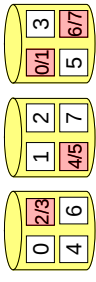


## 2 Einsatz mehrere redundanter Platten (5)

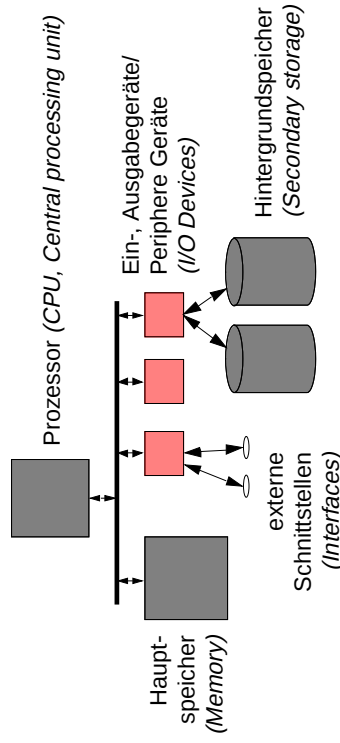
- Verstreuter Paritätsblock (RAID 5)
  - ◆ Paritätsblock wird über alle Platten verstreut
- 
- ◆ zusätzliche Belastung durch Schreiben des Paritätsblocks wird auf alle Platten verteilt
  - ◆ heute gängigstes Verfahren redundanter Platten
  - ◆ Vor- und Nachteile wie RAID 4
- ▲ Problem
- ◆ fehlerhafter Paritätsblock zerstört mehrere Blöcke, falls eine Platte ausfällt

SP I

F.57

## G Ein- und Ausgabe

- Einordnung

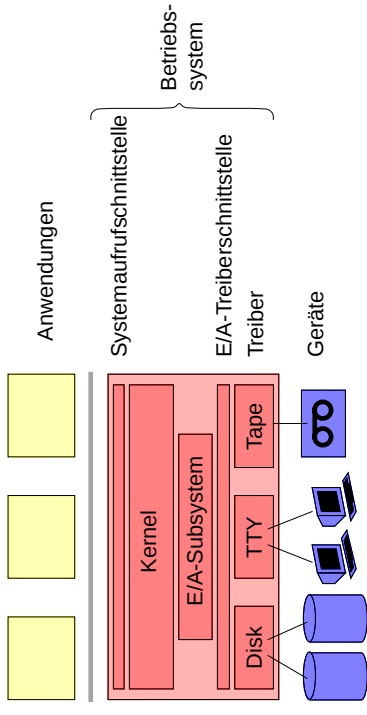


SP I

G.1

## G.1 Gerätezugang und Treiber

- Schichtung der Systemsoftware bis zum Gerät



Nach Vahalia, 1996

SP I

G.2

## 1 Geräterepäsentation in UNIX

- Periphere Geräte werden als Spezialdateien repräsentiert
  - ◆ Geräte können wie Dateien mit Lese- und Schreiboperationen angesprochen werden
  - ◆ Öffnen der Spezialdateien schafft eine Verbindung zum Gerät, die durch einen Treiber hergestellt wird
  - ◆ direkter Durchgriff vom Anwender auf den Treiber
- Blockorientierte Spezialdateien
  - ◆ Plattenlaufwerke, Bandlaufwerke, Floppy Disks, CD-ROMs
- Zeichenorientierte Spezialdateien
  - ◆ Serielle Schnittstellen, Drucker, Audiokanäle etc.
  - ◆ blockorientierte Geräte haben meist auch eine zusätzliche zeichenorientierte Repräsentation

SP I

G.3

## 1 Geräterepräsentation in UNIX (2)

- Eindeutige Beschreibung der Geräte durch ein Tupel:  
( Gerätetyp, *Major number*, *Minor number* )
- ◆ Gerätetyp: Block device, Character device
- ◆ Major number: Auswahlnummer für einen Treiber
- ◆ Minor number: Auswahl eines Gerätes innerhalb eines Treibers

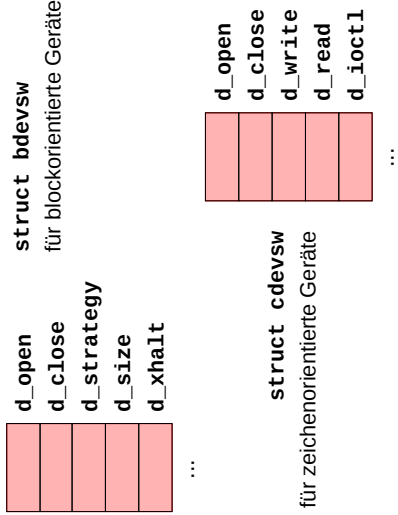
## 1 Geräterepräsentation in UNIX (3)

