

6Übung1Aufgabe

1. I/O, DMA

I/O steht für Input/Output und bezeichnet die Kommunikation des Systems mit der Außenwelt

z.B Benutzern und Aktionen zum Lesen und Schreiben von Daten.

(https://de.wikipedia.org/wiki/Eingabe_und_Ausgabe)

DMA (Direct Memory Access) ist eine I/O Art, bei der die Aufgaben ausprogrammiert werden, indem sie an einen DMA Controller/module delegiert werden. Dieser besteht aus

eine for-loop in extra Hardware also ein einfacher Prozess, der die Daten Stück für Stück überträgt. Sobald alle Daten übertragen sind, schickt der DMA-Controller einen Interrupt an den Prozessor.

(Quelle: VL Folie 6.8f und Mitschrift)

2. I/O Buffering

Die Raten ein- und ausgelesener Daten müssen zwar im Mittel gleich groß sein, können aber

zwischenzeitlich schwanken. I/O Buffering sorgt mit einem oder heutzutage mehreren Puffern

dafür, dass dies nicht zu Problemen führt. Diese Puffer sind Teile des Hauptspeichers, auf denen zwischenzeitlich Daten abgelegt werden und die vom Betriebssystem verwaltet werden.

(Quelle: VL Folie 6.12 und Mitschrift)

3. RAID(redundant array of independent/inexpensive disks)

Festplatten, besonders günstigere, sind sehr anfällig und gehen schnell kaputt. RAID ist eine Methode bei der man eine Festplatte mehr nutzt (häufig 5 statt 4), als man braucht und die

Informationen so redundant speichert, dass falls eine Festplatte ausfällt, deren Inhalt

durch die anderen wiederhergestellt werden kann. Das Betriebssystem sieht die vielen

Festplatten als eine logische Einheit.

(Quelle: VL Folien 6.31,32 und Mitschrift)

4. Tree-Structured Directory

Diese Dateistruktur nennt man auch hierarchisch, da es ein master/root Verzeichnis gibt mit

Unterverzeichnissen. Jedes Verzeichnis/jeder Ordner kann Dateien sowie weitere Ordner beinhalten

und einige Betriebssystem haben auch mehrere Bäume mit verschiedenen master Verzeichnissen.

Man kann eine Datei mit ihrem Pfad vom master/root Verzeichnis verorten, oder aber auch

ausgehend vom momentanen Arbeitsverzeichnis (Current Working Directory CWD).

(Quelle: VL Folien 6.59,60)

5. Virtual File System (VFS)

Das virtuelle Dateisystem in Linux ist eine Schicht zwischen dem kernel und dem Dateisystemcode, welches

alle verschiedenen Dateisysteme verwaltet, die montiert bzw. ausgerüstet sind.

Die echten Dateisysteme werden entweder in den kernel selbst eingebaut oder als ladbare Module gebaut.

(Quelle: VL Folien 6.93)

6. Indexed Allocation

Dateien werden in Blöcke zerlegt und so auf der Festplatte gespeichert. Um zu wissen,

welche Blöcke zu welcher Datei gehören, wird eine File Allocation Table (FAT) angelegt.

Indexed Allocation ist ein Kompromiss zwischen chained und contiguous Allocation, denn

die FAT Tabelle enthält einen index für jede Datei, der einen Eintrag für jeden der Datei

zugehörigen Block hat. Dadurch kennt der Schreib-Lese-Kopf der Festplatte jeden index und

kann diese sinnvoll sortieren, sodass unnötiges, zeitaufwendiges Platzieren des

6Übung1Aufgabe

Kopfes vermieden wird.

(Quelle: VL Folien 6.68,70 und Mitschrift)

7. Inode

Bei UNIX werden alle Arten von Dateien (Ordner sind auch Dateien) vom Betriebssystem über inodes verwaltet. Diese Kontrollstruktur speichert alle wesentlichen Infos für eine Datei wie z.B.

Benutzerkennung (Rechte), Größe, Zugriffsdaten (wann zuletzt geöffnet, modifiziert...).

Ein aktiver inode ist immer mit genau einer Datei assoziiert, die von dem inode kontrolliert wird.

Während Dateien jedoch unterschiedlich groß sein können, haben inodes immer eine feste Länge (128 Byte).

(Quelle: VL Folien 6.83,94)

b) Informieren Sie sich über die Dateisysteme NTFS und ext4. Arbeiten Sie die wesentlichen Unterschiede heraus und nennen Sie Vor- bzw. Nachteile. Welches Dateisystem bevorzugen Sie? Benutzen Sie hier ausschließlich Primärliteratur.

