
Übungsblatt 6: Weitere Aufgaben

1 Welche Aufgaben hat der *Buffer Cache* in Unix?

- Bereitstellen einer byteorientierten Schnittstelle zu Dateien (obwohl auf die Platte nur blockorientiert zugegriffen werden kann).
- Caching-Funktionalität (u. U. Mehrfachzugriff möglich).

2 Was geschieht durch den Systemaufruf `mount()` in etwa?

Einhängen eines Dateisystems (auf einer (virtuellen) Platte) in den Gesamt-Dateibaum eines UNIX-Systems (am Mount Point).

3 Welche Vorteile bietet es, Dateien mit dem Unix-Systemaufruf `mmap()` in den virtuellen Adressraum eines Prozesses abzubilden?

Direkter Zugriff auf die Inhalte einer Datei, ohne über die read/write-Schnittstelle des Dateisystems gehen zu müssen. Kein Umkopieren vom Buffer Cache in den virtuellen Adreßraum des Prozesses erforderlich.

4 Wie ist eine Platte intern organisiert? Wie wirkt sich dies auf den Informationszugriff aus? Wie geht das *Unix Fast File System* damit um?

Organisation einer Platte: Mehrere Oberflächen mit Spuren und Sektoren (jeder Sektor enthält einen Plattenblock). Armbewegungen sind relativ teuer.

UNIX Fast File System: Einführung von Zylindergruppen.

- Datei und zugehöriger Inode in derselben Zylindergruppe.
- Directory und enthaltene Plain Files in derselben Zylindergruppe.
- Unterdirectories in anderer Zylindergruppe.

5 Welche Vorteile bietet eine vereinheitlichte Betriebssystemschnittstelle zum Zugriff auf Geräte? Wie sieht sie in Unix in etwa aus?

Geräteschnittstelle:

- gleiche Namensgebung wie bei „normalen“ Dateien (Einhängen in Verzeichnisbaum)
- Verdecken von Hardwareeigenschaften
- einfache Ein-/Ausgabeumlenkung

Bereitstellung von

- open()
- close()
- read()
- write()
- Operationen wie bei „normalen“ Dateien.

6 Was ist ein Gerätetreiber, was ein Geräte-Controller? Welche Aufgaben haben sie?

Gerätetreiber: Betriebssystemcode zum Absetzen von Aufträgen an das Gerät, zur Interruptbehandlung, zur Verwaltung von Warteschlangen.

Gerätecontroller: Karte/Hardware (in der Regel auch kleiner Prozessor) zum Absetzen von Steuersignalen an Gerät und Bereitstellung von Pufferbereichen. Gerätereister darin nehmen Aufträge auf.

7 Warum erfolgt der Zugriff auf Geräte häufig über Warteschlangen? Wozu besitzen diese in der Regel eine High Water Mark bzw. eine Low Water Mark?

Durch Warteschlangen bessere Entkopplung von Prozessen und Geräten möglich, d. h. mehrere Aufträge/Ergebnisse können anstehen. Treiber kann bei Fertigmeldung selbsttätig nächsten Auftrag absetzen.

High Water Mark: Maximale Puffergröße ausgeschöpft (nicht mehr Platz für die Warteschlange reserviert).

Low Water Mark: Nachfüllen vorzeitig anstoßen. Effizienzsteigerung.

8 Worin unterscheidet sich *DMA (Direct Memory Access)* von *Programmed I/O*?

DMA: Plattencontroller kopiert gelesene Blöcke selbst in Hauptspeicher (und nicht die CPU, genauer: der Interrupthandler innerhalb des Gerätetreibers nach Fertigmeldung der Platte).

Programmed I/O: Das Umkopieren von Daten wird über (in den Hauptspeicher eingeblendete) Gerätereister gesteuert.

9 Warum werden Terminal-Treiber in Unix parametrisiert? Nenne typische Parameter.

Terminal-Treiber werden parametrisiert, weil es so viele verschiedene Terminals und Betriebsmodi gibt (z. B. Raw Mode vs. Cooked Mode, Übertragungsrate, Echoverhalten, Signale abfangen oder nicht, Zeilenende-Zeichen, ...).