- Beispiel eines Kataloglisting von `/dev` (Ausschnitt)

```
crw----- 1 fzhauk 108, 0 Oct 16 1996 audio
crw----- 1 fzhauk 108,128 Oct 16 1996 audioc1
crw-rw-rw- 1 root 21, 0 May 3 1996 conslog
brw-rw-rw- 1 root 36, 2 Oct 16 1996 fd0
crw----- 1 fzhauk 17, 0 Oct 16 1996 mouse
crw-rw-rw- 1 root 13, 2 Jan 13 09:09 null
crw-rw-rw- 1 root 36, 2 Jul 2 1997 rfd0
crw-r----- 1 root 32, 0 Oct 16 1996 rsd3a
crw-r----- 1 root 32, 2 Oct 16 1996 rsd3c
brw-r----- 1 root 32, 0 Oct 16 1996 sd3a
brw-r----- 1 root 32, 1 Oct 16 1996 sd3b
brw-r----- 1 root 32, 2 Oct 16 1996 sd3c
crw-rw-rw- 1 root 22, 0 Sep 19 09:11 tty
crw-rw-rw- 1 root 29, 0 Oct 16 1996 ttya
crw-rw-rw- 1 root 29, 1 Oct 16 1996 ttyb
```

## 1 Geräterepräsentation in UNIX (4)

- Interne Treiberschnittstelle
- ◆ Vektor von Funktionszeigern pro Treiber (Major number):



## 1 Geräterepräsentation in UNIX (5)

- Funktionen eines Block device-Treibers

- ◆ `d_open`: Öffnen des Gerätes
- ◆ `d_close`: Schließen des Gerätes
- ◆ `d_strategy`: Abgeben von Lese- und Schreibaufträgen auf Blockbasis
- ◆ `d_size`: Ermitteln der Gerätegröße (z.B. Partitions- oder Plattengröße)
- ◆ `d_xhalt`: Abschalten des Gerätes
- ◆ u.a.

- Funktionen eines Character device-Treibers

- ◆ `d_open`, `d_close`: Öffnen und Schließen des Gerätes
- ◆ `d_read`, `d_write`: Lesen und Schreiben von Zeichen
- ◆ `d_ioctl`: generische Kontrolloperation
- ◆ u.a.

## 1 Geräterepräsentation in UNIX (6)

- Felder für den Aufruf von Treibern (`bdevsw[]` und `cdevsw[]`)



### Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

G-InOut.doc: 1998-01-20 10:09

Reproduzieren oder Verwenden dieser Vorlesung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg. Jede Weiterverbreitung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg. Jede Weiterverbreitung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg.

G.8

## G.2 Disk Scheduling

- Treiber hat in der Regel mehrere Aufträge in seiner Warteschlange
  - ◆ Warteschlange wird z.B. in UNIX durch Aufruf der Funktion `d_strategy()` gefüllt
  - ◆ eine bestimmte Ordnung der Ausführung kann Effizienz steigern
  - ◆ Zusammensetzung der Bearbeitungszeit eines Auftrags:
    - Positionierzeit: abhängig von der aktuellen Stellung des Plattenarms
    - Latenzzeit: Zeit bis der Magnetkopf den Sektor bestreicht
    - Übertragungszeit: Zeit zur Übertragung der eigentlichen Daten
- ★ Ansatzpunkt: Positionierzeit

### Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

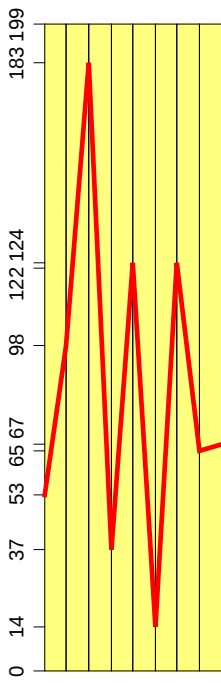
G-InOut.doc: 1998-01-20 10:09

Reproduzieren oder Verwenden dieser Vorlesung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg. Jede Weiterverbreitung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg. Jede Weiterverbreitung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg.

G.9

## 1 FCFS Scheduling

- Bearbeitung gemäß Anknüpfung des Auftrags
- ◆ Referenzfolge (Folge von Zylindernummern):  
98, 183, 37, 122, 14, 124, 65, 67
- ◆ Aktueller Zylinder: 53



- ◆ Gesamtzahl der Spurwechsel: 640
- ◆ Weite Bewegungen des Schwenkarms: mittlere Bearbeitungsdauer lang

### Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

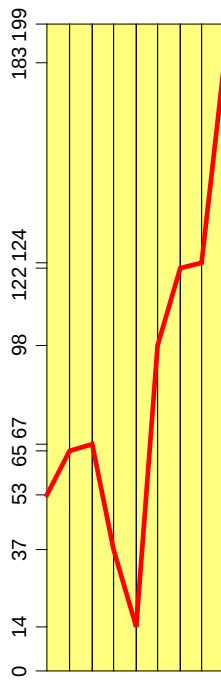
G-InOut.doc: 1998-01-20 10:09

Reproduzieren oder Verwenden dieser Vorlesung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg. Jede Weiterverbreitung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg. Jede Weiterverbreitung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg.

