

Betriebssysteme

Übungsblatt 8

Micha Erkel

Felix Ruh

Aufgabe 1

a) symbolische vs harte Links:

	symbolische Links	harte Links
Unterschied:	Symbolische Links enthalten Pfade zu Objekten	Hardlinks erstellen ein Verzeichniseintrag, mit neuem Namen, mit Verweisung auf das eigentliche Zielobjekt auf der Festplatte.
Vorteile:	Kann auf Verzeichnisse zeigen	Funktioniert nach umbenennen und verschieben des Referenz Objekts noch
Nachteile:	Funktioniert nach umbenennen oder verschieben des Referenz Objekts nicht mehr Zum löschen des Referenz Objekts muss nur das Objekt selbst gelöscht werden	Können zur Sicherung des Referenz Objekts dienen, da Datei beim löschen in dem Link erhalten bleibt Referenz Objekt muss im selben Dateisystem liegen

etwas unpräzise

meint ihr über Dateisystemgrenzen hinweg?

sehr unpräzise

jo, kinda,, auch wenn das nicht die Absicht ist, ist es ein Nebeneffekt

- b) Ein möglicher Grund ist, dass wenn ein Hardlink angelegt wird auch der Linkzähler um 1 erhöht wird. Trenne ich jetzt beide Dateisysteme (bspw. die Verbindung zwischen zwei Festplatten wird unterbrochen) kann das System, beim löschen der Referenzdatei, diesen Link nicht entfernen, der Linkzähler geht nicht auf 0 und die Datei könnte nie gelöscht werden.
- c) Harte Links auf ein Verzeichnis würden die hierarchische Baumstruktur des Verzeichnissystems stören und könnten zu Schleifen führen.

Aufgabe 2

a) Die Tabelle:

Angabe	Angabe in Bits		Angabe in Bytes	
	2er-Potenz	dezimal	2er-Potenz	dezimal
2Byte	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) Die HerstellerInnen wählen 3TB = 3 TeraByte, da dadurch Herstellungskosten gespart werden können. Bei einer 3TB Festplatte fällt dadurch ein Unterschied von 278,032 GiB an.

Aufgabe 3

a) Ein Hardlink zeigt auf ein Referenzobjekt auf der Festplatte. Er würde also bei aufrufen des Links keine auslesen des entsprechenden Verzeichniseintrags statt finden, sonder direkt die Datei ausgelesen werden. Dies würde zu schwerwiegenden Problemen, z.B. wegen den umgangenen Zugriffsbeschränkung, führen.

b) FAT:

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
...	...

-1 es ist alles sehr waage und bleibt unklar
 >_<

Verzeichniseinträge:

Dateiname	Erweiterung	Datei-Attribute	Erster Plattenblock	Datei-Größe
BRIEF	TXT	(...)	4	129 KB
EDITOR	EXE	(...)	6	101 KB
AUFGABE	DOC	(...)	15	158 KB

Liste freier Plattenblöcke:

5	0	...
---	---	-----

Aufgabe 4

- a) Die Formel: $N_b(b, z) = 10 + \sum_{j=1}^3 \left(\frac{b}{z}\right)^j$

-0.5 floor Klammern, weil man keinen halben oder viertel Zeiger haben kann. Zeiger zeigen auf Adressen, diese muss vollständig sein, damit er auf die richtige Stelle zeigt.

Das Maximum an freien Datenblöcken bedeutet, dass keiner bisher belegt ist. Dabei ist b/z die Anzahl der Zeiger (wie in der Vorlesung berechnet) die auf die nächsten Daten- bzw. Plattenblöcke zeigen. Die Potenzen folgen daraus, dass jeder Plattenblock diese Zahl an Zeigern enthält und diese wiederum auf jeweils ein weiteren Plattenblock gefüllt mit dieser Zahl an Zeigern, welche auf die Datenblöcken zeigen, deutet.

- b) Für 1 KB:

$$g_{max}(1KB, 4Byte) = 1KB \cdot N_b(1KB, 4Byte)$$

$$g_{max}(1KB, 4Byte) = 15.687.760.000Byte \approx 15,7GB$$

Für 4 KB:

$$g_{max}(4KB, 4Byte) = 4KB \cdot N_b(4KB, 4Byte)$$

$$g_{max}(4KB, 4Byte) = 4.004.004.040.000Byte \approx 4TB$$

-0.5 man sollte hier mit 1024 rechnen, nicht 1000:

$$a = 1000 / 4$$

$$(10 + a + a^{**2} + a^{**3}) * 1000$$

$$a = 4000 / 4$$

$$(10 + a + a^{**2} + a^{**3}) * 4000$$

sollte eign. 4096 sein

32-Bit Blocknummer $\rightarrow 2^{32}$ eindeutige Adressierungsmöglichkeiten