

FSST Zusammenfassung

für die Reife -und Diplomprüfung

Datenmanagement Teil 1

Höhere Technische Bundeslehr- und Versuchsanstalt Anichstrasse

Verfasst von: Manuel Wechselberger

1 Dateisysteme

Dateisysteme ist Bestandteil des Betriebssystems und bilden die Schnittstelle zwischen diesem und den Datenträgern. Es legt fest wie Daten gespeichert, gelesen, verändert und gelöscht werden. Das Dateisystem besteht aus Dateien, Verzeichnissen und Adressen über die die Dateien lokalisiert werden.

1.1 Aufgabe

- Persistenz
- Konsistenz
- mutual exclusion
- Journaling
- Fragmentierung klein halten
- Verschlüsselung, Komprimierung, Verstecken, ...

Persistenz Mit Persistenz ist die Fähigkeit, Daten über einen längeren Zeitraum (vor allem über einen Programmabbruch hinaus) bereitzuhalten, gemeint. Das bedeutet beispielsweise, dass die Daten auch nach Beenden des Programms vorhanden (gespeichert) bleiben, und bei erneutem Aufruf des Programms wieder rekonstruiert und angezeigt werden können.

Konsistenz Konsistenz von Daten bedeutet dass die Korrektheit der gespeicherten Daten gewährleistet wird. Konsistenz hat vor allem Bedeutung bei Verteilten System wie sie beim Cloud-Computing und beim RAID zu finden sind.

mutual exclusion Damit ist eine Gruppe von Verfahren gemeint die verhindert, dass nebenläufige Prozesse bzw. Threads gleichzeitig oder zeitlich verschränkt gemeinsam genutzte Datenstrukturen verändern und diese Datenstrukturen somit inkonsistent werden würden. Auch wenn die Aktionen jedes einzelnen Prozesses oder Threads für sich betrachtet konsistenzhaltend sind.

Journaling Beim Journaling werden alle Änderungen in einem dafür vorgesehen Speicherbereiche, dem Journal, gespeichert. Damit ist es zu jedem Zeitpunkt möglich, einen konsistenten Zustand der Daten zu rekonstruieren. Befindet sich das Dateisystem in einem inkonsistenten Zustand (erster Schreibvorgang beim Kopieren) und der Computer muss neu gestartet werden (Absturz, Stromausfall), so muss das Dateisystem erst aufwändig auf solche Fehler untersucht werden.(Beim überspringen dieses Vorgangs: schwerwiegende Folgefehler, Datenverlust).

Journaling wirkt dem entgegen in dem die Änderungen in das Journal geschrieben werden. Zudem wird aufgezeichnet wie viele Änderungen bereits durchgeführt wurden. Somit kann nach einem System Neustart aus dem Journal entnommen werden

wo der Vorgang unterbrochen wurde und an dieser Stelle kann weiter gemacht werden.

Fragmentierung klein halten Das Dateisystem muss durch verschiedene Strategien Daten Fragmentierung vermeiden.

- Freilassen von 5 bis 20 Prozent des Speicherplatzes
- Trennen von Programmen und von Nutzerdaten.
- Verwendung einer höheren Blockgröße des Dateisystems
- Parametrierung der Applikationen
 - Preallocation (man reserviert vorsorglich schon Blöcke, obwohl diese noch nicht benötigt werden)
 - Zerlegung einer Festplatte in Cluster, Blockgruppen, Blöcke
 - Spätes Festlegen der zu benutzenden Speicherblöcke (late allocation) statt sofortigen Festlegens (early allocation)

1.2 Arten von Dateisystemen

1.2.1 Lineare Dateisysteme

Lineare Dateisysteme wurde früher vor allem bei Lochbändern und Lochkarten eingesetzt. Heutzutage finden sie noch Anwendung in der Professionellen Daten Sicherung bei Magnetbändern. Die Dateien liegen in einem einzigen Verzeichnis und können nur sequenziell gelesen und geschrieben werden. Durch die zunehmende Anzahl an Daten und die dadurch immer längeren Zugriffszeiten war es unvermeidlich ein neues strukturelles Konzept zur Datenorganisation zu entwickeln

