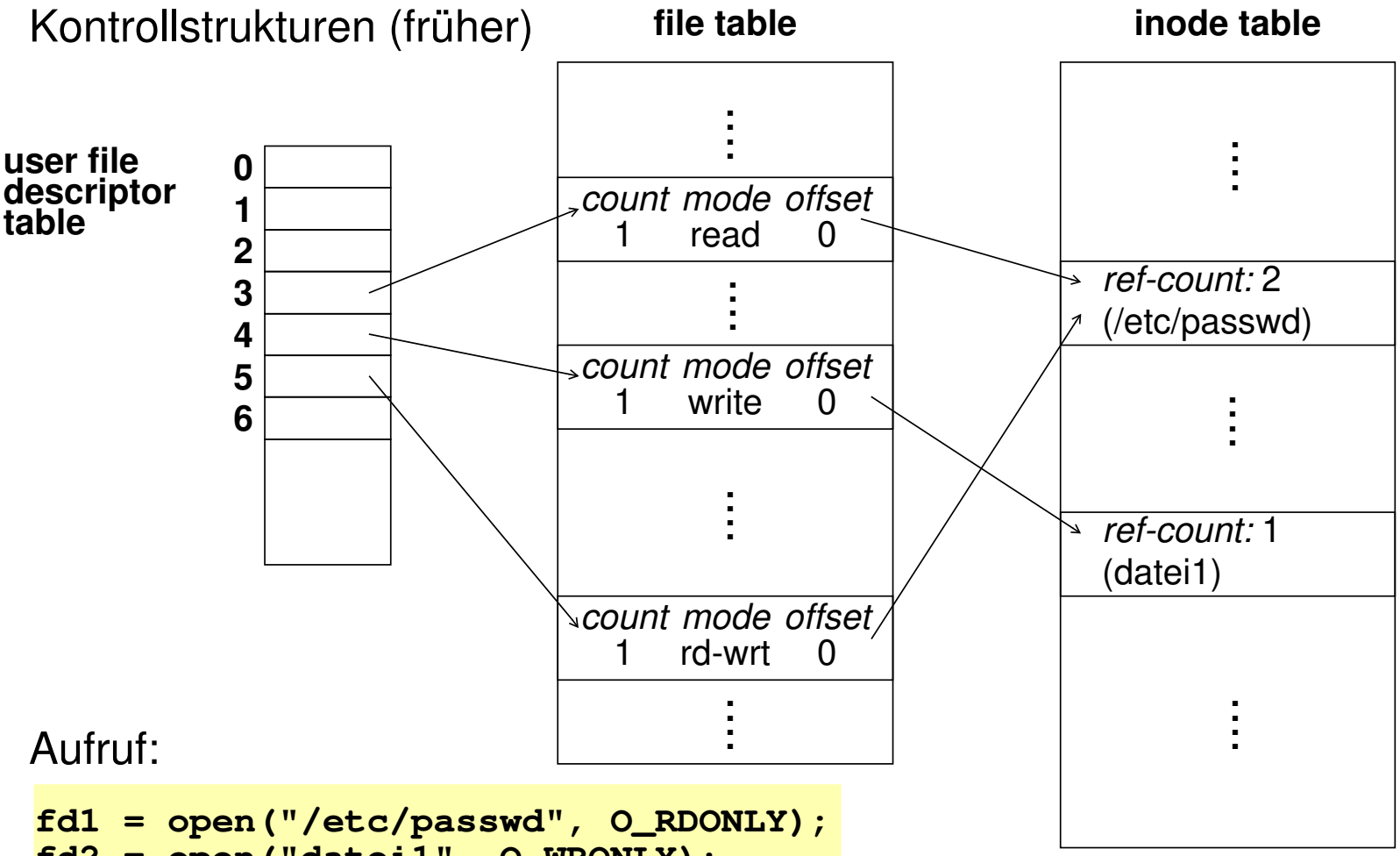
1 Umwandlung: Pfad : Inode (2)

- Berücksichtigt werden hierbei:
  - Symbolic Links
    - werden beim Absteigen im Baum verfolgt
  - Mount-Punkte
    - beim Absteigen im Baum am Mount-Punkt Wechsel zur *root* des eingehängten Dateisystems
    - beim Aufsteigen im Baum (über "..") an der Root eines Dateisystems Wechsel auf den Mount-Punkt
  - Current Root
    - Überschreiten beim Aufsteigen im Baum (über "..") wird verhindert

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

2 Datei öffnen (3)

- Kontrollstrukturen (früher)

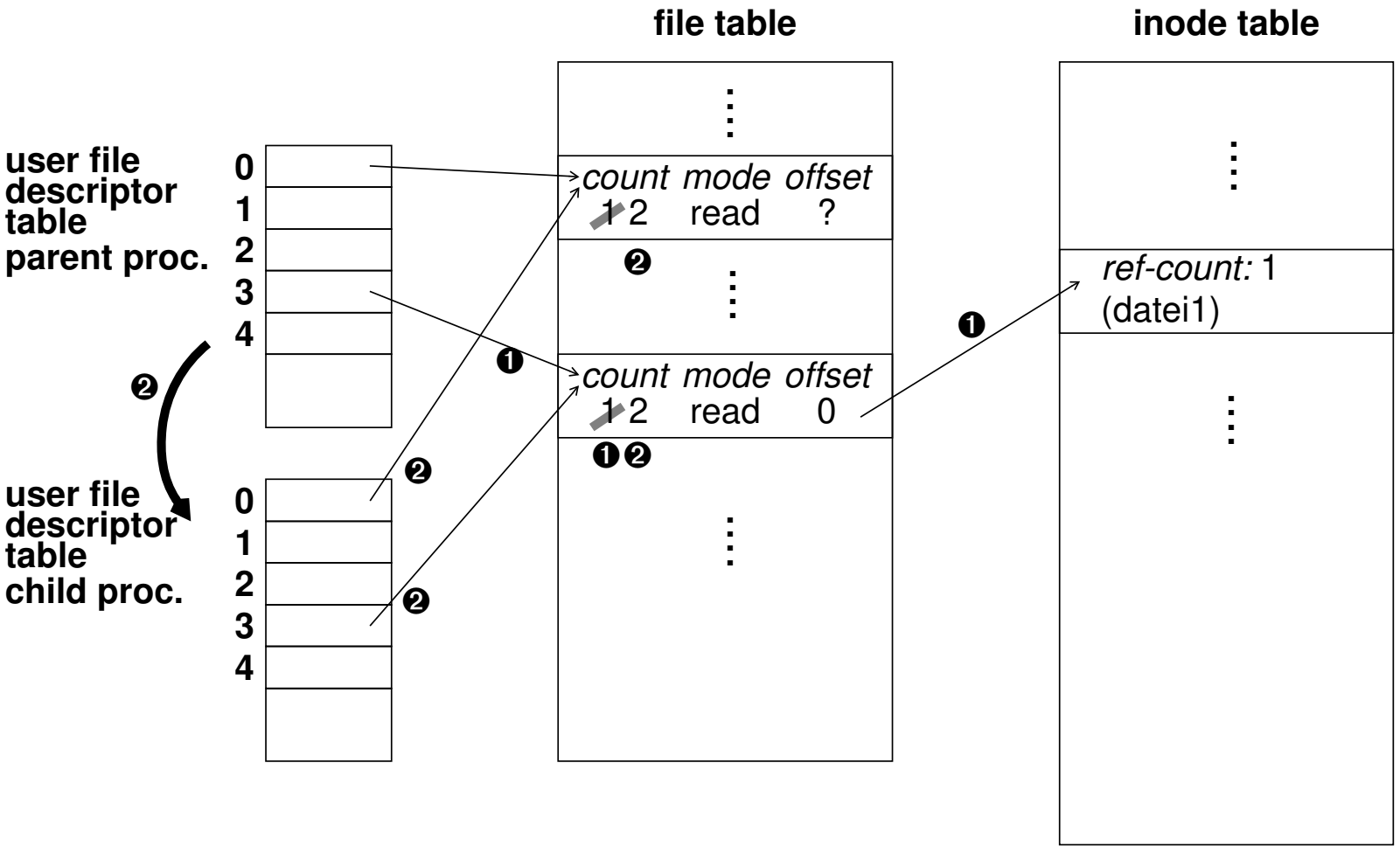


- Aufruf:

```
fd1 = open("/etc/passwd", O_RDONLY);
fd2 = open("datei1", O_WRONLY);
fd3 = open("/etc/passwd, O_RDWR);

-> fd1 == 3, fd2 == 4, fd3 == 5
```


Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.



