Institut für Informatik

Arbeitsgruppe Autonome Intelligente Systeme

Freiburg, 24. Januar 2017

Systeme I
Musterlösung zu Übungsblatt 12 - Wiederholung

Aufgabe 1 (1+2+1 Punkte)

Realisierung von Dateien in Dateisystemen

In der Vorlesung wurden unter anderem „Zusammenhängende Belegung“ und „Verkettete Listen“ als Konzepte zur Realisierung von Dateien in einem Dateisystem vorgestellt.

a) Nennen Sie die grundlegenden Unterschiede in den Arbeitsweisen der Dateiverwaltung mit „zusammenhängender Belegung“ und „verketteten Listen“.

b) Welche Vor- und Nachteile haben die beiden Verfahren im Vergleich? Nennen Sie vier Stichpunkte und begründen Sie jeden Stichpunkt kurz.

c) Nennen Sie ein Anwendungsgebiet, in dem Dateisysteme mit zusammenhängender Belegung sinnvoll eingesetzt werden können und begründen Sie kurz.

Lösung:

a)
- Zusammenhängende Belegung: Alle Blöcke der Dateien sind direkt hintereinander angeordnet. [0.5]
- Verkettete Liste: Verkettete Liste von Zeigern auf Blöcke. [0.5]

b) [maximal 2 Punkte]

- Zusammenhängende Belegung
  - Vorteile: Wahlfreier Zugriff sehr effizient, da die Adresse des gesuchten Blocks einer Datei direkt aus der Position des ersten Blocks der Datei berechnet werden kann. [0.5]
  - Nachteile: Wenn Dateien gelöscht werden und eine einzelne Lücke nicht ausreicht, um eine neue, größere Datei aufzunehmen oder wenn eine Datei vergrößert werden soll, können Abschnitte mit ungenutzten Böcken entstehen (Stichwort externe Fragmentierung). [0.5]
- Verkettete Liste
  - Vorteile: Keine externe Fragmentierung, da jeder Block Teil der verketteten Liste einer Datei werden kann und somit alle Blöcke nutzbar sind. [0.5]
  - Nachteile: Problematisch beim Wahlfreien Zugriff: $n-1$ Lesezugriffe auf die Platte bei der Suche nach dem n-ten Block (Wenn im Speicher abgelegt: Komplette FAT muss im Speicher gehalten werden, auch wenn die Platte fast leer ist.) [0.5]

c) Die zusammenhängende Belegung kann überall dort effizient eingesetzt werden, wo sich Dateigrößen nicht mehr ändern können (z.B. CD-ROM, Backup). Durch die zusammenhängende Belegung können in diesen Fällen Dateien schneller gelesen werden. [0.5 Punkte für ein Szenario, 0.5 Punkte für die korrekte Begründung]

Aufgabe 2 (3+2 Punkte)

Zugriffsrechte und Links

Nehmen Sie an, Sie führen unter Linux den Befehl `ls -a -l` aus und erhalten dabei folgendes Ergebnis:

```
$ ls -a -l
 drwxr-xr-x 4 osswald staff    4096 2013-12-16 13:29 .
 drwxr-xr-x 3 root    root        0 2013-12-16 10:00 ..
 drwxr-x--x 2 osswald staff    4096 2012-02-15 14:01 meine_dateien
 drwxrwsrwx 2 osswald staff    4096 2012-05-03 16:17 gemeinsame_dateien
 -rw-r----- 3 mueller student 38400 2014-01-07 08:02 bericht.txt
 -rw-r----- 3 mueller student 38400 2014-01-07 08:02 kopie.txt
 lrwxrwxrwx 1 osswald staff      11 2014-01-07 08:04 eintrag1 -> bericht.txt
```

Der Verzeichniseintrag `kopie.txt` wurde mit `ln bericht.txt kopie.txt` erstellt, ist also ein Hardlink auf die Datei `bericht.txt`.

Für die Benutzer gelten folgende Gruppenmitgliedschaften:

Benutzer	Standardgruppe	alle Gruppenmitgliedschaften
osswald	staff	staff, hrl
mueller	student	student

a) Entscheiden Sie, ob die folgenden Aussagen richtig oder falsch sind.

	Behauptung	**richtig**	**falsch**
1.	Werden die Zugriffsrechte von `kopie.txt` geändert, so ändern sich die Zugriffsrechte von `bericht.txt` automatisch mit.	☐	☐
2.	Benutzer `osswald` bekommt eine Fehlermeldung, wenn er auf `eintrag1` lesend zugreift, z.B. mit `cat eintrag1`.	☐	☐
3.	Wird `bericht.txt` gelöscht, kann auf `eintrag1` immer noch zugegriffen werden.	☐	☐
4.	Wird `bericht.txt` gelöscht, kann auf `kopie.txt` immer noch zugegriffen werden.	☐	☐
5.	Benutzer `mueller` erstellt eine neue Datei im Ordner `gemeinsame_dateien`. Dann gehört die neue Datei der Gruppe `student`.	☐	☐
6.	Benutzer `mueller` darf die Datei `bericht.txt` löschen.	☐	☐

b) Beschreiben Sie kurz, wie Hardlinks in einem Dateisystem mit I-Nodes implementiert werden. Gehen Sie insbesondere darauf ein, welche Änderungen das Betriebssystem an I-Nodes und Datenblöcken vornimmt, wenn ein Hardlink auf eine Datei angelegt bzw. gelöscht wird und wann die entsprechenden Datenblöcke wieder als „frei“ markiert werden.