G.10

## 2 SSTF Scheduling

- Es wird der Auftrag mit der kürzesten Positionierzeit vorgezogen (*Shortest seek time first*)
- ◆ Gleiche Referenzfolge  
(Annahme: Positionierzeit proportional zum Zylinderabstand)



- ◆ Gesamtzahl von Spurwechseln: 236
- ◆ ähnlich wie SJF kann auch SSTF zur Aushungerung führen
- ◆ noch nicht optimal

### Systemprogrammierung I

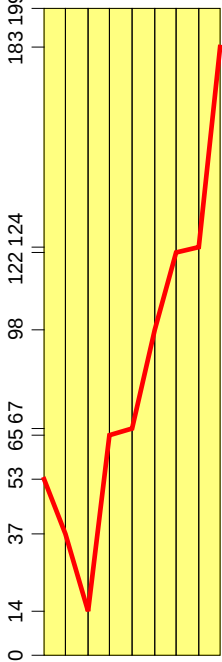
© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

G-InOut.doc: 1998-01-20 10:09

Reproduzieren oder Verwenden dieser Vorlesung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg. Jede Weiterverbreitung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg. Jede Weiterverbreitung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg.

G.11

### 3 SCAN Scheduling

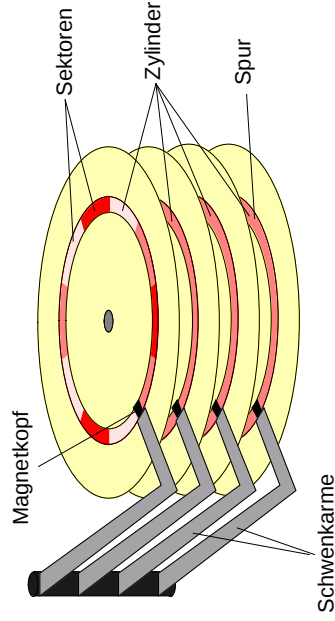
- Bewegung des Plattenarm in eine Richtung bis keine Aufträge mehr vorhanden sind (Fahrstuhlstrategie)
- ◆ Gleiche Referenzfolge (Annahme: bisherige Kopfbewegung Richtung 0)  

- ◆ Gesamtzahl der Spurwechsel: 208
- ◆ Neue Aufträge werden miterledigt ohne zusätzliche Positionierzeit und ohne mögliche Aushungerung
- ◆ Variante C-SCAN (*Circular SCAN*): Bewegung nur in eine Richtung

SP I

G.12

### G.3 Plattenphysik

- Aufbau einer Platte



SP I

G.13

### G.3 Plattenphysik (2)

- Datenblätter zweier Beispielpplatten

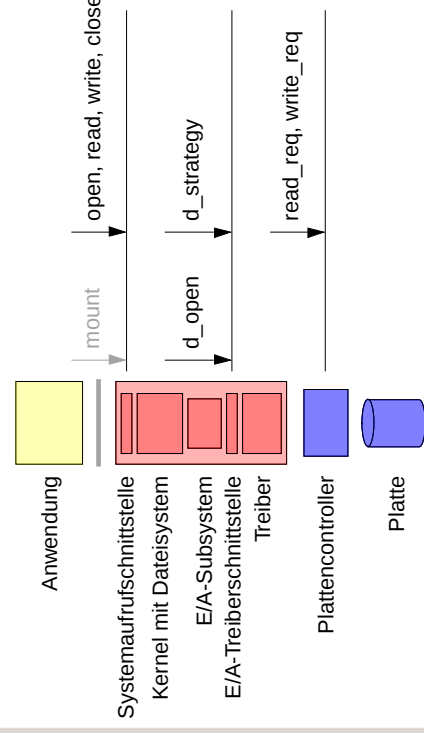
|                       |                |               |
|-----------------------|----------------|---------------|
| Plattentyp            | Fujitsu M2344K | Fujitsu M2652 |
| Kapazität             | 690 MB         | 2,0 GB        |
| Zylinderzahl          | 624            | 1893          |
| Spuren pro Zylinder   | 27             | 20            |
| Sektoren pro Spur     | 1–128          |               |
| Positionierzeiten     |                |               |
| Spur zu Spur          | 4 ms           | 2 ms          |
| mittlere              | 16 ms          | 11 ms         |
| maximale              | 33 ms          | 22 ms         |
| Transferate           | 2.458 MB/s     | 4.750 MB/s    |
| Rotationsgeschw.      | 3.600 U/min    | 5.400 U/min   |
| eine Plattenumdrehung | 17 ms          | 11 ms         |