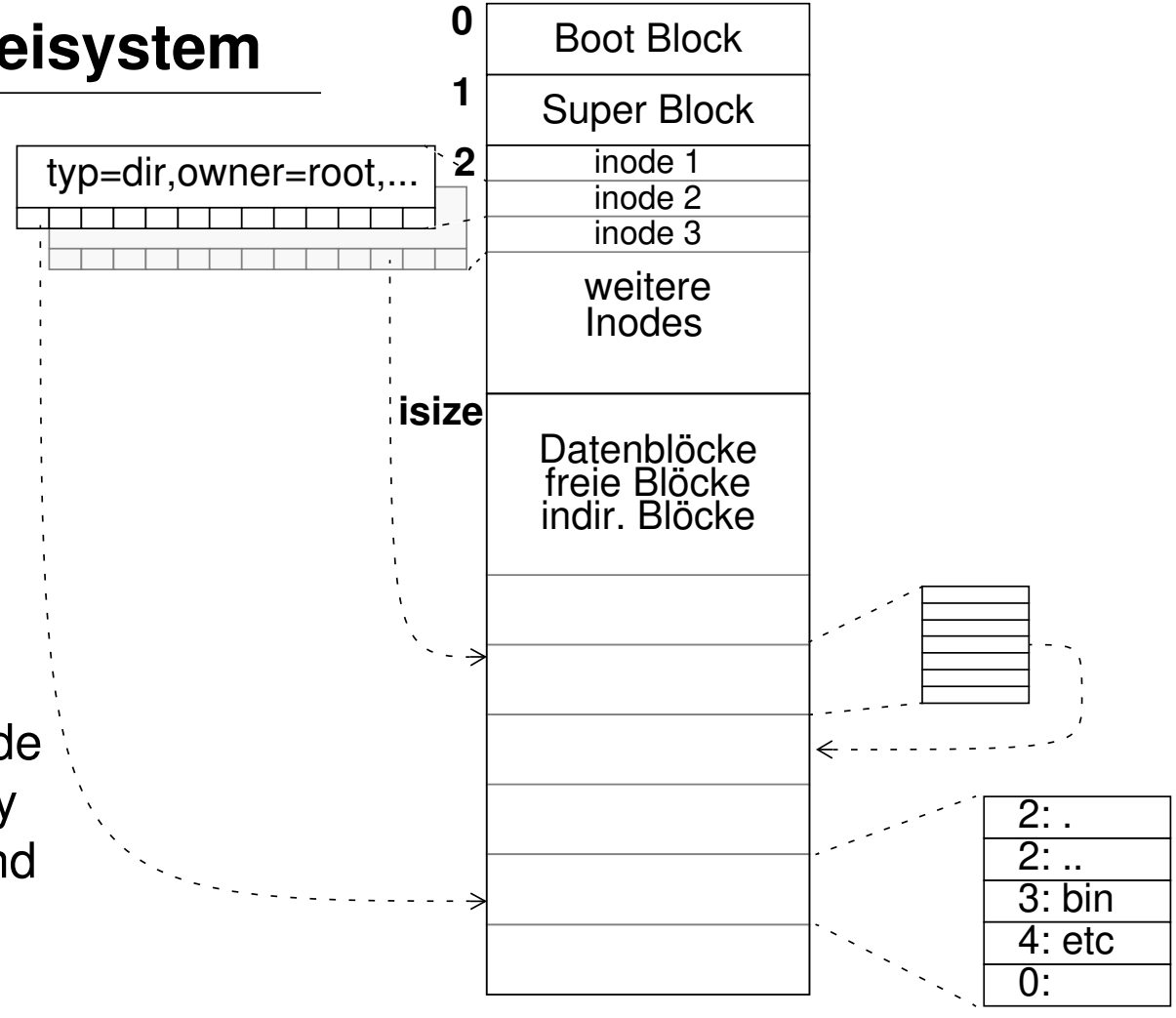

```
❶ fd1 = open("datei", O_RDONLY);
❷ fork( );
```


Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

I.5 Dateisysteme

Aufbau eines Dateisystems auf einer Disk

1 SystemV-Dateisystem



- Blockgröße 512 oder 1024 Byte

- große Unterschiede zwischen Berkeley fast-file-system und UNIX SystemV-Dateisystemen!

