen verschiedenen Verzeichnissen. Die Dateien liegen in einem einzigen Verzeichnis, welches direkt auf die Dateien verweist. Das Medium kann nur sequenziell gelesen und geschrieben werden. Durch die zunehmende Anzahl an Dateien und die dadurch immer länger dauernden Zugriffszeiten war es unvermeidlich ein neues strukturelles Konzept zur Datenorganisation zu entwickeln – die hierarchischen Dateisysteme.

5.2 Hierarchische Dateisysteme

Hierarchische Dateisysteme sind die heute üblichen, jedem bekannten Dateisysteme. Die wichtige Eigenschaft hierarchischer Dateisysteme ist die Möglichkeit Unterordner zu erstellen und so eine für Nutzer sinnvolle Dateistruktur möglich zu machen. Folgende Grafik stellt beispielhafte Verzeichnisbäume von hierarchischen Dateisystemen unter verschiedenen Betriebssystemen dar.

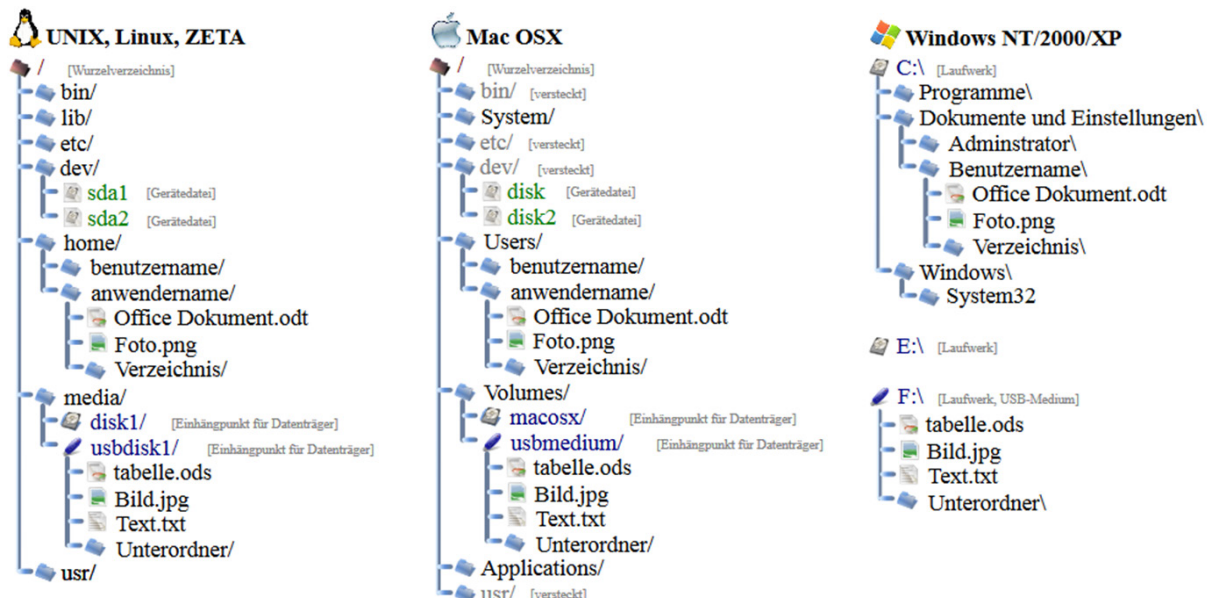


Abbildung 6 - Verzeichnisstruktur verschiedener Dateisysteme

5.2.1 Kommunikationsablauf eines Dateiaufrufs

Die nachfolgende Grafik veranschaulicht, wie bei einem hierarchischen Dateisystem der Aufruf einer Datei als Kommunikationsfluss vom Nutzer bis an den tatsächlichen physikalischen Speicherort delegiert wird und welche Rolle dabei das Dateisystem einnimmt.

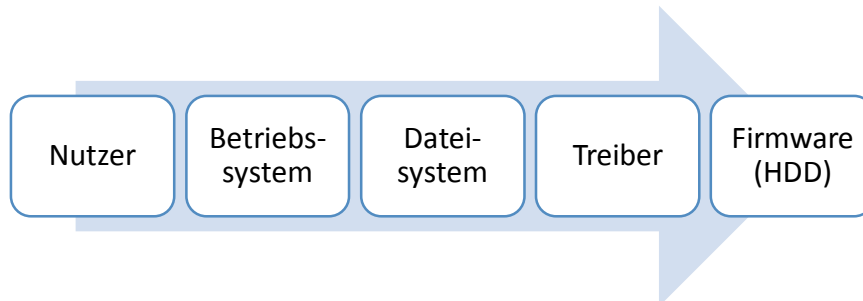


Abbildung 7 - Darstellung des Kommunikationsablaufes bei einem Dateiaufruf

Vorerst wählt der Nutzer die Datei aus, welche er verwenden möchte. Das Betriebssystem erkennt diese Aktion und leitet die Information über die Datei (Dateiname) an das Dateisystem weiter. Das Dateisystem durchsucht die Verzeichnisse und Metadaten und lokalisiert die Datei auf dem physikalischen Medium. Diese Informationen leitet das Dateisystem weiter an den Treiber der Hardware, dieser übergibt sie an die Firmware des Medium und die Datei kann lokalisiert werden. Deutlich wird, dass das Dateisystem die Kommunikation zwischen Betriebssystem und Hardware sicherstellt und zudem die Verwaltung und Organisation der Daten innehat.