SP I

G.14

### G.4 Plattentreiber

- Software und Hardware zwischen Anwender und Platte

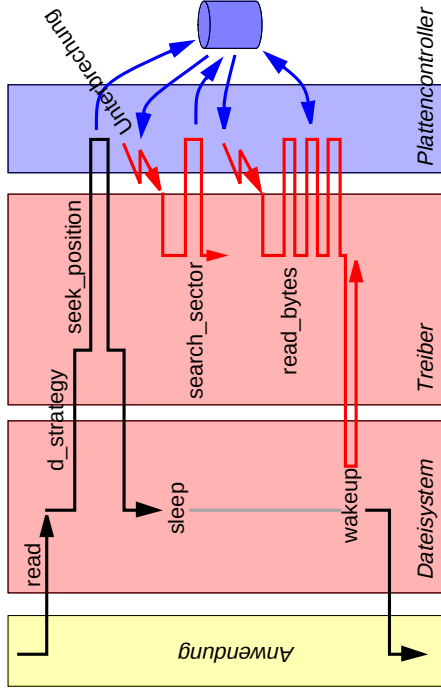


SP I

G.15

# 1 Einfacher Treiber

- ## Ablauf eines Leseauftrags

**SP I** Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1991

G-InOut.doc 1998-01-20 10.09

Stammesaktionen (einfache Art und Weise) verursachen oft kleine Unterschiede, außer im Falle von starken, nicht-linearen Transformationen des Kontexts.

## Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

G-InOut.doc 1998-01-20 10.09

Beim Anblick der Versammlung, deren Urteil zu mir auf der Türe stand, an die Handlung der Zerstörung der Kirche.

## 1 Einfacher Plattentreiber (2)

- ◆ Anwendung führt read() Systemaufruf aus.
- ◆ Dateisystem prüft, ob entsprechender Block im Speicher vorhanden.
- ◆ Falls der Block nicht vorhanden ist, wird ein Speicherplatz bereitgestellt und d\_strategy im entsprechenden Treiber aufgerufen.
- ◆ Die Ausführung von d\_strategy stößt Plattenpositionierung an.
- ◆ Die Anwendung blockiert sich im Kern. System kann andere Prozesse ablaufen lassen.
- ◆ Plattencontroller meldet sich bei erfolgter Positionierung durch eine Unterbrechung.
- ◆ Unterbrechungsbehandlung stößt Sektorsuche an.
- ◆ In erneuter Unterbrechung nach gefundenem Sektor werden die Daten im Pollingbetrieb eingelesen.
- ◆ Schließlich wird der Anwendungsprozess wieder aufgeweckt (in den Zustand bereit überführt).

## Systemprogrammierung I

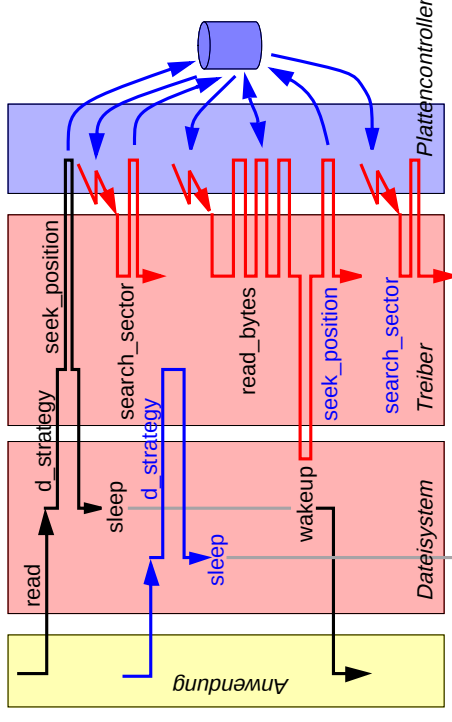
© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1991

G-InOut.doc 1998-01-20 10.09

Beurteilung dieser Art oder Verwerfung dieser Urteilung auf der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors

# 1 Einfacher Plattentreiber (3)

- Ablauf mehrerer Leseaufträge**



1 ds

© Franz J. Hauck, Universität Erlangen

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

G-InOut.doc 1998-01-20 10.09

Beim Anblick der Versammlung, deren Urteil zu mir auf der Türe stand, an die Handlung der Zerstörung der Kirche.

