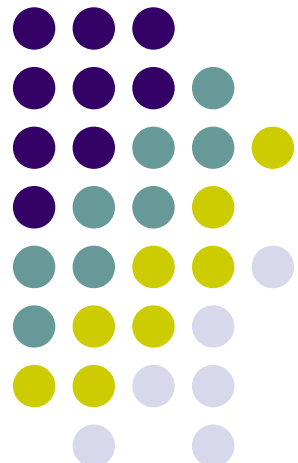
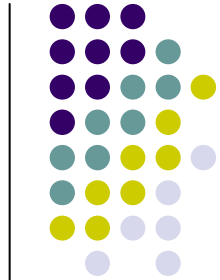
Device Driver

Einführung in Device Driver unter Linux / Unix, Schwerpunkt Netzwerktreiber

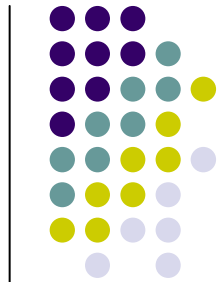


Geräte Treiber unter Unix



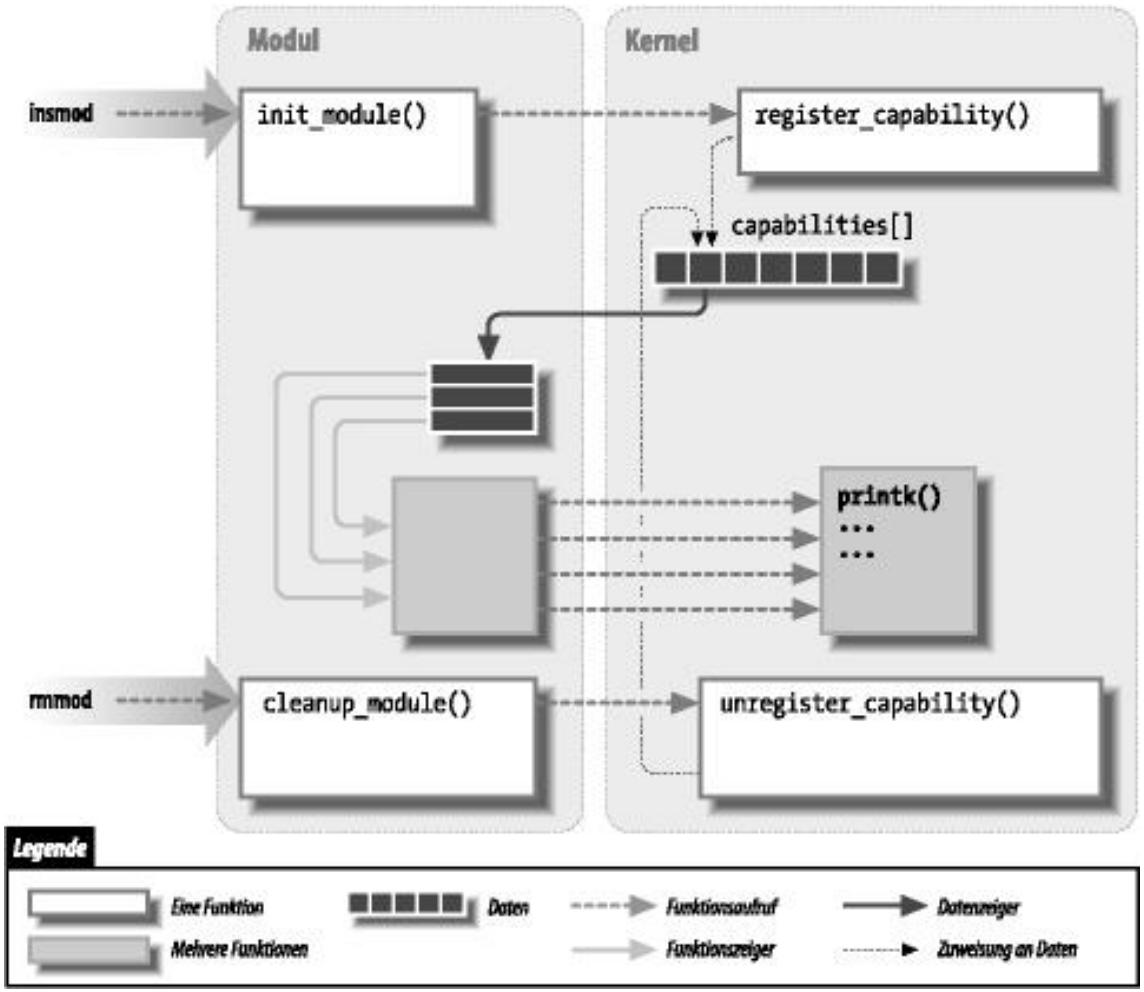
- Was sind Treiber
- Arten von Treibern
- Wie funktioniert ein Netzwerktreiber

