Script **generated by TTT**

Title: Grundlagen_Betriebssysteme (23.01.2013)

Date: Wed Jan 23 13:14:44 CET 2013

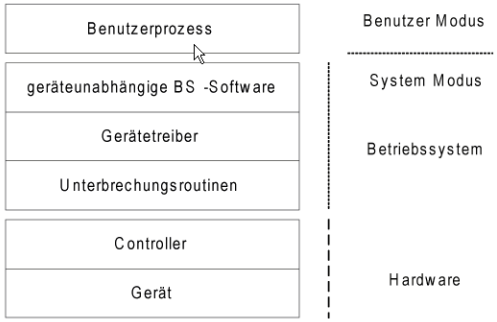
Duration: 44:08 min

Pages: 13

Ein-/Ausgabe

Hauptaufgabe eines BS: Steuerung und Überwachung aller E/A-Geräte.

Ein E/A-System eines BS ist typischerweise in mehrere Schichten unterteilt:



Unterbrechungsroutinen: Handhabung der Rückmeldungen vom Geräte-Controller, z.B. nach Beendigung eines Druckauftrags.

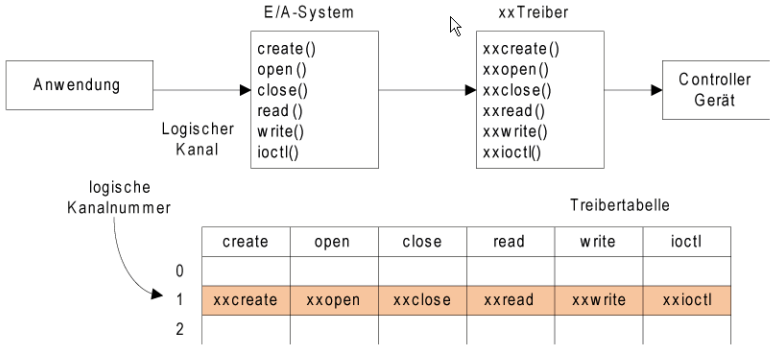
Gerätetreiber: Ausführung der geräteabhängigen Steuersoftware; zuständig für alle Geräte eines Gerätetyps.

Geräteunabhängige Software: belegen eines Geräts, puffern von Information.

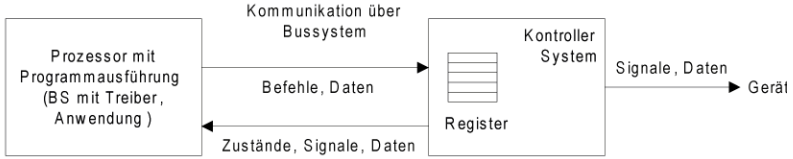
Geräteverwaltung

Eine Hauptaufgabe des BS ist die ***einheitliche Darstellung*** der unterschiedlichen E/A-Geräte und Treiber.

Zuordnung von logischen Kanälen zu physischen Geräten.



Treiber sind gerätetyp-spezifisch und sie schaffen die Verbindung zwischen Anwenderprozessen und den Geräten bzw. deren Controller.



[Aufgaben eines Treibers]

Generated by Targeteam

Treiber bedienen die Hardware zur Gerätesteuerung (Controller), um

- Gerätezustände abzufragen
- Befehle an das Gerät zu übermitteln,
- Daten von/zum Gerät zu übermitteln.

Aufgaben eines Treibers sind

- definiert das Gerät gegenüber dem BS.
- definiert die gerätespezifische Datenbasis.
- initialisiert den Controller und das Gerät beim Systemstart.
- wandelt allgemeine E/A-Anforderungen in gerätespezifische Befehle um.
- aktiviert das Gerät.
- antwortet auf Hardwaresignale des Geräts bzw. des Controllers.
- meldet Geräte- und Controllerfehler.
- empfängt/sendet Daten und Zustandsinformation vom/zum Gerät.
- verarbeitet mehrere E/A-Anforderungen gleichzeitig oder überlappt (Multithreading).
- puffert Daten bei Ein- und Ausgabe.

Geräteunabhängige E/A

Die Realisierung abstrakter Geräte und die Definition einer generischen Gerätearchitektur ist charakteristisch für viele Betriebssysteme. Ein Aspekt ist die ***Bereitstellung einer einheitlichen Schnittstelle*** zwischen Gerätetreiber und dem Rest des BS.

⇒ vereinfacht die Programmierung und Einbindung neuer Treiber-Software.

Namensgebung von E/A-Geräten

Nutzung von symbolischen Namen für Geräte.

geräteunabhängiger Teil des BS bildet symbolische Namen auf die entsprechenden Treiber ab.

[Einbettung der E/A in Unix]

[Standardisierte E/A-Funktionen]

[Pufferung]

[Spooling]

Generated by Targeteam

in Unix erfolgt der Zugriff auf praktisch jedes E/A-Gerät über Funktionen des Dateisystems.

in vielen Unix Systemen werden alle Geräte unter dem Teilbaum `/dev` verwaltet.

der Dateiname charakterisiert den jeweiligen Typ des E/A-Geräts.