# 1 Einfacher Plattentreiber

- ◆ Unterbrechungsbehandlung ist auch für weitere Aufträge zuständig
- ◆ Ist der Auftrag abgeschlossen muß die Unterbrechungsbehandlung den nächsten Auftrag auswählen und aufsetzen, da der zugehörige Prozeß bereits blockiert ist.
- ◆ Die Unterbrechungen laufender Aufträge sorgen für die Abwicklung der folgenden Aufträge.

15

© Franz J. Hauck, Universität Erlang

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

Reproduktionen jeder Art oder Vervielfältigung dieser Unterlagen, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

## 2 Treiber mit DMA

- DMA (*Direct memory access*) erlaubt Einlesen und Schreiben ohne Prozessorbeteiligung
- ◆ DMA Controller erhält verschiedene Parameter:
    - die Hauptspeicheradresse zum Abspeichern bzw. Auslesen eines Plattenblocks
    - die Adresse des Plattencontrollers zum Abholen bzw. Abgeben der Daten
    - die Länge der zu transferierenden Daten
  - ◆ DMA Controller löst bei Fertigstellung eine Unterbrechung aus
- Vorteile
- ◆ Prozessor muß Zeichen eines Plattenblocks nicht selbst abnehmen
  - ◆ Plattentransferzeit kann zum Ablauf anderer Prozesse genutzt werden

★ Vorteile

SPI

# Systemprogrammierung I

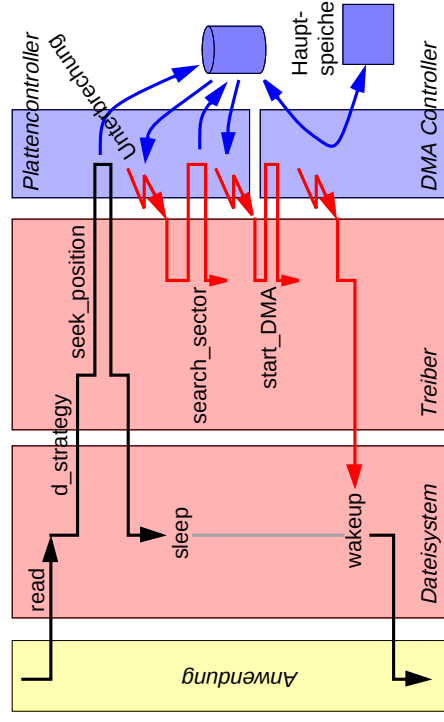
360

G-InOut.doc 1998-01-20 10.09

G.20

Beurteilungen (siehe auch die Vorwörter dieser Unterlagen) über möglicherweise bestehende Zusammenhänge zwischen den Faktoren

## 2 Treiber mit DMA (2)



SPI

## Systemprogrammierung I

360

G-InOut.doc 1998-01-20 10.09

G21

Beurteilung jeder Art oder Verwendung dieser Urkunden an der Universität zu Berlin bedarf der Zustimmung des Senats.

## 2 Treiber mit DMA (3)

- ◆ Große Systeme mit mehreren DMA-Kanälen und vielen Platten
  - ◆ es muß ein freier DMA-Kanal gesucht werden und evtl. auf einen freien gewartet werden bevor der Auftrag ausgeführt werden kann
  - ◆ Anforderung kann parallel zur Plattenpositionierung erfolgen
- ### Mainframe-Systeme
- ◆ Steuereinheit faßt mehrere Platten zu einem Gerät zusammen
  - ◆ mehrere Steuereinheiten hängen an einem Kanal zum Hauptspeicher
  - ◆ zum Zugriff auf die eigentliche Platte muß erst der Kanal und dann die Steuereinheit belegt werden (Teilwegbelegung)
- ### DMA und Caching
- ◆ heutige Prozessoren arbeiten mit Datencaches
  - ◆ DMA läuft am Cache vorbei: Betriebssystem muß vor dem Aufsetzen von DMA-Transfers Caches zurückschreiben und invalidieren

SPI

# Systemprogrammierung I

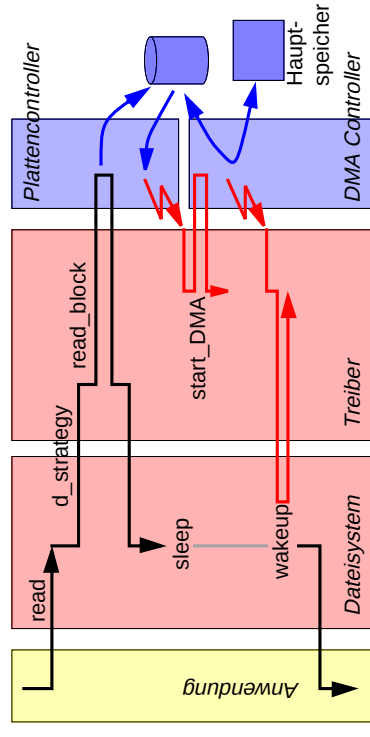
360

G-InOut.doc 1998-01-20 10.09

*Wissenschaftliche Institute, Arbeitskreise, Verbände, Vereine, etc.* (Büro, etc.) (die verschiedenen in der Adresse)