Das Dateisystem NTFS (New Technology File-System) bietet Leistung, Zuverlässigkeit, erhöhte Sicherheit und erweiterte Funktionen.

Zum Beispiel enthält NTFS integrierte Sicherheitsfunktionen wie Datei- und Ordnerberechtigungen und

das Encrypting File System (EFS). Dieses wird verwendet für das Speichern von verschlüsselten Dateien auf NTFS-Volumes.

NTFS bietet auch Unterstützung für Volumes bis zu 256 Terabyte Größe, sowie für Datenträger Quoten und

Komprimierung als auch für montierte Laufwerke an.

Mit NTFS kann man Berechtigungen für eine Datei oder einen Ordner festlegen und die Gruppen und Benutzer angeben,

deren Zugriff man einschränken oder zulassen möchte. Jeder Eindringling, der versucht, auf die verschlüsselten

Dateien zuzugreifen, wird daran gehindert, auch wenn dieser Eindringling physischen Zugriff auf den Computer hat.

-> erhöhte Sicherheit

NTFS ist eher für die Windows-Reihe proprietär und es unterstützt auch große Bände und leistungsstarke Speicherlösungen wie RAIDS.

Was ist mit Zuverlässigkeit der Leistung gemeint?

Wenn der Computer im Falle eines Systemfehlers neu startet, verwendet NTFS seine Protokolldatei und

Checkpoint-Informationen, um die Konsistenz des Dateisystems wiederherzustellen.

Im Falle eines Fehlers des Fehlers wird NTFS den Cluster, der den schlechten "Sektor" enthält,

dynamisch neu zuordnen und einen neuen Cluster für die Daten zuordnen. Der Cluster,

wo der Fehler war, wird als "schlecht" markiert, damit dieser nicht mehr verwendet wird.

Zum Beispiel, durch die Formatierung eines POP3 Mail-Servers mit NTFS,

kann der Mail-Store Protokollierung und Wiederherstellung anbieten. Im Falle eines Server-Crashes kann NTFS Daten

wiederherstellen, indem es seine Protokolldateien wiedergibt. -> Steigerung der Zuverlässigkeit

Das EXT4 (fourth extended filesystem) ist für Linux bestimmt.

EXT4 erlaubt Zeichen im Dateinamen, die nicht in NTFS erlaubt sind

(/, \, :, *, ", ?, <, >, |).

Dies hat aber nichts mit der Leistung von EXT4 zu tun. EXT4 gilt als sehr stabil und bietet ein größeres

Dateisystem und eine größere Dateigröße an. Damit hat es 1 EiB als maximalen Dateisystemgröße und 16 TiB der

maximalen Dateigröße in Linux, wobei die Dateigröße auch dem gesamten

6Übung1Aufgabe

Dateisystem entsprechen kann(siehe Notiz unten).

EXT4 verdoppelt die Unterverzeichnisse zu seinem Vorgänger EXT3 auf ganze 64000. Das EXT4 besitzt sogenannte Extents, welche die Geschwindigkeit bei der Verwaltung größerer Dateien erhöht.

Extends(Ausmaße) sind eine Art Haufen von zusammenhängenden, physischen Blöcken. Riesige Dateien sind in mehrere

Ausmaße aufgeteilt. Die Ausdehnung verbessert die Leistung und hilft die Fragmentierung zu reduzieren, da ein Ausmaß

kontinuierliche Layouts auf der Festplatte fördert. Wenn neue Daten auf die Festplatte geschrieben werden verwendet

EXT4 einen "Multiblock-Zuweiser" (mballoc), der viele Blöcke in einem einzigen Anruf verteilt, anstelle eines einzelnen

Blocks pro Anruf. Dies verbessert nochmals die Leistung und ist besonders nützlich bei verzögerter Zuteilung und Extents.

Dieses Vorgehen wirkt sich dabei nicht auf das Festplattenformat aus.

Verbesserung der Inode bei EXT4. Die Inode-Reservierung besteht darin, mehrere Inodes zu reservieren, wenn ein Verzeichnis

erstellt wird und erwartet, dass sie in der Zukunft verwendet werden. Dies verbessert die Leistung, denn wenn

neue Dateien in diesem Verzeichnis erstellt werden, können sie die reservierten Inodes verwenden.

Die Erstellung und Löschung von Dateien ist daher effizienter. EXT4 weist auch einen verbesserten Zeitstempel auf Nanosekunden-Basis auf.