Lösung:

a) Behauptungen [0.5 für korrekte Antwort, kein Abzug für falsche/fehlende Antwort]:

	Behauptung	richtig	falsch
1.	Werden die Zugriffsrechte von `kopie.txt` geändert, so ändern sich die Zugriffsrechte von `bericht.txt` automatisch mit.	☒	☐
2.	Benutzer `osswald` bekommt eine Fehlermeldung, wenn er auf `eintrag1` lesend zugreift, z.B. mit `cat eintrag1`.	☒	☐
3.	Wird `bericht.txt` gelöscht, ist `eintrag1` immer noch zugreifbar.	☐	☒
4.	Wird `bericht.txt` gelöscht, ist `kopie.txt` immer noch zugreifbar.	☒	☐
5.	Benutzer `mueller` erstellt eine neue Datei im Ordner `gemeinsame_dateien`. Dann gehört die neue Datei der Gruppe `student`.	☐	☒
6.	Benutzer `mueller` darf die Datei `bericht.txt` löschen.	☐	☒

b) Implementierung Hardlinks:

- Jeder Hardlink stellt einen Verzeichniseintrag dar. Der Verzeichniseintrag verweist auf den *I-Node der Datei*. [0.5]
- Der I-Node selbst enthält einen Zähler, der die Anzahl der Verzeichniseinträge (Hardlinks) auf den I-Node beschreibt. [0.5]
- Beim Anlegen eines Hardlinks wird der Linkzähler inkrementiert und beim Löschen dekrementiert. [0.5]
- Der Dateiinhalt wird gelöscht (d.h., die entsprechenden Datenblöcke werden als frei markiert), wenn der Linkzähler auf 0 dekrementiert wird. [0.5]

Aufgabe 3 (1,5+1,5 Punkte)

I-Node-Dateisysteme

a) In der Vorlesung wurden I-Nodes und ihre Struktur bei dem Betriebssystem *System V* vorgestellt. Es besteht aus:

- 10 direkten Zeigern
- 1 Zeiger auf einen einfach indirekten Block
- 1 Zeiger auf einen zweifach indirekten Block
- 1 Zeiger auf einen dreifach indirekten Block

Die Blockgröße betrage 2 KiB und die Zeigergröße betrage 4 Byte.

Geben Sie den Rechenweg an, um die maximal mögliche Größe einer Datei auf diesem System zu berechnen (das Endergebnis als Zahl müssen Sie nicht ausrechnen).

b) Wie läuft ein wahlfreier Zugriff auf das Byte Nr. 20500 einer Datei bei diesem Dateisystem ab? Der entsprechende I-Node sei schon im Hauptspeicher vorhanden; die Nummerierung der Bytes fängt mit der Nummer 0 an.

Bitte geben Sie an, welche Zeiger daran beteiligt sind, an welcher Position in den Blöcken diese zu finden sind und wohin sie zeigen.

Lösung:

a) Anzahl Zeiger pro Block [0.5]:

$$\frac{2 \text{ KiB/Block}}{4 \text{ Byte/Zeiger}} = 512 \text{ Zeiger/Block}$$

Anzahl adressierte Blöcke:

- Direkte Zeiger: 10
- Zeiger in indirekt verbundendem Datenblock auf 1. Stufe: 512
- Zeiger in indirekt verbundenden Datenblöcken auf 2. Stufe: 512^2
- Zeiger in indirekt verbundenden Datenblöcken auf 3. Stufe: 512^3

Insgesamt: $10 + 512 + 512^2 + 512^3$ Zeiger [0.5], damit $(10 + 512 + 512^2 + 512^3) \cdot 2$ KiB ansprechbar [0.5].

b)
- Die zehn direkten Zeiger zeigen auf die zehn Datenblöcke, die Bytes 0 bis 20479 enthalten (10 Zeiger, jeweils Block mit 2048 Byte).
- Folge dem Zeiger auf den einfach indirekten Block [0.5], der 512 Zeiger auf Datenblöcke enthält. Der erste Zeiger [0.5] in diesem Block zeigt auf den Datenblock, der die Bytes 20480 bis 22527 enthält. Folge diesem Zeiger auf den Datenblock.
- An Position 20 (d.h. am 21. Byte) in diesem Datenblock befindet sich das Byte Nr. 20500 der Datei [0.5].

Aufgabe 4 (1+1,5+1,5+1 Punkte)

Multitasking und Prozessmodelle

a) Wie unterscheidet sich das präemptive vom nicht-präemptiven Prozessmodell?

b) In der Vorlesung haben Sie fünf Prozesszustände für Prozesse im Hauptspeicher kennengelernt. Tragen Sie diese fünf Zustände in die Kreise der Abbildung 1 ein.

c) Abbildung 1 enthält zudem Pfeile, die die Übergänge von Zuständen beschreiben. Beschriften Sie diese Pfeile entsprechend dem präemptiven Prozessmodell.

d) Angenommen, ein rechenbereiter Prozess bekommt bei einem Prozesswechsel die CPU zugeteilt. Wie wird sichergestellt, dass die CPU das Programm des Prozesses an der richtigen Stelle fortsetzt?