Treiber als Schnittstelle



- Viele Geräte werden als Datei repräsentiert
- Kernel gibt Schnittstellen Funktionen vor
- Treiber implementiert diese Funktionen
- Schnittstellen Funktionen sind für alle Geräte gleich

Treiber als Module

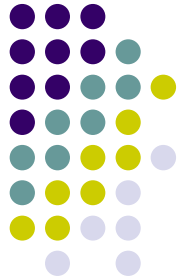


- Module können zum Kernel hinzugelinkt werden
- Module bedienen Anfragen
- Module müssen initialisiert werden

Ressourcen

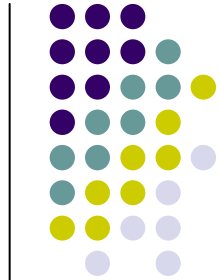
- Module können I/O Speicher, Speicher und Interrupt Leitungen anfordern.
- Ressourcen sollten explizit angefordert werden, um Eindeutigkeit sicherzustellen.
- Interrupts müssen angefordert werden um benutzt zu werden.

Interrupts



- Verhindert warten auf langsame Geräte
- Treiber muss Interrupt registrieren (IRQ)
- Eine Interrupt Leitung kann von mehreren Geräten benutzt werden

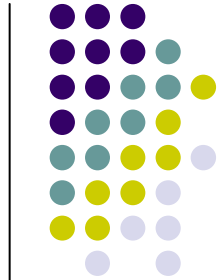
Major und Minor Nummern



- Major Nummern ordnen Treiber einem Gerät zu

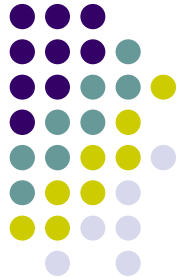
- Treiber unterscheidet über Minor Nummer zwischen den ihm zugeordneten Geräten

Registrieren beim Kernel



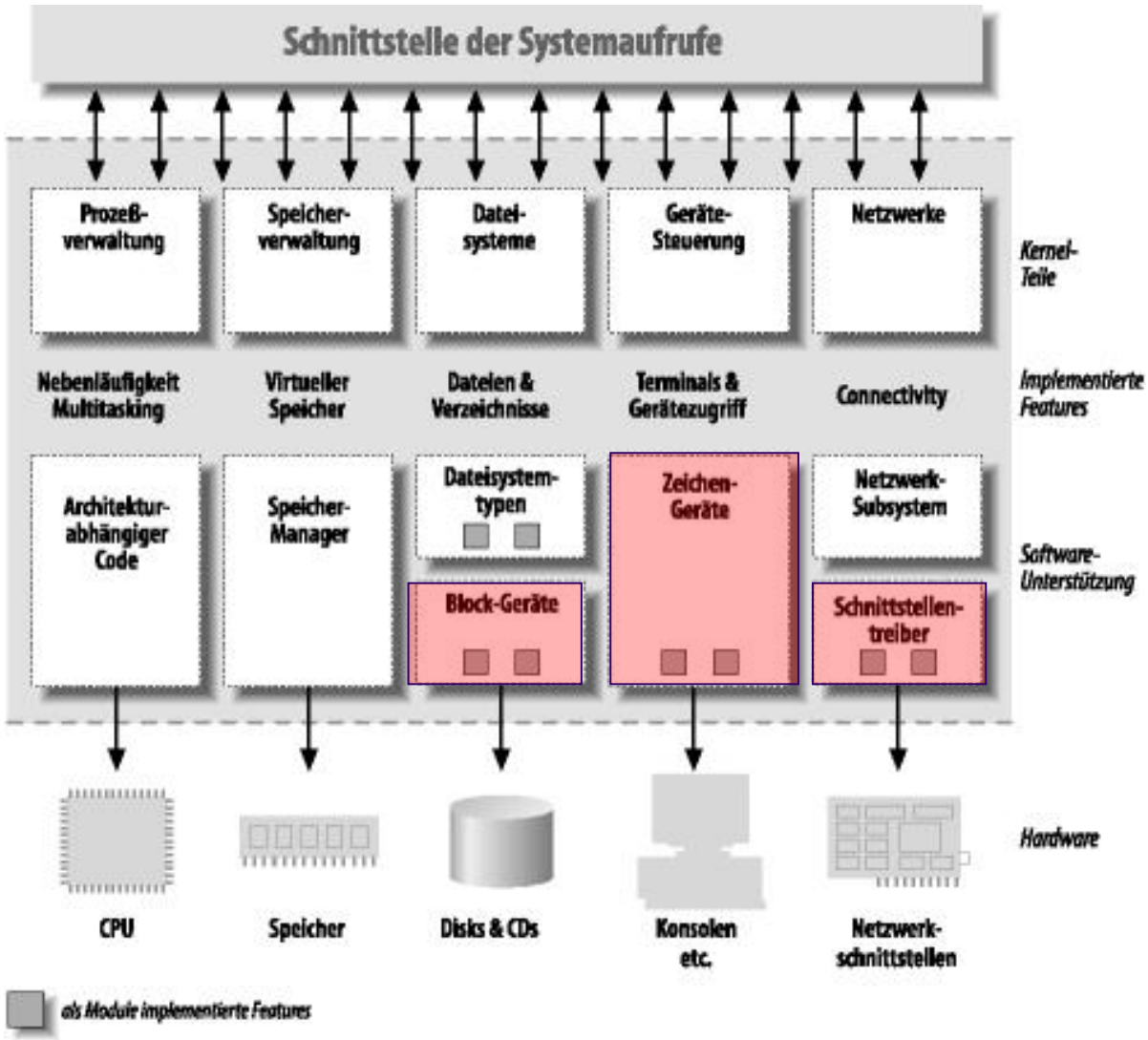
- Registrieren der Treiber Funktionen
- Ermitteln der physikalischen Adresse
- Anfordern von Ressourcen erst beim ersten Aufruf

