

Betriebssysteme Blatt 8

Baran Güner, bg160

Tobias Hangel, th151

16. Dezember 2022

Aufgabe 1

3/4

a)

Symbolische Links enthalten Pfade zu Referenzobjekten, wohingegen harte Links einen neuen Verzeichniseintrag in Dateisystem erstellen, indem das Dateiojekt einen neuen Namen erhält.

unpräzise aber korrekt ^^

Vorteile an symbolischen Links sind, dass ein beliebiges Objekt im Dateibaum verlinkt werden kann und dass kein zusätzlicher Verzeichniseintrag in Dateisystem erstellt werden muss.

-0.25 doch...

Nachteilhaft ist, dass bei Verschieben, Löschen oder umbenennen des Objekts der Link nicht mehr funktioniert, da er nun ins Leere zeigt.

Außerdem kann der Link nicht als Backup für das Dateiojekt fungieren. Wenn die Datei gelöscht wird, ist sie endgültig verschwunden.

kreativ, hab ich bisher nicht gesehen ^_^

nen Hardlink ist einfach nur ein Verzeichniseintrag, ein Symbolischer Link ist Verzeichniseintrag + I-Node + Datenblock

Vorteile an harten Links sind, dass sie auch noch funktionieren, wenn die Datei umbenannt oder verschoben wird und dass die Datei erst dann komplett gelöscht wird, wenn der Linkzähler den Wert 0 hat.

Nachteile sind, dass sich das Dateiojekt in der selben Festplattenpartition befinden muss wie der Link und dass ein neuer Verzeichniseintrag notwendig ist.

b)

-1

Ein Harter Link referenziert die I-Node der Datei. Wenn die Datei sich aber in einer fremden Partition oder auf einem externen Medium befindet, kann auf diese I-Node nicht zugegriffen werden. Würde man die Datei beispielsweise verschieben, würde sich auch die I-Node ändern, wovon der Hard Link nichts wissen könnte.

mir ist unbekannt, was sich durch Verschieben am I-Node ändern sollte

es wird nur ein Verzeichniseintrag gelöscht und woanders eben dieser wieder erstellt

das Problem ist wie im Tutorat angesprochen, dass man Dateisysteme unmounten kann und es dann zu Problemen wegen Linkzähler kommt usw.

c)

Hierdurch könnten unendliche Loops entstehen, wenn man beispielsweise einen Hard Link in einem Verzeichnis erstellen würde, der zu seinem eigenen Elternverzeichnis führt. Hierdurch würde auch die Eindeutigkeit von Eltern- und Kinderverzeichnissen zerstört werden und Dateien könnten sich multiplizieren.

Aufgabe 2

a)

3.5/4

	Angabe in Bits		Angabe in bytes	
Angabe	2er-Potenz	dezimal	2er-Potenz	dezimal
2 Byte	2^4	16	2^1	2
2048 MiB	2^{34}	17179869184	2^{31}	2147483648
32 Byte	2^8	256	2^5	32
16 MiBit	2^{24}	16777216	2^{21}	2097152
1024 KiBit	2^{20}	1048576	2^{17}	131072

b)

Der Hersteller wählt höchstwahrscheinlich die traditionelle Interpretation, da 3 TB kleiner sind als 3 TiB und der Speicher so größer wirkt, als er tatsächlich ist. Die 3 TB Platte hätte dann ca 2,73 TiB.

gut erklärt! ^_^

traditionel im Sinne von vor

der Erfindung von Computern?

als man noch nicht

binär nachdenken musste?

-0.5 man sollte noch den Unterschied berechnen

3/3

Aufgabe 3

a)

Im Allgemeinen sind Hardlinks Verzeichniseinträge zu einem Zielobjekt und in einem Dateisystem mit I-Nodes ist es ein Verzeichniseintrag auf die

Hard Links referenzieren I-Nodes, welche die Metadaten/Dateiattribute enthalten. Im FAT32 System existieren jedoch keine I-Nodes und die Dateiattribute stehen direkt im Verzeichnis. Würde man im FAT32 System einen Hard Link auf den Anfangsblock einer Datei erstellen, hätte der Link keinen Zugriff auf die Dateiattribute.

Datenstruktur, in der die Datenblöcke der Datei verlinkt sind

man könnte natürlich Fallunterscheidungen machen, aber das Problem tritt vor allem auf, wenn man den Linkzähler miteinbezieht und den Fakt mit einbezieht, dass man

Dateisysteme unmounten kann usw.

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

Liste freier Plattenblöcke:

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

Verzeichniseinträge:

Dateiname	Erweiterung	Dateiattribute	Erster Plattenblock	Dateigröße
BRIEF	TXT	(...)	4	129 KB
EDITOR	EXE	(...)	6	101 KB
AUFGABE	DOC	(...)	15	158 KB

Aufgabe 4

3/4

a)

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

-0.5 ceil Klammern nicht vergessen, es gibt keine halben oder Viertel Zeiger bzw. Adressen

b)

1KB:

$$(10 + (1024/4) + (1024/4)^2 + (1024/4)^3) \cdot 1 = 16843018 \text{ KiB} = 16,06 \text{ GiB}$$

4KB:

$$(10 + (4096/4) + (4096/4)^2 + (4096/4)^3) \cdot 4 = 4299165736 \text{ KiB} = 3,99 \text{ TiB}$$

-0.5 Frage: " Wie groß ist die maximale Zahl aller Datenblöcke des Dateisystems, die eindeutig adressiert werden können?"
vergessen zu beantworten.