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

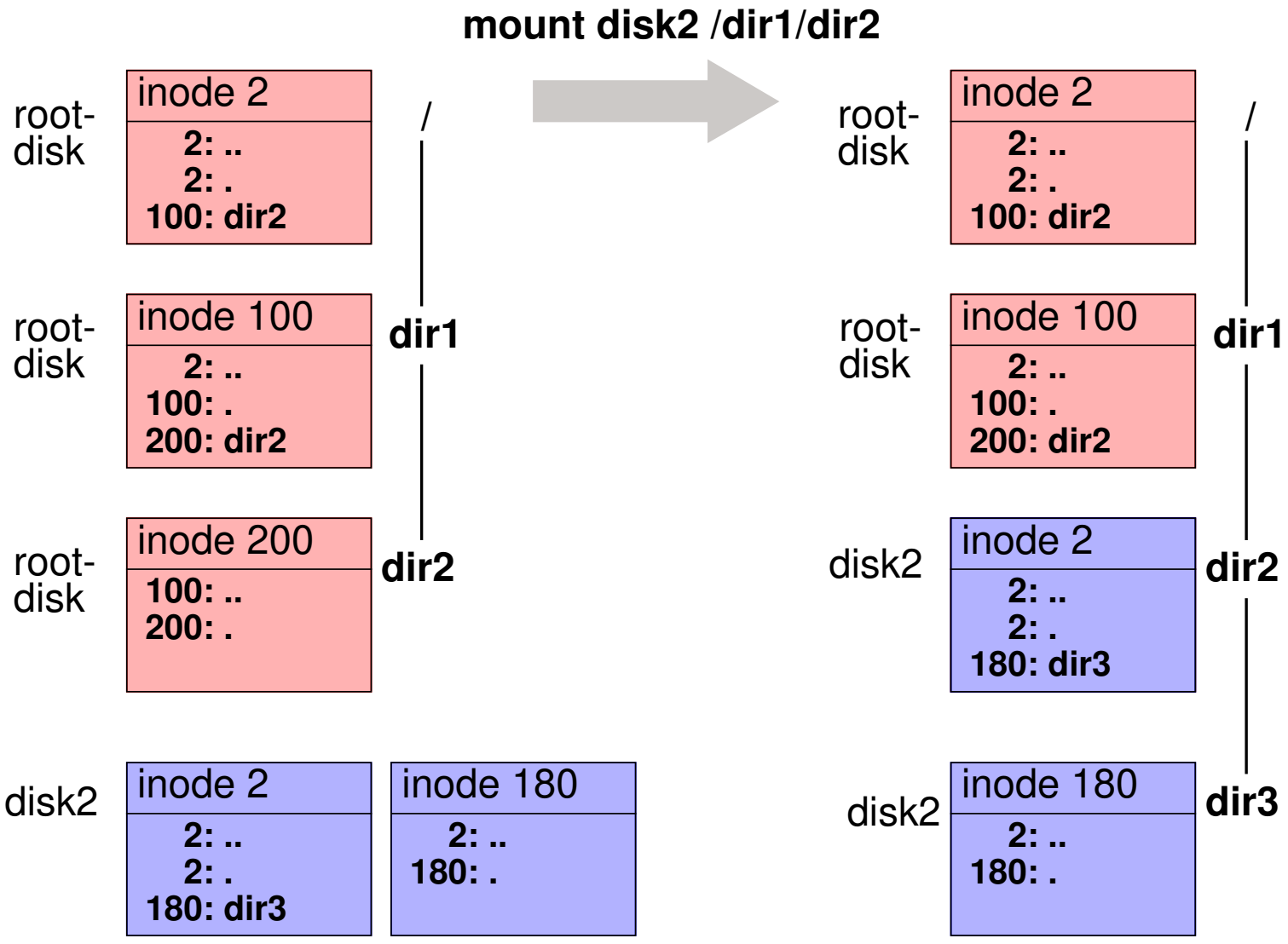
I.6 Virtuelles Dateisystem

1 Überblick

- Vnodes bilden Schnittstelle zu dateisystemunabhängigen Operationen auf Dateien
- das *Virtual File System* (**VFS**) stellt eine vergleichbare Schnittstelle für Operationen auf Dateisystemen zur Verfügung — z. B.
  - ***vfs_mount***, ***vfs_unmount***
  - ***vfs_statvfs*** Statistikdaten ermitteln
  - ***vfs_root*** root-vnode ermitteln
  - ***vfs_sync*** gepufferte Daten schreiben
- Implementierung durch VFS-switch-Tabelle
  - Dateisystemname, Initialisierungsfunktion, ***vfsops*** — Feld mit Zeigern auf die FS-Funktionen

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

2 Mount-Mechanismus (4)



Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

2 Einordnung in das ISO/OSI-Referenzmodell

7	Anwendungsschicht	NFS
6	Darstellungsschicht	XDR
5	Sitzungsschicht	RPC
4	Transportschicht	UDP / TCP
3	Netzwerkschicht	IP
2	Verbindungsschicht	Ethernet
1	physikalische Schicht	

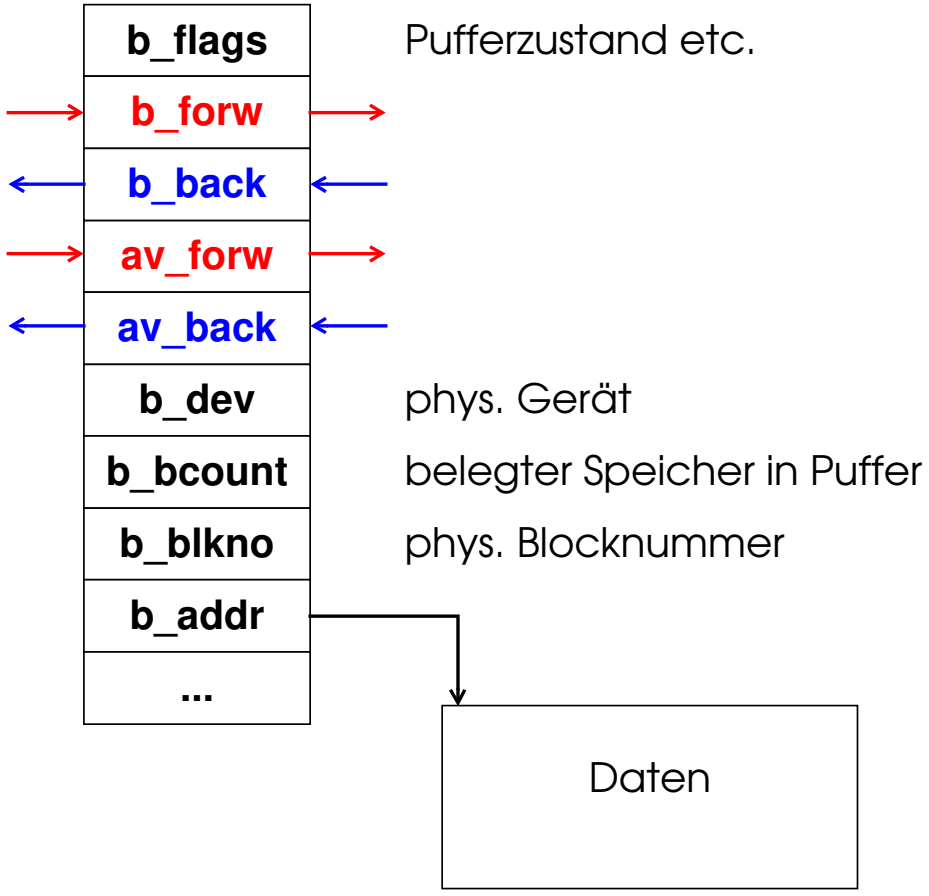
Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

5 Implementierung (2)

- Besonderheiten:
  - wird eine Datei auf Client-Seite gelöscht, obwohl noch Prozesse gültige Dateideskriptoren darauf halten:
    - ➤ wird sie auf der Serverseite nur umbenannt (nfs_...)
    - ➤ wird sie erst endgültig gelöscht, wenn die letzte Referenz auf den vnode auf der Client-Seite geschlossen wird

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

- in BSD4.4 sind die Puffer zusätzlich an den *vnodes* verankert
- Buffer-Header (struct buf, siehe *<sys/buf.h>)*



© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1998

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

2 Treiber-Funktionsvektoren

- *block-device-switch*:

```
struct bdevsw {
        int         (*d_open)();
        int         (*d_close)();
        int         (*d_strategy)();
        int         (*d_dump)();
        int         (*d_psize)();
        int         d_flags;
};
```

- *read*- und *write*-Aufrufe werden über den Block-Puffer abgewickelt
- die *strategy*-Funktion regelt die E/A-Operation zwischen Block-Puffer und Gerät
- *major*- und *minor number* werden im Inode gespeichert

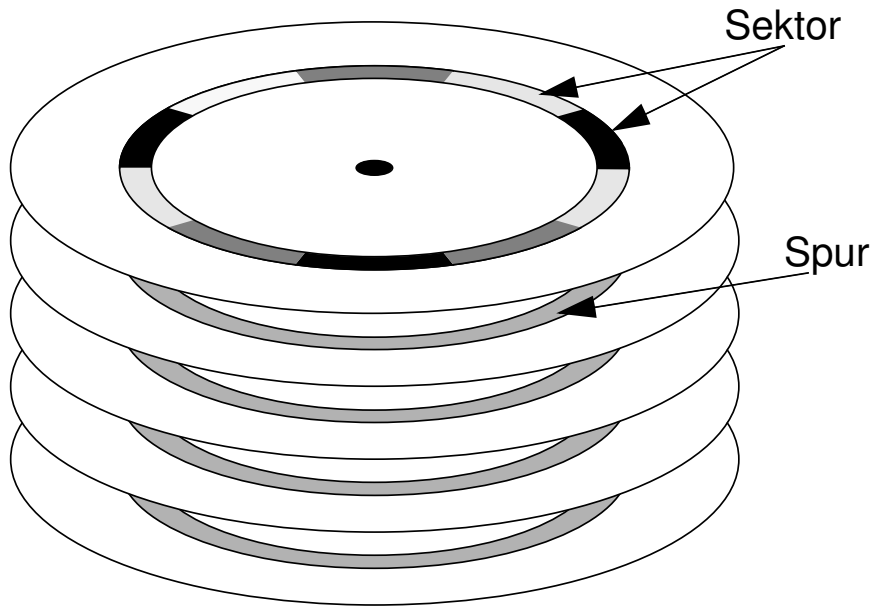
Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