Arten von Geräte Treibern



- Zeichen Geräte Treiber
- Block Geräte Treiber
- Netzwerk Treiber
- SCSI-, USB-Treiber, FireWire-Treiber

Kernel Übersicht



Zeichen Geräte Treiber

- Auf Daten wird Zeichen - weise zugegriffen
- Zugriff meist sequenzielle
- Zugriff über Dateisystem

Beispiele: Drucker, Geräte an seriellen Ports

Zeichen Geräte Treiber

```
int register_chrdev(unsigned int major, const char *name,
struct file_operations *fops);
```

- In *fops sind Zeiger auf alle Funktionen des Treibers

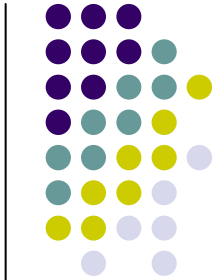
- stuct file_operations wird in <linux/fs.h> definiert

Block Geräte Treiber

- Auf Daten wird Blockweise zugegriffen
- Block Geräte können ein Dateisystem aufnehmen, bzw. gemountet werden
- Random access Zugriff

Beispiele: Festplatten, Disketten usw.

Block Geräte Treiber



*int register_blkdev(unsigned int major, const char *name, struct block_device_operations *bdops);*

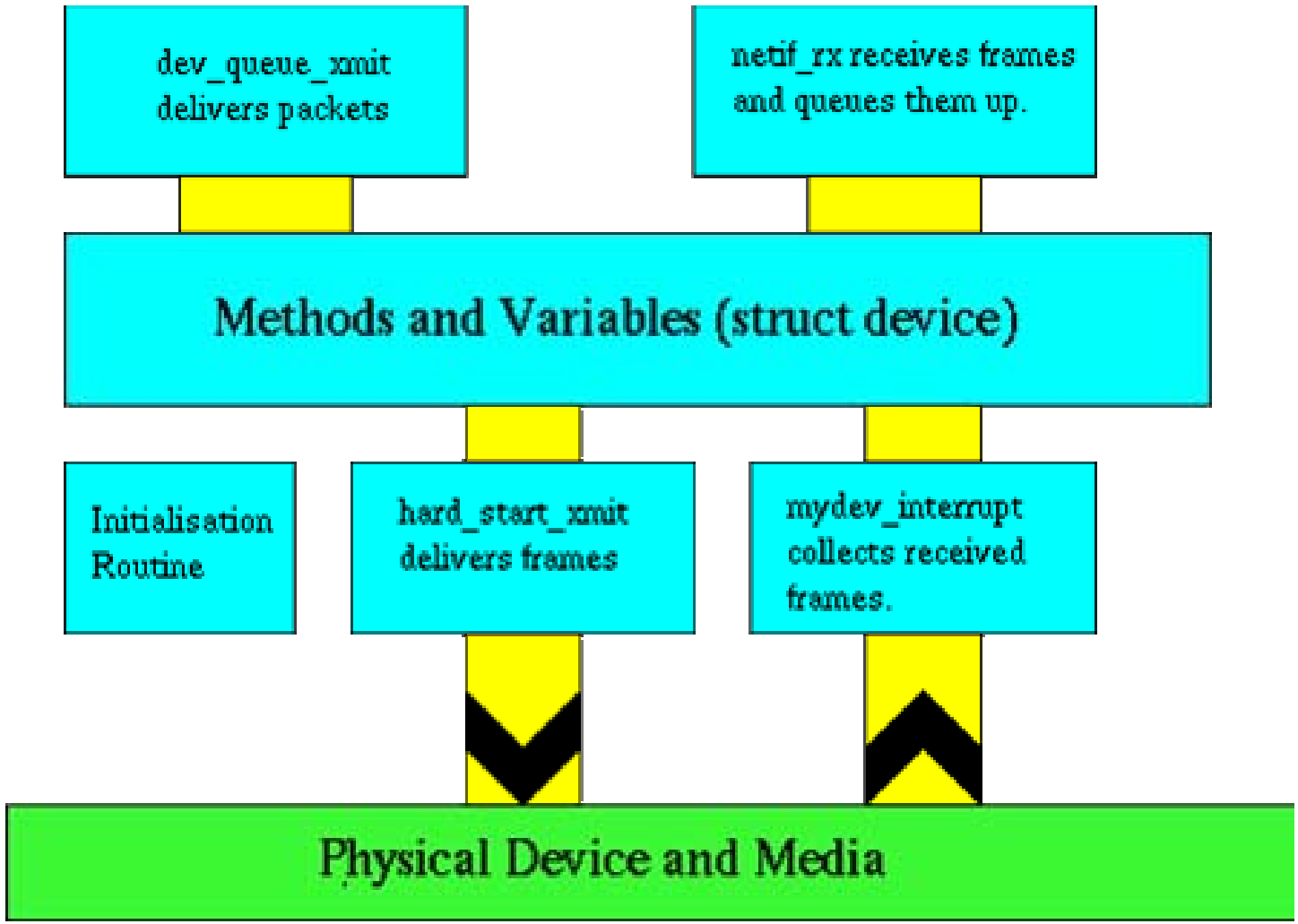
- In *bdops sind Zeiger auf alle Funktionen des Treibers
- stuct file_operations wird in <linux/blkdev.h> definiert