### 3 Treiber für intelligente Platte

- Intelligente Platten besitzen eigenen Prozessor für
  - ◆ das Umsortieren von Aufträgen (interne Plattenstrategie)
  - ◆ eigene Bad block-Erkennung, etc.



SPI

## Systemprogrammierung I

360

G-InOut.doc 1998-01-20 10.09

Rechtsabteilung jeder Art oder Veranstaltung dieser Unterklasse, zu ihrer im Lehrzwecke an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

## G.5 Treiber für serielle Schnittstellen

- Einsatz serieller Schnittstellen (z.B. RS-232)
  - ◆ Terminals
  - ◆ Drucker
  - ◆ Modems
- Datenübertragung
  - ◆ zeichenweise seriell (z.B. Startbit, Datenbits, Stopbits)
  - ◆ getaktet in bestimmter Geschwindigkeit (Baudrate, z.B. 38.400 Bit/s) (im Vergleich zu Platten relativ langsam)
  - ◆ Flußkontrolle (d.h. Empfänger kann Datenfluß bremsen)
  - ◆ bidirektional
- Treiber
  - ◆ zeichenorientiertes Gerät
  - ◆ vom Prinzip her ähnlich dem Plattentreiber

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

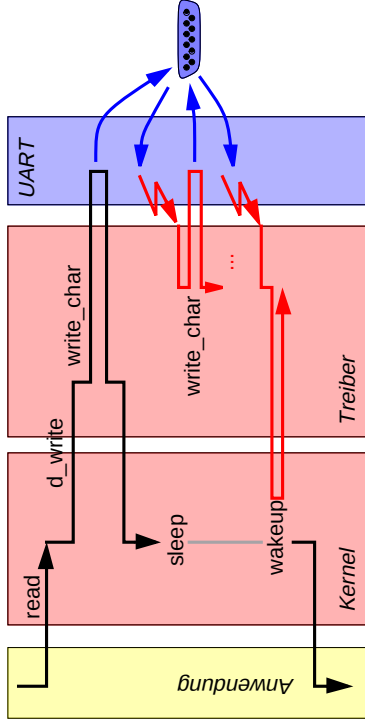
G-inOut.doc 1998-01-20 10:09

Reproduktion oder die auch Verwendung dieser Vorlesung ist in der Universität Erlangen-Nürnberg gestattet. Bei der Reproduktion des Dokuments.

G.24

## 1 Einfacher TTY-Treiber

- TTY-Treiber (*Teletype*, Fernschreiber) und der Ablauf eines Schreibaufrufs



- ◆ UART = Universal asynchronous receiver / transmitter

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

G-inOut.doc 1998-01-20 10:09

Reproduktion oder die auch Verwendung dieser Vorlesung ist in der Universität Erlangen-Nürnberg gestattet. Bei der Reproduktion des Dokuments.

G.25

## 1 Einfacher TTY-Treiber

- Enger Zusammenhang zwischen Ein- und Ausgabe
  - ◆ Echofunktion (getippte Zeichen werden angezeigt)
    - eingelesene Zeichen werden gleich wieder ausgegeben
  - ◆ Flußkontrolle (bestimmtes Zeichen in der Eingabe hält Ausgabe an: ^S)
    - wird ^S eingelesen wird Ausgabe angehalten bis ^Q eingelesen wird
- Zeilenorientierte Treiber
  - ◆ Anwendung will Zeichen zeilenweise, z.B. Shell
  - ◆ Treiber blockiert Prozeß bis Zeilenende erkannt
  - ◆ Treiber erlaubt das Editieren der Zeile (Backspace, etc.)
- Signale
  - ◆ bestimmte Zeichen lösen Signale an korrespondierende Prozesse aus

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

G-inOut.doc 1998-01-20 10:09

Reproduktion oder die auch Verwendung dieser Vorlesung ist in der Universität Erlangen-Nürnberg gestattet. Bei der Reproduktion des Dokuments.