1 Organisation einer Disk - Laufwerk

- typische Winchester-Festplatten bestehen aus mehreren, übereinander angeordneten Magnetplatten und einem Schreib-/Lese-Kopf pro Plattenoberfläche

- Begriffe:

Spur *(Track)*	Umlaufbahn auf einer Diskoberfläche
Sektor	Teil einer Spur
Zylinder	Menge der Spuren mit gleichem Radius über alle Oberflächen



Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

3 Organisation einer Disk — BS (2)

- die Position der einzelnen Partitionen wird festgelegt:
  - in älteren UNIX-Systemen statisch im Kern
  - in neueren UNIX-Systemen in Tabelle (*VTOC - Volume Table of Contents*) am Anfang der Platte (z. B. Zyl. 0, Spur 0, Sekt. 0)
- Partitionierung am Beispiel Solaris

```
Current partition table (original):
Total disk cylinders available: 2733 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders      Size            Blocks
 0        root    wm       0 -  101    75.70MB   (102/0/0)    155040
 1        swap    wu     102 -  446   256.05MB   (345/0/0)    524400
 2      backup    wm       0 - 2732     1.98GB   (2733/0/0) 4154160
 3  unassigned    wm       0              0       (0/0/0)          0
 4  unassigned    wm       0              0       (0/0/0)          0
 5  unassigned    wm       0              0       (0/0/0)          0
 6         usr    wm     447 - 2732     1.66GB   (2286/0/0) 3474720
 7  unassigned    wm       0              0       (0/0/0)          0
```


Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

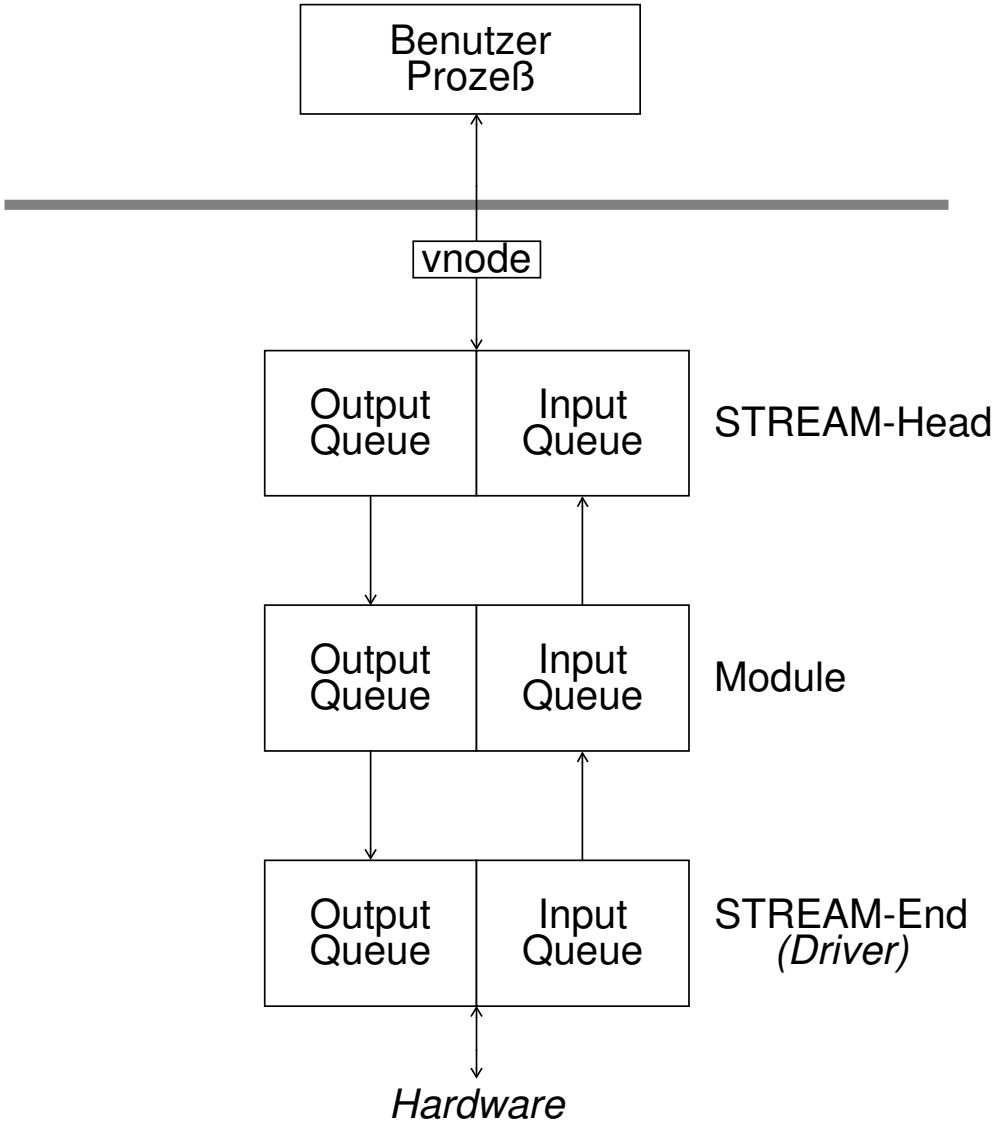
I.12 Disk-Treiber: Datenstrukturen und Kontrollfluß

1 Überblick

- Jedem Gerät (Platte) ist eine Warteschlange für Puffer des Buffer-Caches zugeordnet
- Puffer für Aufträge an den Disktreiber werden in die Warteschlange des zugehörigen Geräts einsortiert (*disksort( )*)
- Disktreiber entnimmt für freie Geräte Aufträge und startet den Controller
- bearbeitete Aufträge werden durch Interrupt gemeldet
  - ➥ Interrupt-Bearbeitung
    - – eigener Aktivitätsträger
    - – unabhängig von den beauftragenden Prozessen!
      - • weckt wartende Prozesse
      - • startet nächsten Auftrag aus Warteschlange

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

2 Funktionsweise (2)



Stream-Head:
Schnittstelle zum Benutzerprozeß, Zugang über *major/minor*-Nummer und *cdevsw*

Driver:
Gerätetreiber mit Hardware-Schnittstelle (Interrupt-Service-Routinen, etc.) oder *Pseudo-Device-Treiber*

Modul:
transformiert Daten auf dem Weg zwischen *Head* und *End* (Konvertierung, Pufferung, ...)