5.3 Virtuelle Dateisysteme

Virtuelle Dateisysteme stehen eine Ebene über den normalen hierarchischen Dateisystemen. Ein virtuelles Dateisystem existiert nicht statisch auf der Festplatte sondern nur virtuell. Es verwaltet mehrere andere Dateisysteme in einem ohne, dass der Nutzer dies mitbekommt. Für den Nutzer wirkt es so, als lägen alle Dateien in einem Dateisystem vor. So können zum Beispiel externe Datenbanken derart in die Ordnerstruktur integriert werden, dass der Nutzer glaubt, sie wären alle lokal auf seinem Gerät abgespeichert.

6 Sicherheitsaspekte

Auch bei Dateisystemen spielt, wie in fast jeder IT-Komponente, die Sicherheit eine äußerst wichtige Rolle. Die beiden wesentlichen Bereiche der Sicherheit bei Dateisystemen, Zugriffsschutz und Datensicherheit, werden im Folgenden erläutert.

6.1 Zugriffsschutz

Da das Dateisystem für die Verwaltung der Dateien und Ordner auf dem Speichermedium verantwortlich ist, liegt es nahe, dass auch die Verwaltung von Zugriffsrechten für eben diese vom Dateisystem übernommen wird. Dafür nutzen Dateisysteme Zugriffsrechte, welche in der Metadaten-Tabelle mit abgespeichert werden. Für verschiedene Nutzer

oder auch Nutzergruppen kann so individuell festgelegt werden, welche Rechte sie haben. Standardmäßig unterscheiden die meisten Dateisysteme zwischen drei Zugriffsstufen: kein Zugriff, schreibgeschützter Zugriff und voller Zugriff.

Einige Dateisysteme, wie zum Beispiel NTFS, erlauben zudem das Verstecken von Dateien, sodass diese für Nutzer ohne Administratorrechte nicht sichtbar sind. Außerdem erlaubt NTFS mit Hilfe des dateisystemeigenen Werkzeugs AES eine Verschlüsselung von Daten um Dateien noch besser zu schützen. Eine Verschlüsselung kann aber auch durch dateisystemfremde Tools wie zum Beispiel TrueCrypt erfolgen, wenn sie vom Dateisystem nicht angeboten wird.

6.2 Datensicherheit

Für die Datensicherheit muss garantiert werden, dass ein Dateisystem in keinem Fall Daten verliert oder ungewollt verändert. Multizugriffe und Stromausfälle stellen dabei besondere Herausforderungen an das Dateisystem und müssen gelöst werden.

6.2.1 Multitasking

Multitasking beschreibt das Problem, dass mehrere Anwendungen zeitgleich auf verschiedene Dateien zugreifen wollen. Aufgabe des Dateisystems ist es nun, diese einzelnen Vorgänge sauber zu trennen und sicherzustellen, dass jede Anwendung genau auf die von ihr angefragte Datei und keine andere Zugriff erhält. Des Weiteren ist es auch möglich, dass mehrere Anwendungen gleichzeitig auf eine Datei zugreifen möchten oder eine Anwendung auf eine Datei zugreifen möchte, die bereits in einer anderen Anwendung verwendet wird. Hier gilt es genau wie bei Datenbankzugriffen zu serialisieren und sicherzustellen, dass Änderungen an der Datei nicht zu Datenverlusten aus der anderen Anwendung führen. Erreicht werden kann dies durch das Setzen von Sperren (Lesesperre, Exklusivsperrung und keine Sperre). Ist eine Datei mit einer Lesesperre belegt, heißt dies, eine Anwendung hat diese Datei zum Lesen ihrer Daten angefordert, dann können auch andere Anwendungen ohne Probleme eine Lesesperre anfordern, da gleichzeitiges Lesen einer Datei von mehreren Anwendungen nicht zu Datenverlusten führen wird. Zum Schreiben in einer Datei müssen entsprechende Anwendungen aber Exklusivsperrungen anfordern. Ist eine Datei mit einer solchen belegt, dann kann keine andere Anwendung mehr parallel auf diese Datei zugreifen, da es zu Konflikten kommen kann.

