

Aufgabe 1

a) Unterschied zwischen harten und symbolischen Links:

- Der Hauptunterschied zwischen den beiden ist das symbolische Links quasi den Pfad zur Originaldatei enthalten in einer neuen Datei, diese können dann auch Volumes und Dateisysteme überqueren. Hardlinks hingegen sind quasi Spiegelkopien der Originaldatei, welches mithilfe der Verknüpfung der Inode realisiert wird.

Symbolische Links Vorteile:

würde ich nicht als Verknüpfung bezeichnen, es ist ein Verzeichniseintrag

mit der Metapher kann nicht unbedingt jeder was anfangen

- Man kann dadurch Volumes und Dateisysteme überqueren
- Link kann problemlos gelöscht werden ohne Auswirkungen und Einschränkungen

Symbolische Links Nachteile:

- Nach Umbenennung / Verschiebung der Originaldatei funktioniert der Link nicht mehr
- Wenn die Originaldatei gelöscht wird, zeigt der Link aufs leere

Hardlinks Vorteile:

- Nach Umbenennung / Verschiebung der Originaldatei funktioniert der Link noch
- Die Datei wird erst gelöscht, wenn auch wirklich alle Hardlinks inklusive Originaldatei gelöscht wird.

-0.125 bei Hardlinks auch
-0.125 das ist das gleiche Argument, dass die Originaldatei nicht mehr da ist

Hardlinks Nachteile:

- Man kann dadurch Volumes und Dateisysteme nicht überqueren
- Man kann Hardlinks nicht auf Verzeichnisse machen (z.B. unter Linux)

sowas muss man nicht erläutern, aber ist natürlich nicht falsch

b) Zu jeder Datei gehört eine Datenstruktur, der sogenannte I-Node. I-Node ermöglicht Zugriff auf alle Blöcke der Datei. Diese befindet sich im Hauptspeicher, sobald die Datei offen ist. Befinden wir uns nun in einem anderen Dateisystem und wollen einen Hardlink zu einer Datei im anderen Dateisystem erstellen, funktioniert ja die Verknüpfung der Inodes nicht mehr, da diese sich ja auf dem Dateisystem der Datei befindet. Man kann also hier nur softlinks verwenden. Würde es gehen und wir würden ein hardlink auf ein externen Speicher anlegen (USB) und entfernen diesen wieder ohne den Link zu löschen bleibt der Linkzähler unverändert und die Originaldatei kann nicht mehr gelöscht werden unter Umständen.

was soll das sein?

perfekt ^_^

c) Somit werden Konsistenzprobleme und Schleifen der Dateistruktur vermieden. Dadurch kann die Baumstruktur verletzt werden.

4/4

Aufgabe 2

Angabe	Angabe in Bits		Angabe in Bytes	
	2er-Potenz	dezimal	2er-Potenz	dezimal
2 Byte	2^4 Bit	16 Bit	2^1 Byte	2 Byte
2048 MiB	2^34 Bit	17.179.869.184 Bit	2^31 Byte	2.147.483.648 Byte
32 Byte	2^8 Bit	256 Bit	2^5 Byte	32 Byte
16 MiBit	2^24 Bit	16.777.216 Bit	2^21 Byte	2.097.152 Byte
1024 KiBit	2^20 Bit	1.048.576 Bit	2^17 Byte	131.072 Byte

b) Ich glaube bei der gekauften Festplatte handelt es sich um 3 GB. Was ein erheblicher Unterschied zu 3 TiB wäre. Das wäre für den Hersteller einfach günstiger.

3000 GB = 3.000.000 MB = 3.000.000.000 KB = 3.000.000.000.000 Bytes = 2.929.687.500 KiB =

2.861.022 MiB = 2.793 GiB = 2,728 TiB

3 TiB = 3.298.534.883.328 Bytes also 3072 Gib also Unterschied 279 Gib

etwas ausführlicher: da die Festplattenkapazität in SI-Einheiten "größer" aussieht als in Zweierpotenz-Einheiten aber passt

Aufgabe 3

1.5/3

a) -1

b)

Verzeichniseinträge:

Dateiname	Erweiterung	Datei-Attribute	Erster Plattenblock	Dateigröße
BRIEF	TXT	(...)	4	129 KB
EDITOR	EXE	(...)	6	101 KB
⋮	⋮	⋮	⋮	⋮

AUFGABE DOC (...) 15 158KB

Bei FAT32 Blockgröße von 32KB. Also 5 Blöcke benötigt.

Plattenblock 0

Plattenblock 1 8

Plattenblock 2 10

Plattenblock 3 11

Plattenblock 4 7

Plattenblock 5

Plattenblock 6 3

Plattenblock 7 2

Plattenblock 8 9

Plattenblock 9 -1

Plattenblock 10 12

Plattenblock 11 14

Plattenblock 12 -1

Plattenblock 13 1

Plattenblock 14 -1

Plattenblock 15 13

Liste freier Plattenblöcke:

15	13	1	8	9	5	0	...
---------------	---------------	--------------	--------------	--------------	---	---	-----

Durchstreichen wurde vergessen -0,5

Aufgabe 4

a)

$$N_b = 10 + b/z + (b/z)^2 + (b/z)^3$$

bei Datenmenge | Byte geht man immer von 2^{10} usw. aus

b)

Zeigergröße: 4 Byte

-0.5

man sollte hier mit 1024 und nicht mit 1000 rechnen 2^{10} usw. und nicht 10^3 usw.

Blockgröße 1 KB:

$$\begin{aligned} b * N_b &= 1000 * (10 + 1000/4 + (1000/4)^2 + (1000/4)^3) \\ &= 1000 * (10 + 250 + 62500 + 15.625.000) \\ &= 1000 * 15.687.760 \\ &= 15.687.760.000 = 15,687760 \text{ GB} \end{aligned}$$

aber korrektes Ergebnis
mit Folgefehler

Blockgröße 4 KB:

$$\begin{aligned} b * N_b &= 4000 * (10 + 4000/4 + (4000/4)^2 + (4000/4)^3) \\ &= 4000 * (10 + 1000 + 1.000.000 + 1.000.000.000) \\ &= 4000 * (1.001.001.010) \\ &= 4.004.004.040.000 = 4004,00404 \text{ GB} \end{aligned}$$

Größe eines Zeigers auf Plattenblock: 4 Byte also 2^{32} Adressierungsmöglichkeiten