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

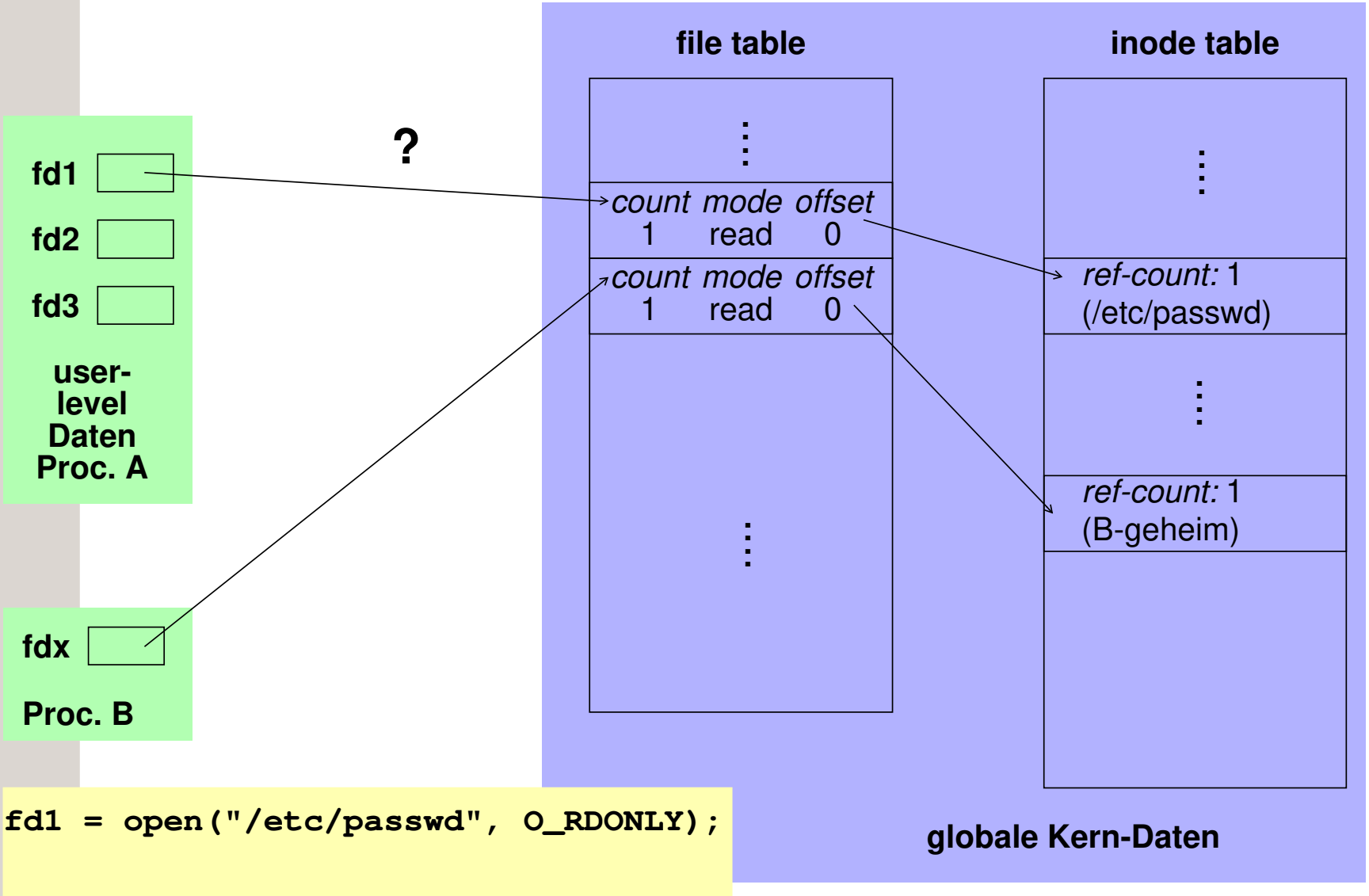
Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

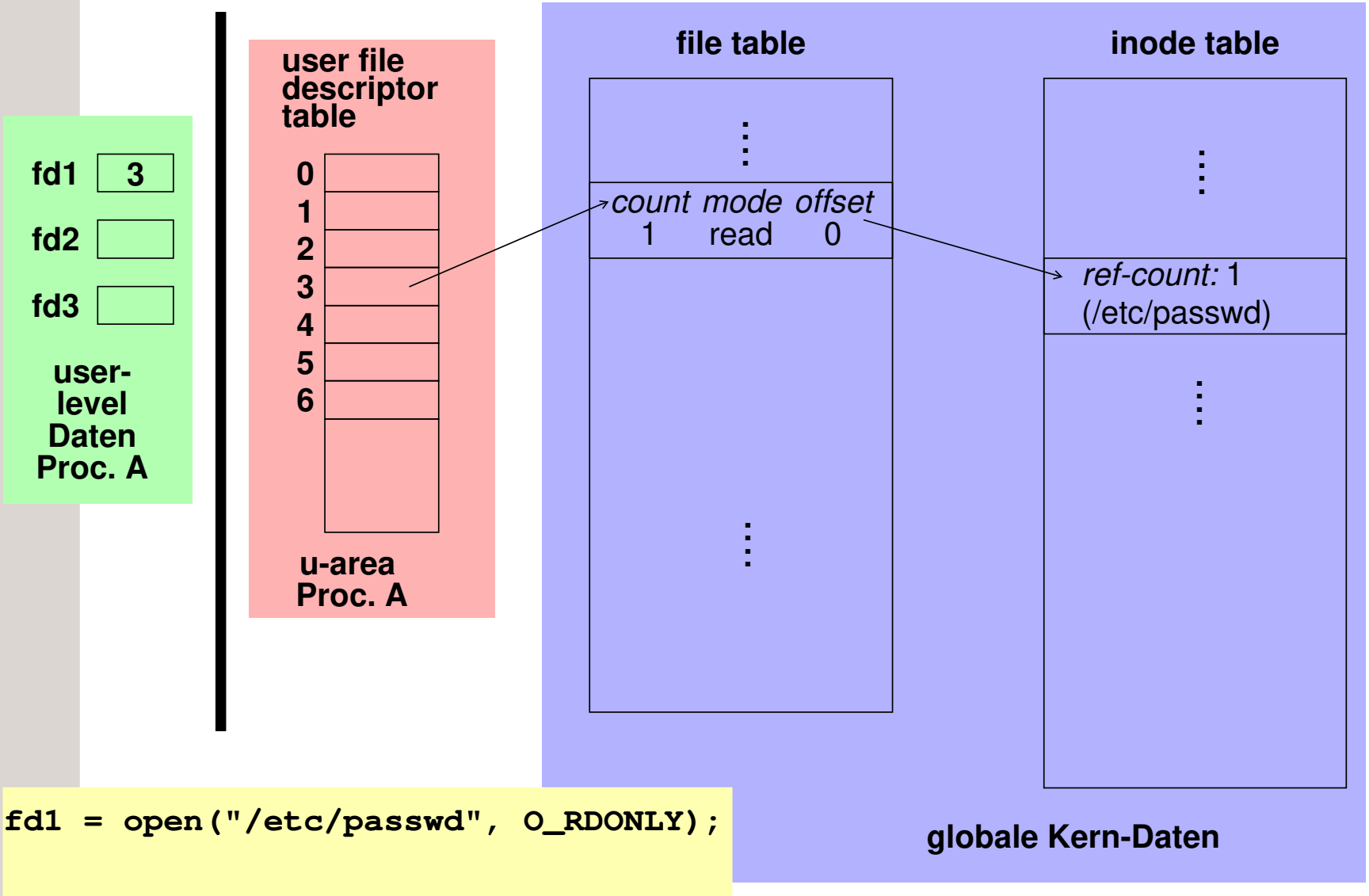
2 Datei öffnen (3)




```
fd1 = open("/etc/passwd", O_RDONLY);
```


Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

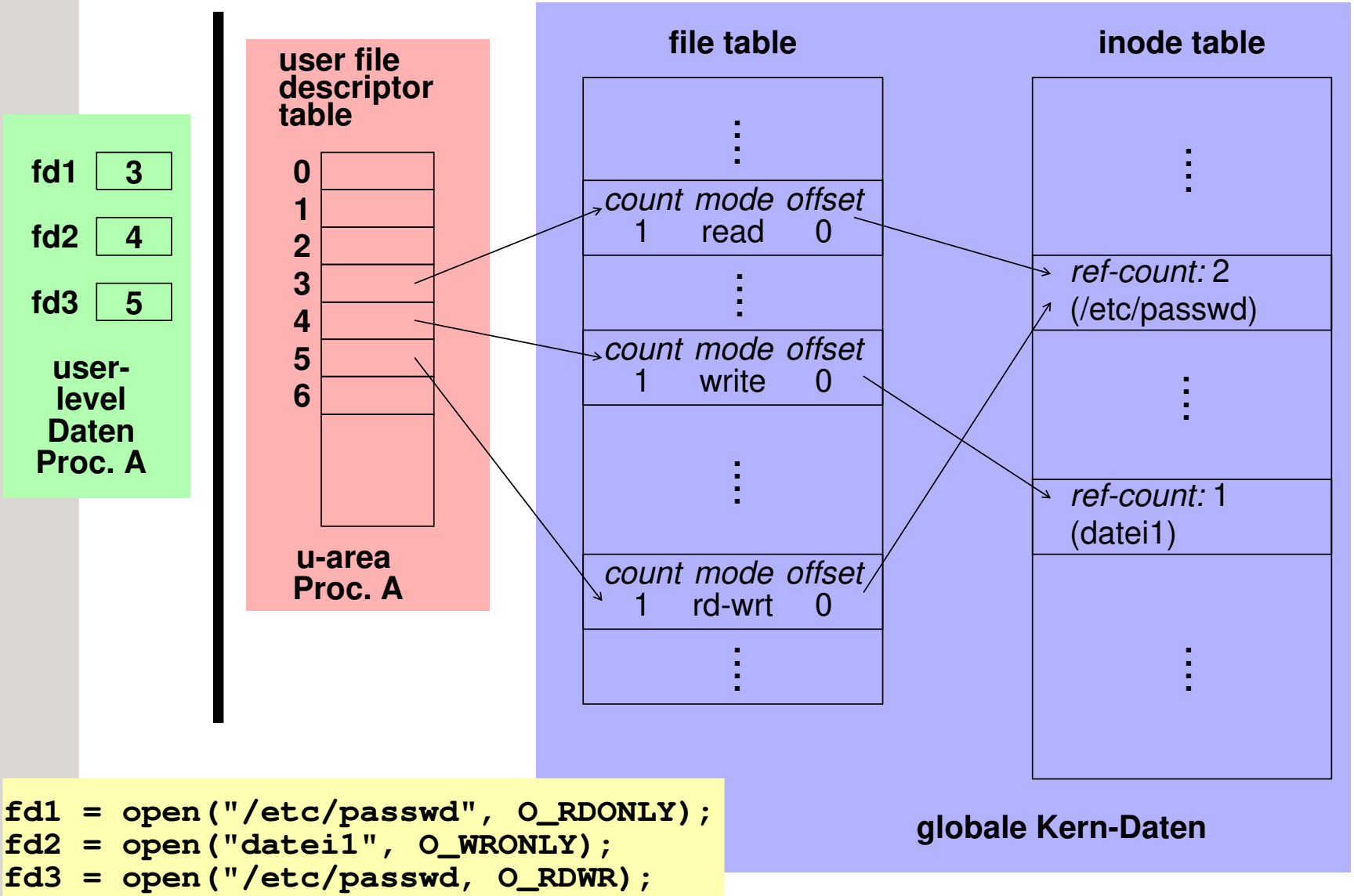
2 Datei öffnen (3)




```
fd1 = open("/etc/passwd", O_RDONLY);
```