Lösung:

a) Beim nicht präemptiven Prozessmodell kann das Betriebsystem dem Prozess die CPU nicht entziehen und ist deshalb auf die Kooperation des Prozesses angewiesen [0.5]. Beim präemptiven Prozessmodell kann die CPU einem Prozess entzogen werden [0.5].

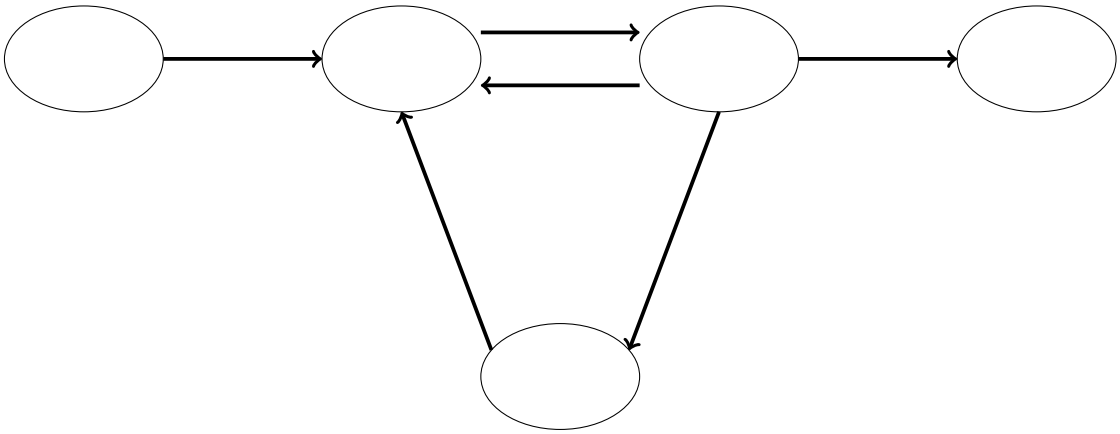

Abbildung 1: Prozessmodell mit fünf Zuständen

b) Siehe Abbildung 2 [1.5 insgesamt; 0.5 Abzug für jeden falschen Zustand]

c) Siehe Abbildung 2 [1.5 insgesamt; 0.5 Abzug für jeden falschen Übergang]

d) Das Betriebssystem speichert den Wert des Program Counter (Befehlszähler) für den Prozess. Diese Information befindet sich im Activation Record in der Prozesstabelle [0.5]. Bei der Zuteilung wird der Wert des Befehlszählers in das entsprechende CPU-Register geschrieben und die Ausführung von dem entsprechenden Befehl aus fortgesetzt [0.5].

Hinweis zur Korrektur: bei c) und d) sind auch Synonyme und Begriffe mit ähnlicher Bedeutung ok.

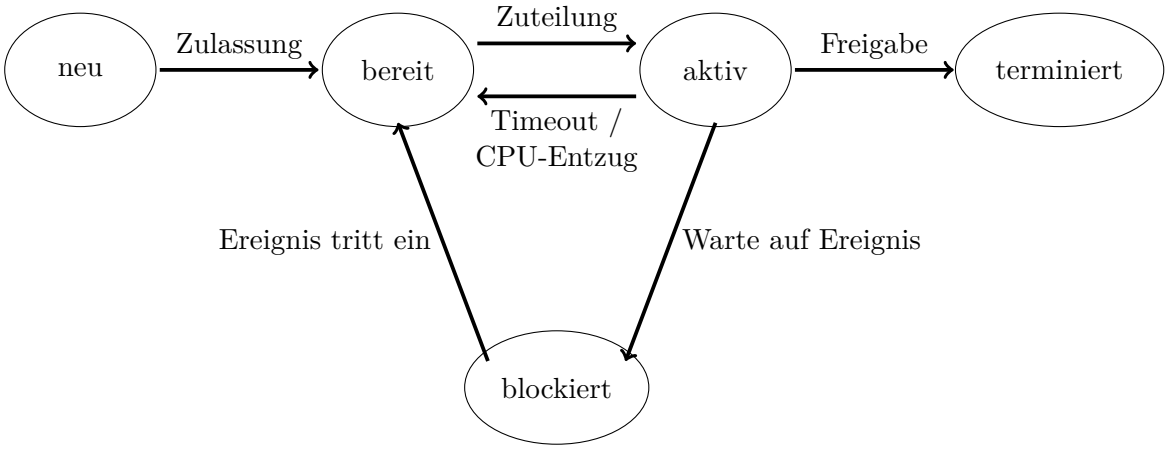

Abbildung 2: Prozessmodell mit fünf Zuständen

Abgabe: Als PDF-Datei im Ilias bis zum 30. Januar 2017, 23:59 Uhr