G.26

## 2 TTY-Treiber in UNIX

- Konfigurierbar
  - ◆ Repräsentation einer seriellen Schnittstellen als zeichenorientiertes Gerät
  - ◆ durch Aufruf von ioctl kann Treiber konfiguriert werden
    - `int ioctl( int fildes, int request, /* arg */ );`
  - ◆ Kommando zum Lesen der Konfiguration: Übergabe einer Strukturadresse
    - `struct termios t;`
    - `ioctl( fd, TCGETS, &t );`
  - ◆ Kommando zum Schreiben einer Konfiguration:
    - `ioctl( fd, TCSETS, &t );`
  - ◆ Struktur enthält Bitfelder für verschiedene Einstellungen
  - ◆ Bitmasken sind als Makros verfügbar
  - ◆ näheres: „man termios“ und „man ioctl“

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

G-inOut.doc 1998-01-20 10:09

Reproduktion oder die auch Verwendung dieser Vorlesung ist in der Universität Erlangen-Nürnberg gestattet. Bei der Reproduktion des Dokuments.

G.27

## 2 TTY-Treiber in UNIX (2)

■ Beispiele für Einstellungsmöglichkeiten:

IGNBRK Ignore break condition.  
IGNPAR Ignore characters with parity errors.  
ISTRIP Strip character.  
ICRNL Map CR to NL on input.  
IXON Enable start/stop output control.  
IMAXBEL Echo BEL on input line too long.  
ISIG Enable signals.  
ICANON Canonical input (erase and kill processing).  
ECHO Enable echo.  
ECHOE Echo erase character as BS-SP-BS.  
ECHOX Echo NL after kill character.  
ECHONL Echo NL.  
ECHOCTL Echo control characters as ^char, delete as ^?.  
ECHOKE BS-SP-BS erase entire line on line kill.  
PENDIN Retype pending input at next read or input character.

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

G-inOut.doc 1998-01-20 10:09

G.28

Reproduktion oder andere Verwendung dieser Vorlesung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg strafbar.

## 2 TTY-Treiber in UNIX (3)

■ Baudrateneinstellung:

B0 Hang up  
B50 50 baud  
B75 75 baud  
B110 110 baud  
B134 134 baud  
B150 150 baud  
B200 200 baud  
B300 300 baud  
B600 600 baud  
B1200 1200 baud  
B1800 1800 baud  
B2400 2400 baud  
B4800 4800 baud  
B9600 9600 baud  
B19200 19200 baud  
B38400 38400 baud  
B57600 57600 baud

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

G-inOut.doc 1998-01-20 10:09

G.29

Reproduktion oder andere Verwendung dieser Vorlesung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg strafbar.

## 2 TTY-Treiber in UNIX (4)

■ Weitere Einstellungsmöglichkeiten:

CRTSCTS Enable outbound hardware flow control  
OPOST Post-process output.  
ONLCR Map NL to CR-NL on output.  
ONOCR No CR output at column 0.  
NLDLY Select newLine delays:  
NL0  
NL1  
CRDLY Select carriage-return delays:  
CR0  
CR1  
CR2  
CR3  
TABDLY Select horizontal tab delays:  
TAB0 or tab expansion:  
TAB1  
TAB2  
TAB3 Expand tabs to spaces.  
XTABS Expand tabs to spaces.

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

G-inOut.doc 1998-01-20 10:09

G.30

Reproduktion oder andere Verwendung dieser Vorlesung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg strafbar.

## 2 TTY-Treiber in UNIX (5)

■ Pseudo-TTY-Treiber (PTY)

- ◆ keine echte serielle Schnittstelle vorhanden
- ◆ dient als gewohnte Schnittstelle von Anwendungsprogrammen
- ◆ Einsatz beispielsweise bei einem Fenstersystem (xterm-Programm)
  - xterm-Programm bedient die Systemseite eines PTTY
  - Shell und Anwendungsprogramme sehen xterm-Fenster wie eine serielle Schnittstelle mit Line editing, Einstellungsmöglichkeiten, Signalezustellung etc.

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

G-inOut.doc 1998-01-20 10:09

G.31

Reproduktion oder andere Verwendung dieser Vorlesung ist ohne schriftliche Genehmigung der Universität Erlangen-Nürnberg strafbar.

## G.6 Bildschirmtreiber

- **Bildspeicher**
  - ◆ zeichenorientiert
  - ◆ pixelorientiert
- **Aufgaben des Treibers**
  - ◆ Bereitstellen von Graphikprimitiven (z.B. Ausgabe von Text, Zeichnen von Rechtecken, etc.)
  - ◆ Ansprechen von Graphikprozessoren (schnelle Verschiebeoperationen, komplexe Zeichenoperationen, 3D Rendering, Textures)
  - ◆ Einblenden des Bildspeichers in Anwendungsprogramme (z.B. X11-Server)
- **Bildspeicher**
  - ◆ spezieller Speicher, der den Bildschirminhalt repräsentiert
  - ◆ Dual ported RAM (Videochip und Prozessor können gleichzeitig zugreifen)

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

G-inOut.doc: 1998-01-20 10:09

G.32

Reproduktion oder andere Verwendung dieser Vorlesungs- oder Lehrunterlagen ist ohne schriftliche Genehmigung ausdrücklich untersagt.