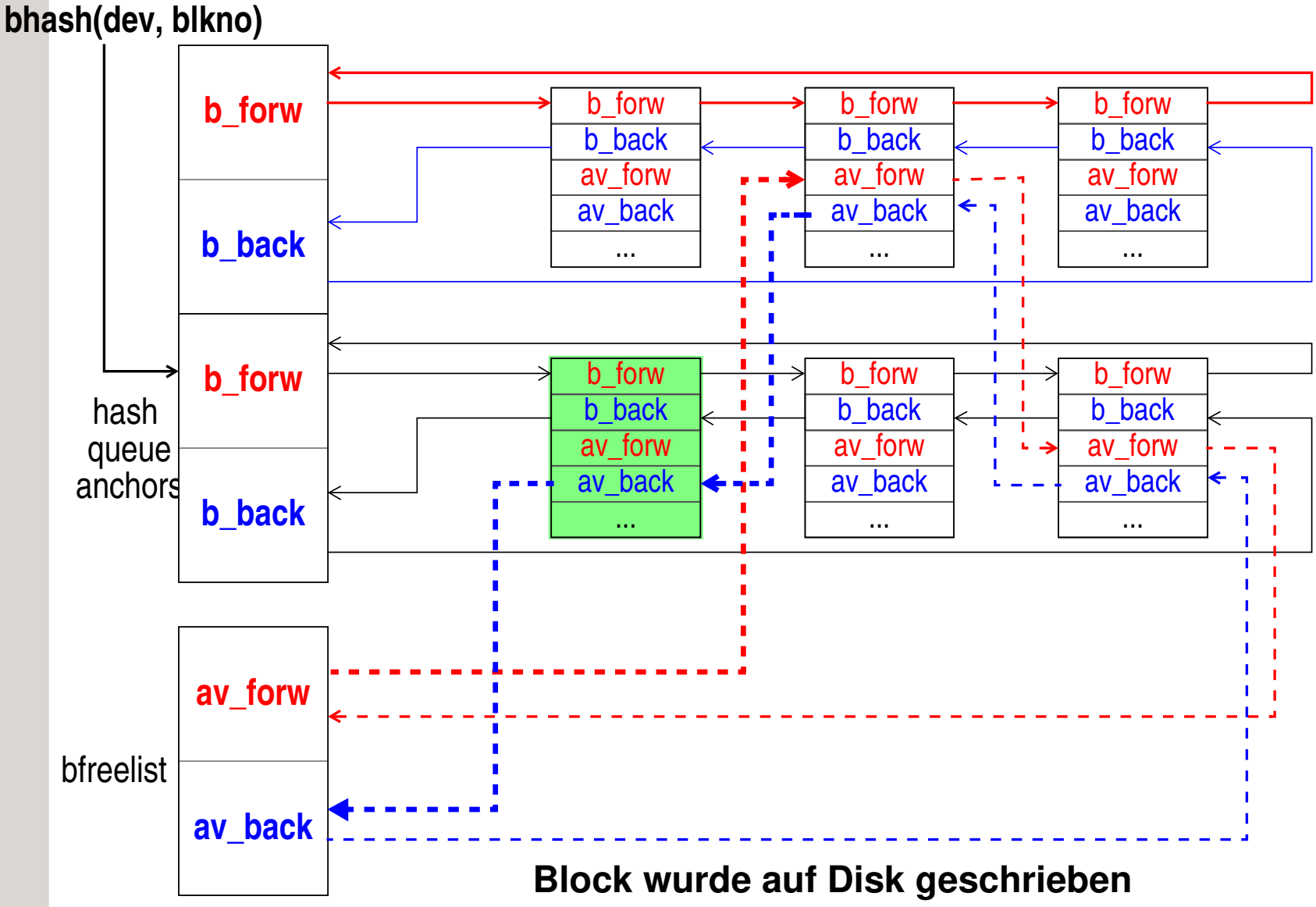

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

2 Datei öffnen (3)

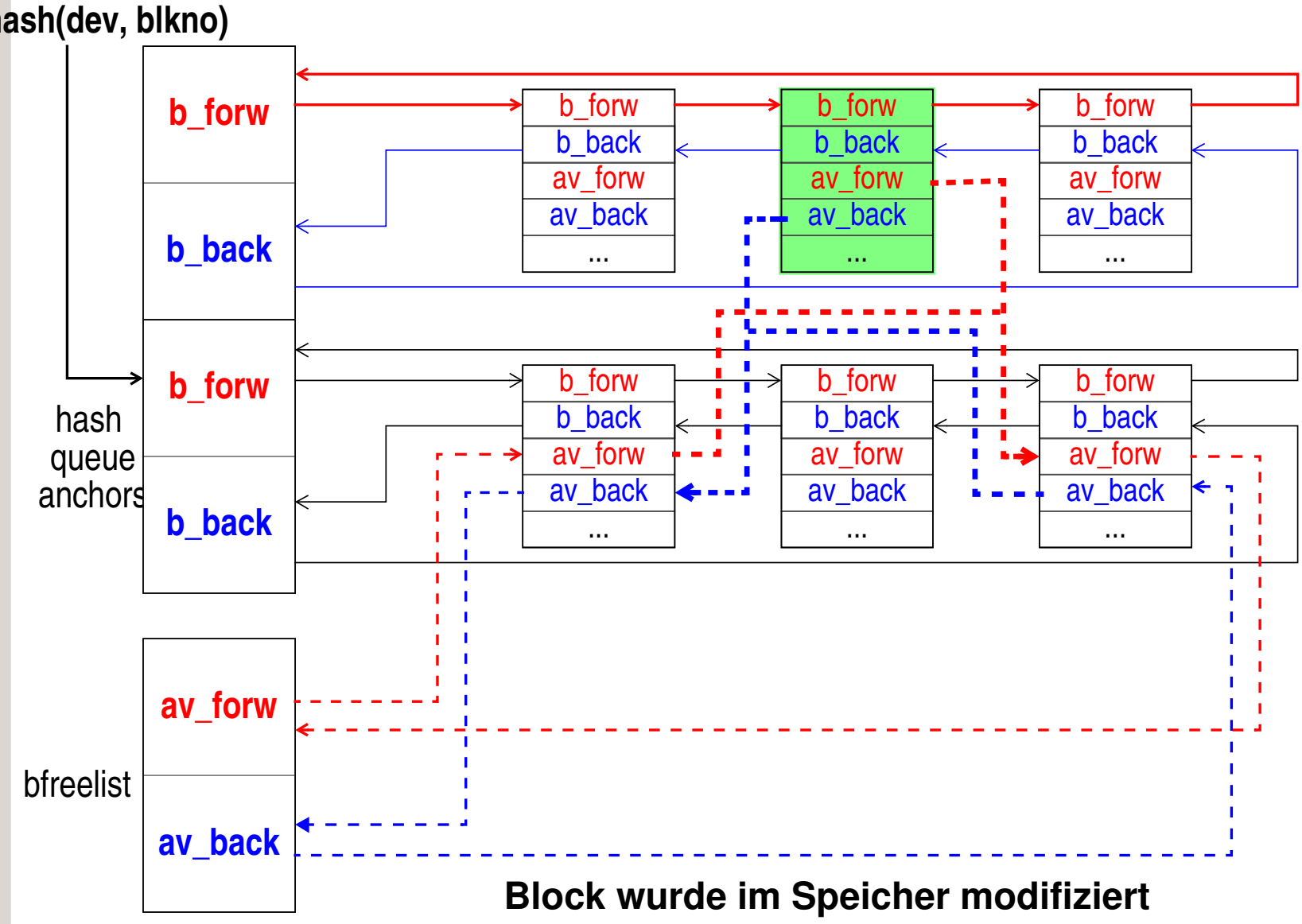



```
fd1 = open("/etc/passwd", O_RDONLY);
fd2 = open("datei1", O_WRONLY);
fd3 = open("/etc/passwd, O_RDWR);
```

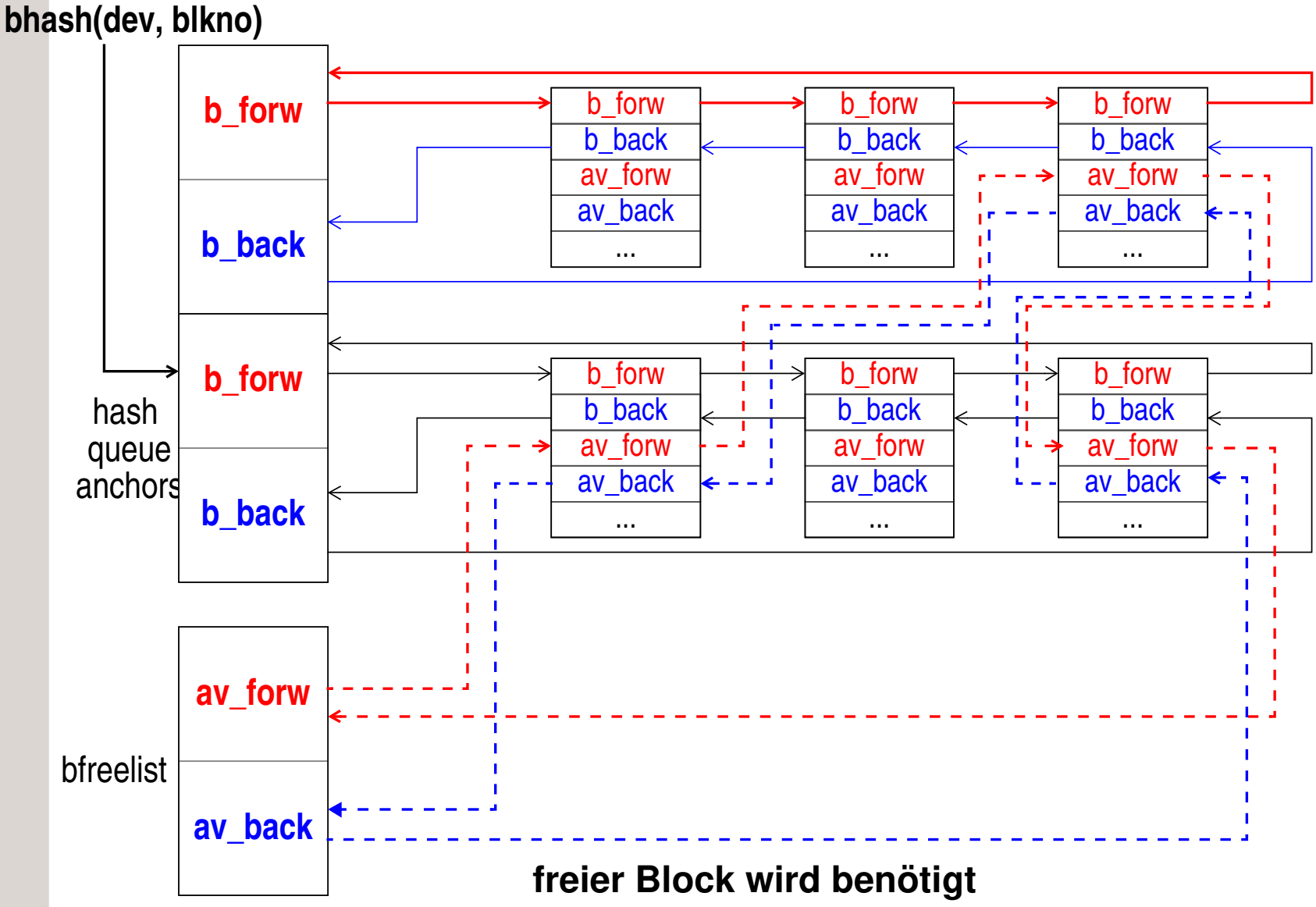

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.



