

# Betriebssysteme

## Übungsblatt 9

Micha Erkel

Felix Ruh

### Aufgabe 1

- a) Befindet man sich in dem zugehörigen I-Node und kennt man die Größe der Blöcke, so kann man die Positionen des gesuchten Bytes mit denen, in den jeweiligen Blöcken enthaltenen, vergleichen. Also im 0. direkten Zeiger sind 4,096 Byte enthalten, das Byte mit der Nummer 50,000 kann also nicht darunter sein. Vergleicht man sukzessive nun alle direkten Zeiger, wird man feststellen, dass das gesuchte Byte nicht in diesen Datenblöcken sein kann. Man geht weiter und folgt dem (ersten) einfach indirekten Link. Dort vergleicht man erneut die Blöcke bis der Block, welcher das Byte enthält gefunden ist. Nun wechselt man in diesen Block und geht ihn durch bis zu dem gesuchten Byte.

Beteiligte Zeiger:

- einfach indirekter Zeiger - Position 10 - zeigt auf einfach indirekten Block
- 2. Zeiger im einfach indirekten Block - Position 2 - zeigt auf Datenblock

Rechenweg:

Wir suchen Byte  $c = 50,001$

Subtrahiere Größe des 0.Elements des I-Node von  $c$ :  $c = c - 4,096 = 45,905$

Subtrahiere Größe des 1.Elements des I-Node von  $c$ :  $c = c - 4,096 = 41,809$

...

Subtrahiere Größe des 9.Elements des I-Node von  $c$ :  $c = c - 4,096 = 9,041$

Folge einfachen indirektem Link

Subtrahiere Größe des 0.Elements des einfach indirekten Blocks von  $c$ :  $c = c - 4,096 = 4,945$

Subtrahiere Größe des 1.Elements des einfach indirekten Blocks von  $c$ :  $c = c - 4,096 = 849$

-> das gesuchte Byte muss im 2.Elements des einfach indirekten Blocks sein

Folge dem entsprechenden Link

Durchlaufe den Block um  $c$  Bytes -> gesuchtes Byte ist gefunden

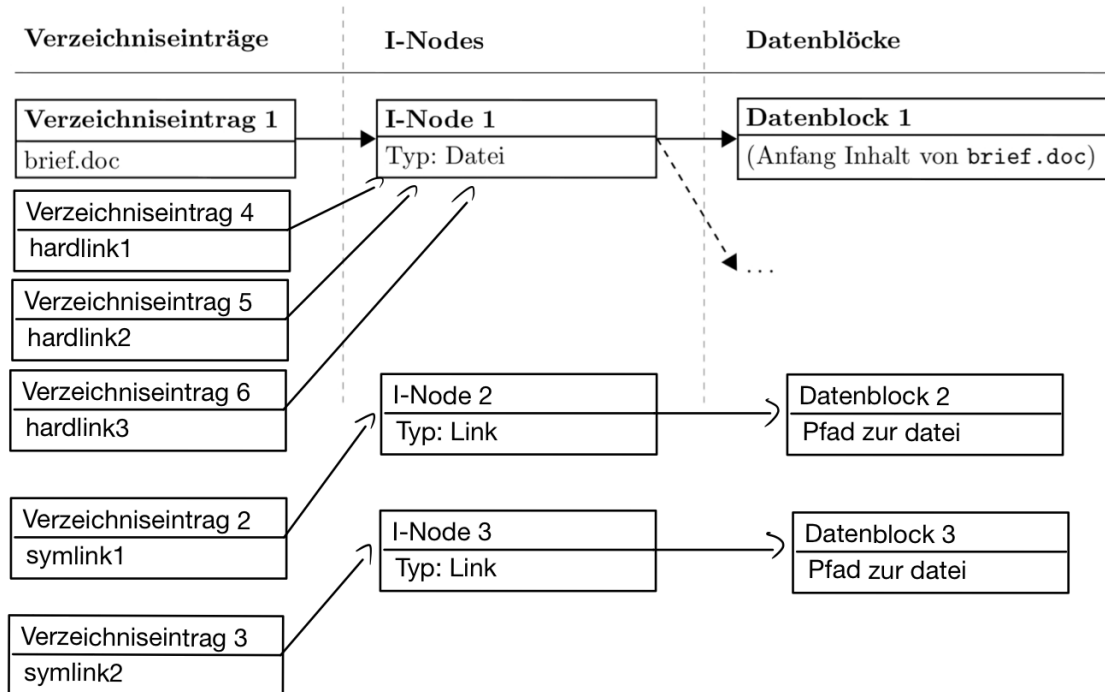
- b) Zunächst wird der Block in dem sich das gesuchte Byte befindet bzw. die Anzahl der zu überspringenden Verweise  $N(n)$  berechnet. Verfolgt man dann diesen Pfad, kann man in dem Block an Stelle  $n \bmod b$  das gesuchte Byte finden.

Anzahl der Verweise  $N(n) = \text{ceil}(\frac{\text{gesuchtes Byte } n}{\text{Blockgröße } b}) - 1$

Laufzeit:  $O(n)$

## Aufgabe 2

a) Die Skizze:



- b) Die Datei benötigt:  $3 \cdot 2^{10} \text{ Byte} = 3,072 \text{ Byte}$ , das sind  $\frac{2,5 \cdot 2^{10} \text{ Byte}}{1024 \text{ Byte}} = 2,5 \rightarrow \text{ceil}(2,5) = 3$  Blöcke  
 Die Links benötigen 2 Datenblöcke, also  $2 \cdot 1024 \text{ Byte} = 2048 \text{ Byte}$   
 Insgesamt sind es 3 I-Nodes und 6 Verzeichniseinträge
- c) Die Hardlinks haben immer die gleichen Zugriffsrechte wie die Datei selber. Verändert man also diejenigen der Datei ändern sich diejenigen der Links auf gleiche Weise mit. Allerdings ist erkennbar, dass sich die Rechte der Symlinks - die bei der Erstellung auf alles erlauben gesetzt sind - nicht verändern. Das liegt daran, dass die Hardlinks auf den gleichen I-Node wie brief.doc zeigen und die Symlinks auf einen eigenen. Dieser regelt aber den Zugriff auf die Datei mit dem Link, weshalb auch jede Form des Zugriff erlaubt ist.
- d) Die Rechte des Links lassen sich nicht ändern, es ändern sich allerdings die Rechte der Datei und somit auch der Hardlinks. Dies ist darin begründet, dass wenn sie sich ändern würden, die Möglichkeit bestünde den Link zu beschädigen und dies soll verhindert werden.

## Aufgabe 3

- a) Man muss dem Befehl noch ein '-e' hinzufügen.
- b) Der Befehl lautet beispielsweise: `kill-SIGSTOP PID`
- c) SIGSTOP: Pausiert den Prozess.  
SIGCONT: Führt einen pausierten Prozess weiter.  
SIGTERM: Prozess wird ordentlich terminiert.  
SIGKILL: Prozess wird sofort beendet.  
SIGKILL beendet den Prozess sofort, während SIGTERM den Prozess zuerst fertig werden lässt und dann beendet.
- d) Die Befehlsfolge lautet:  
`screen [*click Enter]`  
`./counter.sh [*click Enter]`  
`screen -d PID... [*click Enter]`  
  
`-log out-`  
`-log in-`  
  
`screen -r [*click Enter]`  
`(killall screen [*click Enter])`