Netzwerk Treiber

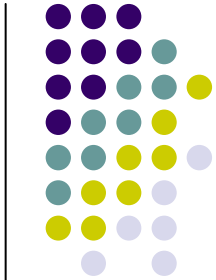
- Netzwerk Geräte werden nicht als Datei repräsentiert

- Netzwerk Geräte werden in einer globalen Liste gespeichert

Netzwerk Treiber



Anmelden von Netwerk Treibern



- Schnittstellen werden als *struct net_device* dargestellt
- Mit register_netdev wird der Treiber registriert
- Initialisieren der Warteschlange ()

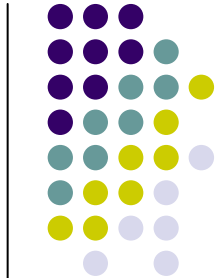
```
void netif_start_queue(struct net_device *dev);
```

Pakete senden

```
int (*hard_start_xmit) (struct sk_buff *skb, struct net_device *dev);
```

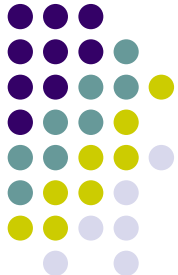
- Pakete sind in Socket-Buffer-Struktur.
  <linux/skbuff.h >
- Pakete müssen nur auf Konsistenz überprüft werden.
- Verwaltung der Warteschlange.

Pakete empfangen



- Empfangen mittels Interrupt
- Allozieren von Speicher für das eingehende Paket
- Weiterreichen der Daten an die Oberen Schichten des Systems

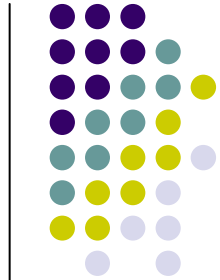
Ein Interrupt registrieren



```
int request_irq(unsigned int irq, void (*handler)(int, void *, struct pt_regs *),
unsigned long flags, const char *dev_name, void *dev_id);
```

unsigned int irq	gewünschte Interrupt Nummer
void (*handler)(int, void *, struct pt_regs *)	Die Handler Funktion
unsigned long flags	Flags zur Verwaltung
const char *dev_name	Zuordung zu Gerät
void *dev_id	Zuordnung bei gemeinsamer Interrupt Benutzung

Interrupt Handler



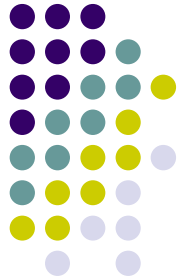
- Keine Kommunikation mit User Level
- Muss atomar sein
- Weckt Prozesse auf

Interrupt Handler bei Netwerk Treibern

- Überprüft, ob gesendet oder empfangen wurde

- Gibt bei Empfang die Warteschlange frei

Timeouts



- Treiber legt Übertragungszeitraum fest
- Treiber hat Timeout Routine (siehe *struct net_device*)
- Treiber behebt Fehler und ergänzt ggf verlorene Interrupts

Verbindungen

- Carrier Signal bei aktiver Verbindung
- Signal kann manuell an- und abgeschaltet werden
- Ist kein Carrier Signal auf der Leitung gibt es kein Netzwerk

Fragen ?