6.2.2 Stromausfälle

Eine weitere Bedrohung stellen Stromausfälle dar, sie sind zwar relativ selten, dann aber wirklich tragisch. Auch hier kann das Dateisystem unterstützen die Sicherheit zu erhöhen. Neben Optimierungen der Hardware, wo durch den Einbau von Kondensatoren kleine Energiepuffer eingebaut werden können, kann das Dateisystem besonders durch sogenanntes Journaling die Datensicherheit massiv beeinflussen. Beim Journaling werden alle Schreibvorgänge vor ihrer Ausführung vorerst in einem Journal notiert und dann sequentiell ausgeführt. So kann nach einem Stromausfall einfach nachgeschaut werden,

welche Datei als letztes geschrieben wurde und ob der Schreibvorgang erfolgreich war oder die Datei beschädigt wurde. Sollte die Datei beschädigt sein, können Reparaturmaßnahmen eingeleitet werden. Des Weiteren kann hier auch das Copy-on-Write-Prinzip angewendet werden, dabei wird eine Datei vor einer Änderung dupliziert, die duplizierte Datei bearbeitet und das „Original“ erst nach erfolgreichem Abschluss des Schreibvorgangs wieder gelöscht. Dies macht einen Kompletterverlust der Daten nahezu unmöglich wirkt sich aber negativ auf die Performance aus und wird daher nur bei kritischen Daten angewendet. [vgl. *Dateisystem, Wikipedia, 2015*] ; [vgl. *Journaling, Wikipedia, 2015*]

7 Fazit

Zusammenfassend lässt sich sagen, dass Dateisysteme ein, wenn auch häufig nicht bewusst wahrgenommenes, oder gar unbekanntes Element der Heutigen Gesellschaft sind. Sie bilden die Grundlage für die heute gängigen Systeme, indem sie die riesigen Datenmengen strukturieren und verwalten. Ohne Dateisysteme und die in ihnen verankerte Automatisierung wären wir nicht in der Lage diese Daten zu handhaben und damit derart komplexe Prozesse zu ermöglichen, wie wir sie heutzutage allzu oft vorfinden. Daran, wie unbekannt Dateisysteme im Allgemeinen sind, ist aber auch sehr gut zu erkennen, wie selbstverständlich deren Verwendung für uns im Laufe der Jahre geworden ist.

8 Abbildungsverzeichnis

Abbildung 1:	
Struktur eines Massendatenspeichers (<i>eigene Darstellung</i>)	3
Abbildung 2:	
Ablauf bei Dateiaufruf (<i>eigene Darstellung</i>)	6
Abbildung 3:	
schematisch vereinfachte Darstellung eines MBR (<i>eigene Darstellung</i>)	7
Abbildung 4:	
schematisch vereinfachte Darstellung eines GPT (<i>eigene Darstellung</i>)	7
Abbildung 5:	
vereinfachte Darstellung einer Partition im fragmentierten und defragmentierten Zustand (<i>eigene Darstellung</i>)	8
Abbildung 6:	
Verzeichnisstruktur verschiedener Dateisysteme https://de.wikipedia.org/wiki/Dateisystem#/media/File:Filesystem.svg [letzter Zugriff 23. September 2015]	9
Abbildung 7:	
Darstellung des Kommunikationsablaufes bei einem Dateiaufruf (<i>eigene Darstellung</i>)	10

9 Literaturverzeichnis

[IT-Wissen, 2015]

[Online] Verfügbar auf:

<http://www.itwissen.info/definition/lexikon/Dateisystem-file-system.html> [letzter Zugriff am 23. September 2014]

[Dateisystem, Wikipedia, 2015]

[Online] Verfügbar auf:

<https://de.wikipedia.org/wiki/Dateisystem> [letzter Zugriff am 23. September 2014]

[Datei, Wikipedia, 2015]

[Online] Verfügbar auf:

<https://de.wikipedia.org/wiki/Datei> [letzter Zugriff am 23. September 2014]

[Verzeichnisse, Wikipedia, 2015]

[Online] Verfügbar auf:

<https://de.wikipedia.org/wiki/Verzeichnis> [letzter Zugriff am 23. September 2014]

[Massendatenspeicher, Wikipedia, 2015]

[Online] Verfügbar auf:

<https://de.wikipedia.org/wiki/Massenspeicher> [letzter Zugriff am 23. September 2014]

[MBR, Wikipedia, 2015]

[Online] Verfügbar auf:

https://de.wikipedia.org/wiki/Master_Boot_Record [letzter Zugriff am 23. September 2014]

[GPT, Wikipedia, 2015]

[Online] Verfügbar auf:

https://de.wikipedia.org/wiki/GUID_Partition_Table [letzter Zugriff am 23. September 2014]

[Fragmentierung, Wikipedia, 2015]

[Online] Verfügbar auf:

https://de.wikipedia.org/wiki/Fragmentierung_%28Dateisystem%29 [letzter Zugriff am 23. September 2014]

[Journaling, Wikipedia, 2015]

[Online] Verfügbar auf:

<https://de.wikipedia.org/wiki/Journaling-Dateisystem> *[letzter Zugriff am 23. September 2014]*