Prof. Dr. Christoph Scholl
Dr. Tim Welschehold
Alexander Konrad
Niklas Wetzel

Freiburg, 16.12.2022

Betriebssysteme
Übungsblatt 9

Aufgabe 1 (3 + 2 Punkte)

Wahlfreier Zugriff bei I-Nodes und bei FAT

Betrachten Sie ein Dateisystem mit Blockgröße b in Byte und Zeigergröße z in Byte.

a) Angenommen, es handelt sich um ein UNIX-Dateisystem basierend auf I-Nodes. Gehen Sie von der Implementierung in `System V` aus, d.h. jeder I-Node enthält 10 direkte Zeiger sowie jeweils einen einfach-, zweifach- und dreifach indirekten Zeiger.

   Wie läuft ein wahlfreier Zugriff auf das Byte Nummer $n = 50000$ einer Datei konkret ab? Die Blockgröße sei $b = 4$ KB und die Zeigergröße sei $z = 4$ Byte, die Anzahl der direkten Zeiger ist 10. Die Nummerierung der Bytes fängt mit der Nummer 0 an. Geben Sie an, welche Zeiger daran beteiligt sind, an welcher Position in den Blöcken diese zu finden sind und wohin sie zeigen. Geben Sie den Rechenweg mit an. Bezeichnen Sie dabei das vorderste Element in einem Block als „0. Element“.

b) Wie läuft der wahlfreie Zugriff auf ein beliebiges Byte Nr. n einer Datei ab, falls statt eines I-Node-Systems nun ein FAT32-Dateisystem verwendet wird? Geben Sie einen allgemeinen Ausdruck für die Anzahl der zu verfolgenden Verweise N innerhalb der FAT in Abhängigkeit von der Blockgröße b an. Wie wird sich die Zugriffszeit in Abhängigkeit von n verhalten?

Aufgabe 2 (2 + 2 + 2 + 2 Punkte)

Hardlinks und symbolische Links

Für diese Aufgabe wird angenommen, dass Sie auf einem Dateisystem arbeiten, das auf dem in der Vorlesung vorgestellten I-Node-Konzept basiert. Die Blockgröße des Dateisystems betrage 1024 Byte. Sie haben eine Datei `/home/max/brief.doc`, die 2,5 KB an Daten enthält. Weiterhin haben Sie 2 symbolische Links („`symlink1`“ und „`symlink2`“) und 3 Hardlinks („`hardlink1`“, „`hardlink2`“ und „`hardlink3`“), die alle auf `brief.doc` verweisen.

a) Erläutern Sie anhand einer Skizze, wie diese Situation im Dateisystem realisiert ist. Ergänzen Sie dazu Abbildung 1 um die fehlenden Verzeichniseinträge, I-Nodes und Datenblöcke. Stellen Sie Verweise durch Pfeile zwischen den Elementen dar.

b) Wieviel Plattenplatz (in Anzahl Blöcken und in KB) benötigen die Datenblöcke der Datei `brief.doc` und die Datenblöcke der Links auf `brief.doc`? Wie viele I-Nodes und wie viele Verzeichniseinträge werden benötigt?

c) Die Zugriffsrechte der Datei `brief.doc` werden mit Hilfe des Befehls `chmod` geändert. Wie ändern sich dadurch die Zugriffsrechte der Hardlinks und der symbolischen Links? Probieren Sie dies in der Praxis aus und erklären Sie das beobachtete Verhalten.

d) Wenn Sie einen symbolischen Link anlegen, werden die Dateirechte als `lrwxrwxrwx` angezeigt. Versuchen Sie mit `chmod` die Rechte eines symbolischen Links zu ändern. Was passiert mit den Rechten des symbolischen Links und warum könnte dieses Verhalten sinnvoll sein?