- `/dev/ttya` : physische serielle Schnittstelle
- `/dev/pty01` . . . : abstrakte serielle Schnittstellen
- `/dev/fd0` . . . : Diskettenstationen
- `/dev/sd0` . . . : Festplatten
- `/dev/le0` . . . : Netzkarten

`/dev/sd0` spezifiziert genau einen I-Node einer Spezialdatei. I-Node enthält:

Hauptgerätenummer : Festlegung des Gerätetreibers.

Nebengerätenummer : Parameter an Treiber zur Festlegung des konkreten Geräts, von dem gelesen, bzw. auf das geschrieben wird.

Generated by Targeteam

Die E/A-Programmierschnittstelle wird meist als Teil einer Systembibliothek (z.B. in C/C++) bereitgestellt. Typische generische E/A-Funktionen sind:

open(): eröffnet einen logischen Kanal zu einem Gerät und liefert einen Identifikationswert (Deskriptor, Handle) für die anschließende Nutzung.

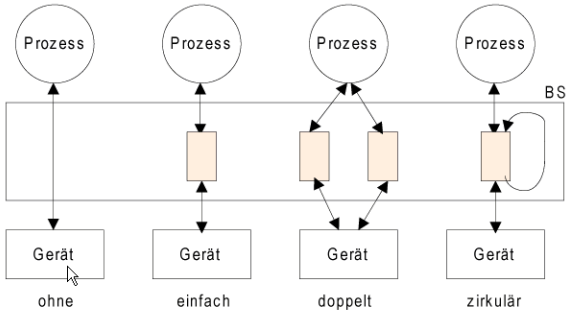
close(): ein vorher geöffneter Kanal wird geschlossen.

read(): liest eine Anzahl von Byte (als Bytestrom) vom Gerät ein.

write(): gibt eine Anzahl von Byte an das Gerät aus.

ioctl(): dient dazu, die Betriebsart des Geräts zu ändern.

Direkte Durchreichung der E/A-Daten zwischen Benutzerprozess und Gerät möglich, meist jedoch Zwischenpufferung im BS.



einfach : auf Puffer wird wechselseitig über BS von Benutzerprozess und Gerät zugegriffen.

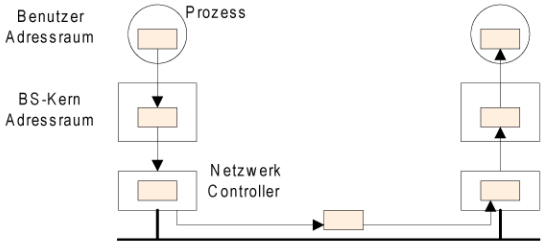
doppelt : Benutzerprozess und Gerät können parallel arbeiten.

zirkulär : Koordination der Zugriffe von Benutzerprozess und Gerät, um gegenseitiges Überschreiben zu vermeiden.

Wiederholte Pufferung

Wiederholte Pufferung

Pufferung ist ein weitverbreitete Technik; jedoch bei häufiger Pufferung ist mehrfaches Kopieren der Daten notwendig, darunter leidet die Geschwindigkeit des Gesamtsystems.



Geräteunabhängige E/A

Die Realisierung abstrakter Geräte und die Definition einer generischen Gerätearchitektur ist charakteristisch für viele Betriebssysteme. Ein Aspekt ist die ***Bereitstellung einer einheitlichen Schnittstelle*** zwischen Gerätetreiber und dem Rest des BS.

⇒ vereinfacht die Programmierung und Einbindung neuer Treiber-Software.

Namensgebung von E/A-Geräten

Nutzung von symbolischen Namen für Geräte.

geräteunabhängiger Teil des BS bildet symbolische Namen auf die entsprechenden Treiber ab.

Einbettung der E/A in Unix

Standardisierte E/A-Funktionen

Pufferung

Spooling

Ein-/Ausgabe

Hauptaufgabe eines BS: Steuerung und Überwachung aller E/A-Geräte.

RAID

Erhöhung der Plattenkapazität durch Zusammenschluss von mehreren kleinen Festplatten zu einer großen virtuellen Platte.

⇒ RAID = redundant array of inexpensive disks.

Für das BS sieht RAID wie eine einzelne Festplatte aus.

keine Änderung an Anwendungssoftware notwendig, um ein RAID-System nutzen zu können.

üblicher Plattencontroller wird durch einen RAID Controller ersetzt.

Daten werden über die Platten verteilt, was eine parallele Verarbeitung ermöglicht.

höhere Zuverlässigkeit durch Redundanz.

Unterscheidung zwischen unterschiedlichen Varianten

RAID Level 0 bis RAID Level 6.

Die restlichen Varianten RAID Level 3 - 6 können im Tanenbaum nachgelesen werden. Kombinationen von RAID Level sind möglich, z.B. RAID 0+1.