1.2.2 Hierarchische Dateisysteme

Durch größere Datenmengen wurde es schwer den Überblick über die Dateien zu behalten, deshalb wurde das Konzept der Unterverzeichnisse eingeführt. Es existiert ein Wurzelverzeichnis das neben normalen Dateien auch Verweise auf Unterverzeichnisse, welche wiederum Verweise auf Unterverzeichnisse, enthalten können. Dadurch entsteht eine Verzeichnisstruktur, die oft als Verzeichnisbaum dargestellt wird. Da sich eine Hierarchie von Verzeichnissen und Dateien ergibt spricht man von einem Hierarchischen Dateisystem. Die Verzeichnisse werden durch Trennzeichen, welche sich von OS zu OS unterscheiden, getrennt.

C:\Dokumente und Einstellungen\benutzername\Eigene Dateien\Texte\Brief1.txt

Das oben gezeigte Beispiel wird auch Pfad genannt und mit Hilfe von ihm kann durch das Dateisystem navigiert werden.

Netzwerkdateisysteme Systemaufrufe können auch über ein Netzwerk an einen Server übertragen werden. Dem Client werden dann die angeforderten Daten zurück gesendet. Da die selben Systemaufrufe verwendet werden unterscheidet sich der Zugriff für den Client nicht von einem Lokalen Dateisystem.

Globale- oder Cluster-Dateisysteme: sind Dateisysteme auf die mehrere Clients parallel Zugriff haben. Dazu werden Metadaten-Server eingesetzt. Er erhält alle Zugriffe und stellt sicher dass kein Datenverlust durch das Überschreiben von Dateien auftritt. Der eigentliche Datenverkehr erfolgt dann über ein SAN(Storage Area Network). Für diese Dateisysteme werden spezielle Netzwerkprotokolle verwendet.

Multiple Streams Moderne Dateisysteme speichern Daten nicht in einer Folge von Bytes (Stream) sondern mehrere dieser Folgen (multiple Streams). Dadurch können Teile einer Datei bearbeitet werden, ohne andere Teile verschieben zu müssen. Ein Problem besteht in der Unterstützung von multiplen Streams. Daten gehen beim Transfer auf andere Dateisystem verloren.

1.2.3 Assoziative Dateiverwaltung

Diese werden häufig fälschlicherweise als Datenbankdateisysteme oder SQL-Dateisysteme bezeichnet, hierbei handelt es sich eigentlich nicht um Dateisysteme, sondern um Informationen eines Dateisystems, die in aufgewerteter Form in einer Datenbank gespeichert und in, für den Anwender intuitiver Form, über das virtuelle Dateisystem des Betriebssystems dargestellt werden.

1.3 Dateisystem Beispiele

NTFS: Im Vergleich zum Dateisystem FAT bietet NTFS unter anderem einen gezielten Zugriffsschutz auf Dateiebene sowie größere Datensicherheit durch Journaling. Allerdings ist keine so breite Kompatibilität gegeben wie bei FAT. Ein weiterer Vorteil von NTFS ist, dass die Dateigröße nicht wie bei FAT auf 4,3 GB beschränkt ist sondern auf 17 TB in der aktuellen Version (NTFS 3.1).

Bei NTFS werden alle Informationen zu Dateien in einer Datei, dem Master File Table (MFT) gespeichert. In dieser Datei befinden sich die Einträge, welche Blöcke zu welcher Datei gehören, die Zugriffsberechtigungen und die Attribute.

Attribute unter NTFS:

- Dateigröße
- Datum der Dateierstellung
- Datum der letzten Änderung
- Freigabe
- Dateityp
- Dateinhalt

2 Arbeitsspeicher

Im Arbeitsspeicher sind die gerade ausgeführten Programme oder Programmteile und die dabei benötigten Daten enthalten. Der Arbeitsspeicher kann in physikalischen und virtuellen Speicher unterteilt werden. Als physischer Speicher wird die Hardware bezeichnet, die direkt über den Adressbus angesprochen werden kann. Der virtuelle Speicher bezeichnet den vom tatsächlich vorhandenen Arbeitsspeicher unabhängigen Adressraum.