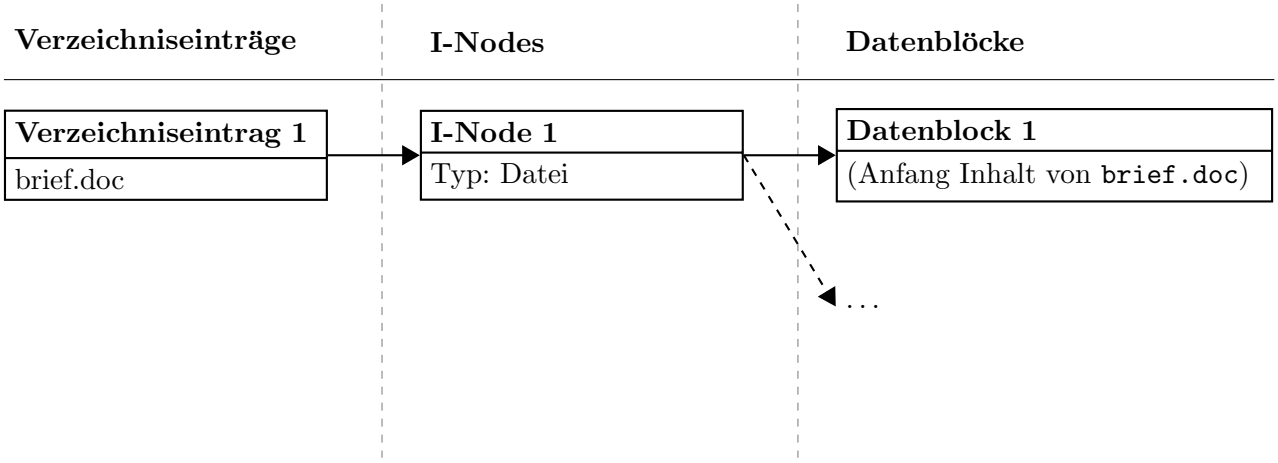

Abbildung 1: Realisierung der Dateien im Dateisystem

Aufgabe 3 (1 + 1 + 2 + 3 Punkte)

Erstellen Sie zunächst eine Datei `counter.sh` mit folgendem Inhalt:

counter.sh

```
#!/bin/bash
let x=0                          # setze Variable x auf 0
while true                       # Wiederhole endlos
do
    echo $x                      # gib Variable x aus
    let x=x+1                    # Zähle x um 1 hoch
    sleep 1                      # Warte 1 Sekunde
done
```

Das Skript implementiert einen einfachen Zähler. Machen Sie das Skript mit `chmod u+x counter.sh` ausführbar und führen Sie es mit `./counter.sh` aus.

Öffnen Sie ein weiteres Terminal oder eine zweite SSH-Verbindung und bearbeiten Sie die Aufgabe, während `counter.sh` im ersten Terminal weiterläuft.

a) Der Befehl `ps` zeigt Informationen über die derzeit laufenden Prozesse an. Ohne Optionen zeigt `ps` nur die Prozesse des aktuellen Benutzers an, die auf der aktuellen Textkonsole gestartet wurden.

   Welche Optionen muss man beim Aufruf von `ps` benutzen, um eine komplette Liste der derzeit laufenden Prozesse zu erhalten?

b) In Linux werden *Signale* verwendet, um Prozesse zu steuern und einfache Nachrichten („Daten stehen bereit“, „Fehler aufgetreten“ usw.) zwischen Prozessen auszutauschen. Dazu gehören unter anderem

   - `SIGSTOP`
   - `SIGCONT`
   - `SIGTERM`
   - `SIGKILL`

   Wie können Sie den Befehl `kill` verwenden, um diese Signale an den Prozess von `counter.sh` zu senden?

   Hinweis 1: Die Prozess-ID (PID) können Sie mit `pidof -x counter.sh` bestimmen oder in der Ausgabe von `ps` ablesen.

   Hinweis 2: Um die Manpage der *Signale* aufzurufen, verwenden Sie den Befehl `man 7 signal`.

c) Beschreiben Sie, was jedes der vier Signale aus Teilaufgabe b) bewirkt. Erklären Sie auch den Unterschied zwischen `SIGTERM` und `SIGKILL`.

d) Sie möchten `counter.sh` starten, sich vom Rechner abmelden, `counter.sh` aber trotzdem weiter laufen lassen und die Ausgabe nach dem nächsten Einloggen wieder aufrufen. Sobald das Terminal-Fenster geschlossen wird, wird jedoch auch der Prozess beendet. Eine mögliche

Lösung besteht darin, das Programm *GNU Screen* zu verwenden. Es wird mit dem Befehl `screen` gestartet. Welche Kommandos und Eingaben werden benötigt, um diese Teilaufgabe mit *GNU Screen* zu lösen?

Hinweis 1: Die englischen Begriffe für das gesuchte Verfahren heißen „detaching“ und „reattaching“.

Hinweis 2: Sollte *GNU Screen* noch nicht auf Ihrem Rechner installiert sein, können Sie das Programm mit dem Befehl `sudo apt-get install screen` installieren. Hierfür benötigen Sie Administratorrechte.

Abgabe: als PDF im Übungsportal bis 23.12.2022 um 12:00