## G.7 Netzwerktreiber

- **Beispiel: Ethernet**
  - ◆ schneller serieller Bus mit CSMA/CD  
(*Carrier sense media access / Collision detect*)  
zu deutsch: es wird dann gesendet, wenn nicht gerade jemand anderes sendet; Kollisionen werden erkannt und aufgelöst
  - ◆ spezieller Netzwerkchip
    - implementiert unterstes Kommunikationsprotokoll
    - erkennt eintreffende Pakete
- **Netzwerktreiber**
  - ◆ wird von höheren Protokollen innerhalb des Betriebssystems angesprochen, z.B. von der IP-Schicht

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

G-inOut.doc: 1998-01-20 10:09

G.33

Reproduktion oder andere Verwendung dieser Vorlesungs- oder Lehrunterlagen ist ohne schriftliche Genehmigung ausdrücklich untersagt.

## G.7 Netzwerktreiber (2)

- **Senden**
  - ◆ Treiber übergibt dem Netzwerkchip eine Datenstruktur mit den notwendigen Informationen: Sendeadresse, Adresse und Länge von Datenpuffern
  - ◆ Netzwerkchip löst Unterbrechung bei erfolgtem Senden aus
- **Empfangen**
  - ◆ Treiber übergibt dem Netzwerkchip eine Datenstruktur mit Adressen von freien Arbeitspuffern
  - ◆ erkennt der Netzwerkchip ein Paket (für die eigene Adresse), füllt er das Paket in einen freien Puffer
  - ◆ der Puffer wird in eine Liste von empfangenen Paketen eingehängt und eine Unterbrechung ausgelöst
  - ◆ Treiber kann die empfangenen Pakete aushängen

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

G-inOut.doc: 1998-01-20 10:09

G.34

Reproduktion oder andere Verwendung dieser Vorlesungs- oder Lehrunterlagen ist ohne schriftliche Genehmigung ausdrücklich untersagt.

## G.7 Netzwerktreiber (3)

- **Übertragung der Daten erfolgt durch DMA**
  - ◆ evtl. direkt durch den Netzwerkchip
- **Intelligente und nicht-intelligente Netzwerkhardware**
  - ◆ intelligente Hardware: kann evtl. auch höhere Protokolle, Filterung etc.
  - ◆ nicht-intelligente Hardware: benötigt mehr Unterstützung durch den Treiber (Prozessor)

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

G-inOut.doc: 1998-01-20 10:09

G.35

Reproduktion oder andere Verwendung dieser Vorlesungs- oder Lehrunterlagen ist ohne schriftliche Genehmigung ausdrücklich untersagt.

## G.8 Andere Geräte

### ■ Uhr

- ◆ Hardwareuhren (z.B. DCF 77, GPS Empfänger)
- ◆ Systemuhr fast immer in Software (wird mit Hardwareuhren synchronisiert)
- ◆ UNIX: `gettimeofday`, `settimeofday`
  - vier Intervalltimer pro Prozess: Signal SIGALRM nach Ablauf
  - Ablauf konfigurierbar:
    - Realzeit, Virtuelle Zeit, Virtuelle Zeit (einschl. Systemzeit des Prozesses)

### ■ Bandlaufwerk

- ◆ zeichenorientiertes Gerät
- ◆ Spuloperationen durch `d_ioctl` realisiert

### SP I

#### Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

G-InfoOut.doc (1998-01-20 10:09)

Reproduction ist oder die Verwendung dieser Vorlesung ist oder in jeder Form an der Universität Erlangen-Nürnberg ist oder der Zustimmung des Autors.

G.36

## G.8 Andere Geräte (2)

### ■ CD-ROM

- ◆ wird wie Platte behandelt (eigener Treiber)
- ◆ nicht beschreibbar
- ◆ spezielle Treiber für Audio-Tracks

### ■ Maus und Tastatur

- ◆ meist über serielle Schnittstellen und bestimmtes Protokoll implementiert

### ■ Floppy-Disk

- ◆ wird im Prinzip wie Platte behandelt (eigener Treiber)
- ◆ spezielle Dateisysteme zur Realisierung von FAT Dateisystemen unter UNIX

### SP I

#### Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1998

G-InfoOut.doc (1998-01-20 10:09)

Reproduction ist oder die Verwendung dieser Vorlesung ist oder in jeder Form an der Universität Erlangen-Nürnberg ist oder der Zustimmung des Autors.

G.37