Bei modernen Computern gibt es oft nicht genug Hauptspeicher für alle aktiven Prozesse, also müssen einige von ihnen auf der Festplatte gespeichert und bei Bedarf dynamisch in den Arbeitsspeicher zurückgeholt werden. Für diese Art von Speicher-verwaltung gibt es zwei grundlegende Ansätze. Die einfachere Strategie ist das so genannte Swapping, bei dem jeder Prozess komplett in den Speicher geladen wird, eine gewisse Zeit laufen darf und anschließend wieder auf die Festplatte ausgelagert wird. Bei der anderen Strategie, dem virtuellen Speicher, können Programme auch dann laufen, wenn sich nur ein Teil von ihnen im Hauptspeicher befindet.

2.1 Swapping

Gerade nicht benötigte Daten werden auf eine Festplatte ausgelagert, um Platz im Arbeitsspeicher zu schaffen. Werden sie wieder benötigt, müssen sie wieder geladen werden (Swapping). Der wichtigste Aspekt der realen Speicherverwaltung ist, dass nur wirklich vorhandener physikalischer Hauptspeicher benutzt wird. Programme können also nicht größer als der zur Verfügung stehende Hauptspeicher sein. Also können in der Regel nicht alle Prozesse gleichzeitig im Speicher gehalten werden.

2.2 Virtuelle Speicherverwaltung

Die Virtuelle Speicherverwaltung ermöglicht es Prozessen mehr Arbeitsspeicher als physikalisch zur Verfügung steht zu verwenden. Dazu wird der virtuelle Speicher benötigt, der mit virtuellen Adressen arbeitet. Diese virtuellen Adressen gehen nun nicht mehr direkt an den Speicherbus sondern an die MMU (memory management unit) welche die virtuellen Adressen auf die physischen abbildet. Durch Swapping ist es möglich, dass der virtuelle Speicher sehr viel größer als der physische Speicher sein kann. Die Größe des virtuellen Speichers hängt von der Architektur des Befehlssatzes ab.

- 32-Bit-Prozessor: maximal $2^{32} \text{Bytes} = 4\text{GB}$
- 64-Bit-Prozessor: maximal $2^{64} \text{Bytes} = 16\text{Exabytes} = 16\text{mio.Terrabytes}$

Zur Organisation des virtuellen Speichers gibt es einerseits den segmentorientierten Ansatz, bei dem der Speicher in Einheiten unterschiedlicher Größen aufgeteilt ist, und andererseits den seitenorientierten Ansatz, bei dem alle Speichereinheiten gleich lang sind.

2.3 Paging

Paging ist eine Methode zur virtuellen Speicherverwaltung von virtuellem Speicher der nach dem seitenorientierten Ansatz aufgeteilt ist.

Begriffserklärung:

- Seite: Als Seite werden die gleich langen Speichereinheiten im virtuellen Speicher bezeichnet.
- Seitenrahmen: sind die Speichereinheiten im physikalischen Speicher. Sie haben die gleiche Größe und sind gleich aufgeteilt wie die Seiten.
- Seitentabelle: Enthält Informationen welche Seite zu welchem Seitenrahmen gehört.

Ein Prozess möchte auf den Arbeitsspeicher zugreifen. Er erhält eine virtuelle Adresse vom Betriebssystem. Diese Adresse enthält eine virtuelle Seitennummer über die die entsprechende Seite in der Seitentabelle lokalisiert werden kann. Der Eintrag in der Seitentabelle enthält einen Verweis auf den entsprechenden Seitenrahmen im physikalischen Speicher.

Seitenauslagerung Ein Beispiel: Die physische Größe des Arbeitsspeichers sei 64 KByte. Ein Programm benötigt insgesamt 200 KByte Speicher. Um das größere Programm trotzdem auf dem kleineren Arbeitsspeicher ausführen zu können, kann man den Adressraum in Seiten (Pages) aufteilen, beispielsweise vier Seiten zu 64 KByte, wobei dann die letzte Seite nur teilweise gefüllt ist. Es befindet sich dann jeweils eine der vier Seiten im physischen Arbeitsspeicher, die anderen drei sind auf die Festplatte ausgelagert.