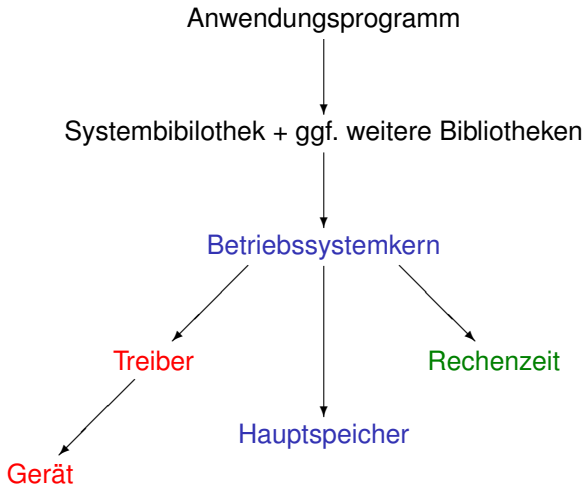


# Treiberentwicklung, Echtzeit- und Betriebssysteme

Prof. Dr. Peter Gerwinski

19. Mai 2014

# Treiberentwicklung, Echtzeit- und Betriebssysteme



# 4 Grafik

Beispiel für User-Space-Treiber

## 4.1 Hardwarenahe Aspekte

- Text- und Grafikmodi
- Interpretation der Bits, Bytes und Bytefolgen
- Interleaving

Umschalten zwischen verschiedenen Modi

- Output-Ports
- Programme im ROM, Software-Interrupts

# 4 Speicherverwaltung

## 4.1 Bank Switching

Apple II (1977),  
Commodore 64 (1982), ...

- Physikalisch adressierbar: 64 kB
- Bedarf nach mehr Speicher
- Lösung: Über Output-Port Teile des Speichers umschalten
- Das Programm muß sich derweil vollständig in einem anderen Teil des Speichers befinden.

→ Anwendung für Grafik: *mehrere oder alternierende Grafikseiten*



# 4 Grafik

## 4.2 Low Level vs. High Level

### Low Level

- direkter Speicherzugriff
- Hardware-Beschleunigung

### High Level

- Portabilität
- Netzwerkfähigkeit
- Trennung verschiedener Programme  
→ Fenster

Kompromiß: kontrollierter Low-Level-Zugriff

Integration in das Betriebssystem

- Flexibilität vs. Einheitlichkeit

# 4 Speicherverwaltung

## 4.2 Speichersegmentierung

IBM PC (1981),  
IBM PC/AT (1984), ...

- Physikalisch adressierbar:  
1 MB bis 16 MB
- Problem: 16-Bit-Register  
können nur 64 kB adressieren
- Lösung: Segment- und Offset-Adressen
- Zusätzlich: Länge der Segmente einstellbar (Deskriptortabelle)  
→ Speicherschutz
- Konzept übernommen für 32-Bit-Prozessoren  
Physikalisch adressierbar: 4 GB  
→ Segmente dienen nur noch dem Speicherschutz

→ Anwendung für Grafik: *Grafik-Speichersegment(e)*



# 4 Speicherverwaltung

## 4.3 Virtuelle Speicherverwaltung

- zusätzlich zur Segmentierung (nachgeschaltet)
- Aufteilung des Speichers in gleich große Kacheln – *Pages*
- Speicher muß nicht wirklich vorhanden sein

—→ bei Zugriff: *Page Fault* („Interrupt“)

—→ Betriebssystem kann Speicher auf Platte auslagern

—→ weitere Anwendung: Dateizugriff über „Speicherbereich“

—→ Anwendung für Grafik: *Frame Buffer*