Was bringt die Persistence preallocation bei EXT4?

Anwendungen sagen dem Dateisystem, den Speicherplatz zuzuordnen und das Dateisystem stellt dann die notwendigen Blöcke

und Datenstrukturen vor. Aber noch gibt es keine Daten darüber, bis die Anwendung wirklich die Daten reinschreibt.

Das macht die P2P-Anwendungen, denn diese ordnet den notwendigen Platz für einen Download vor,

der Stunden dauern würde. Dies hat mehrere Verwendungen:

Erstens, um Anwendungen zu vermeiden, macht es sich selbst ineffizient durch das Ausfüllen einer Datei mit Nullen.

Zweitens, um die Fragmentierung zu verbessern, da die Blöcke zu einer Zeit, so angrenzend wie möglich zugewiesen werden.

Drittens, um sicherzustellen, dass Anwendungen immer den Platz haben, den sie kennen und benötigen.

Ext4 prüft die "Journaldaten", um zu wissen, ob die "Journalblöcke" fehlschlagen oder beschädigt sind. Das nennt man

Journal-checksumming und hat einen Vorteil: Es erlaubt, das zweiphasige

Commit-System von Ext3's Journaling

auf eine einzige Phase umzuwandeln und die Dateisystemoperation in einigen Fällen zu beschleunigen.

So werden gleichzeitig Zuverlässigkeit und Leistung verbessert.

Die wesentlichen Vor- und Nachteile der beiden Filesystems:

Ein Vorteil ist erstmal, dass beide also NTFS und Ext4 ein Journal über alle Schreibvorgänge besitzen,

damit ermöglichen beide Dateisysteme den letzten konsistenten Zustand zu

rekonstruieren. Nachteil der beiden ist,

dass EXT4 nur für Linux bestimmt ist und nicht wirklich kompatibel für andere Betriebssysteme ist. Genauso wie NTFS

nur für Windows geschaffen ist, zwar auch auf Linux läuft, aber beide

Dateisysteme generell ihre Leistung nur in den

jeweiligen Betriebssystemen voll entfalten können.

Das Maximum der Dateigröße bei EXT4 liegt bei 16TiB und hat 1EiB als maximale Dateisystemgröße

Die Dateigröße kann beim EXT4 so groß wie das Dateissystems selbst sein

wohingegen beim NTFS die Größe begrenzt ist.

-> folglich ist NTFS generell gesehen größer als EXT4 und bietet damit mehr

Speicherplatz, aber mit EXT4

kann man größere Dateien anlegen als bei NTFS. Man muss aber beachten, dass die Betriebssysteme die großen

einschränken können.

6Übung1Aufgabe

Das Maximum der Dateilänge von NTFS liegt bei 255 Zeichen, aber bei EXT4 bei 255 Bytes -> also beinahe "identisch"

NTFS erlaubt online-Komprimierung und online-Verschlüsselung und EXT4 nicht, können aber beide ihre Daten an sich verschlüsseln.

Bei einem Absturz des Computers bietet NTFS mehr Sicherheit als EXT4 aufgrund von Zeitverzögerungen bei den Datenblöcken

und den Inodes(siehe oben). Ein weiterer Vorteil gegenüber NTFS hat EXT4 bei der Länge der Pfadnamen,

die ebenfalls in NTFS begrenzt sind aber in EXT4 nicht. NTFS ist auch ein wenig empfindlicher was Groß- und

Kleinschreibung betrifft bei windows. Z.B. kann man keine 2 Ordner oder Dateien anlegen die bspw.

programm und PROGRAMM heißen, da man dann eine Datei durch die andere ersetzen muss.

Wir bevorzugen EXT4, da es von der Geschwindigkeit uns mehr überzeugt durch seine Features und vorallem

es ermöglicht beliebig große Dateien schnell von A nach B zu verlegen, solange diese hält nicht die Dateisystemgröße

übergehen. Durch mballoc ist auch die Fragmentierung damit besser. NTFS würde uns eher durch mehr

Sicherheit der Dateien locken.

Notiz am Rande:

1EiB=1024PiB

1PiB=1024TiB

1TiB=1024GiB

Quellen: <https://msdn.microsoft.com/de-de/library/cc758691.aspx> /

https://ext4.wiki.kernel.org/index.php/Ext4_Howto#Sub_directory_scalability /

<https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/plain/Documentation/filesystems/ext4.txt?id=HEAD>