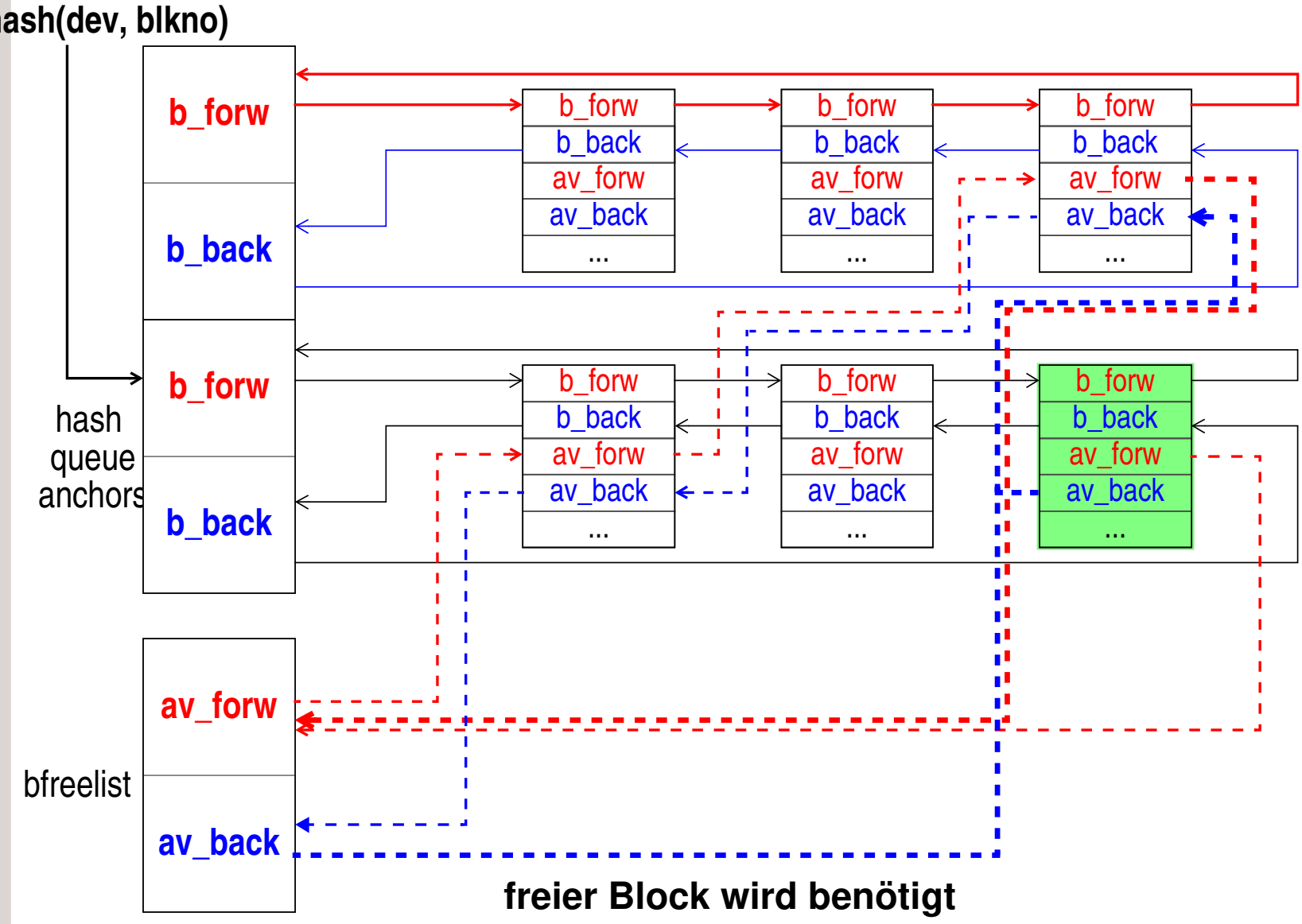
Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.



Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

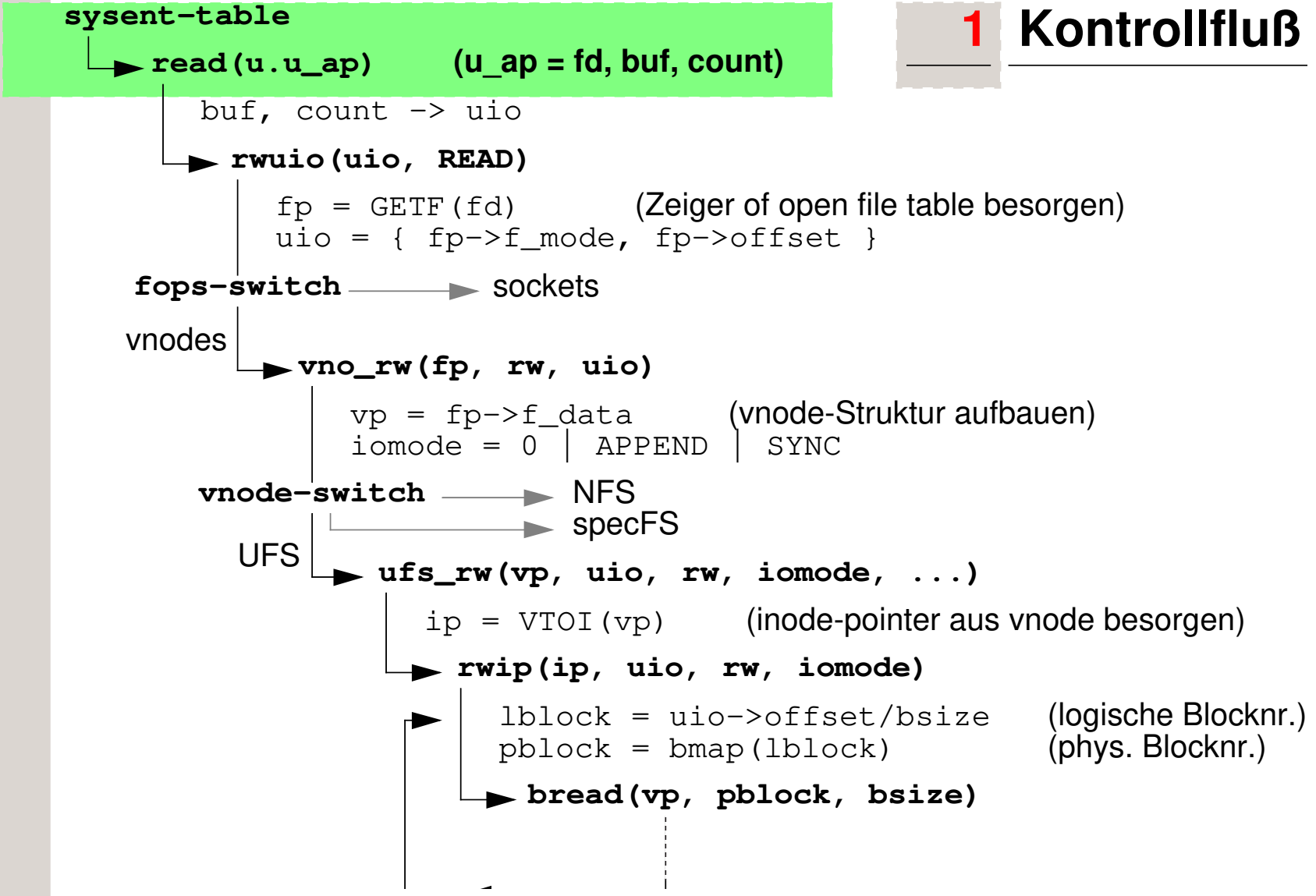


Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.



Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

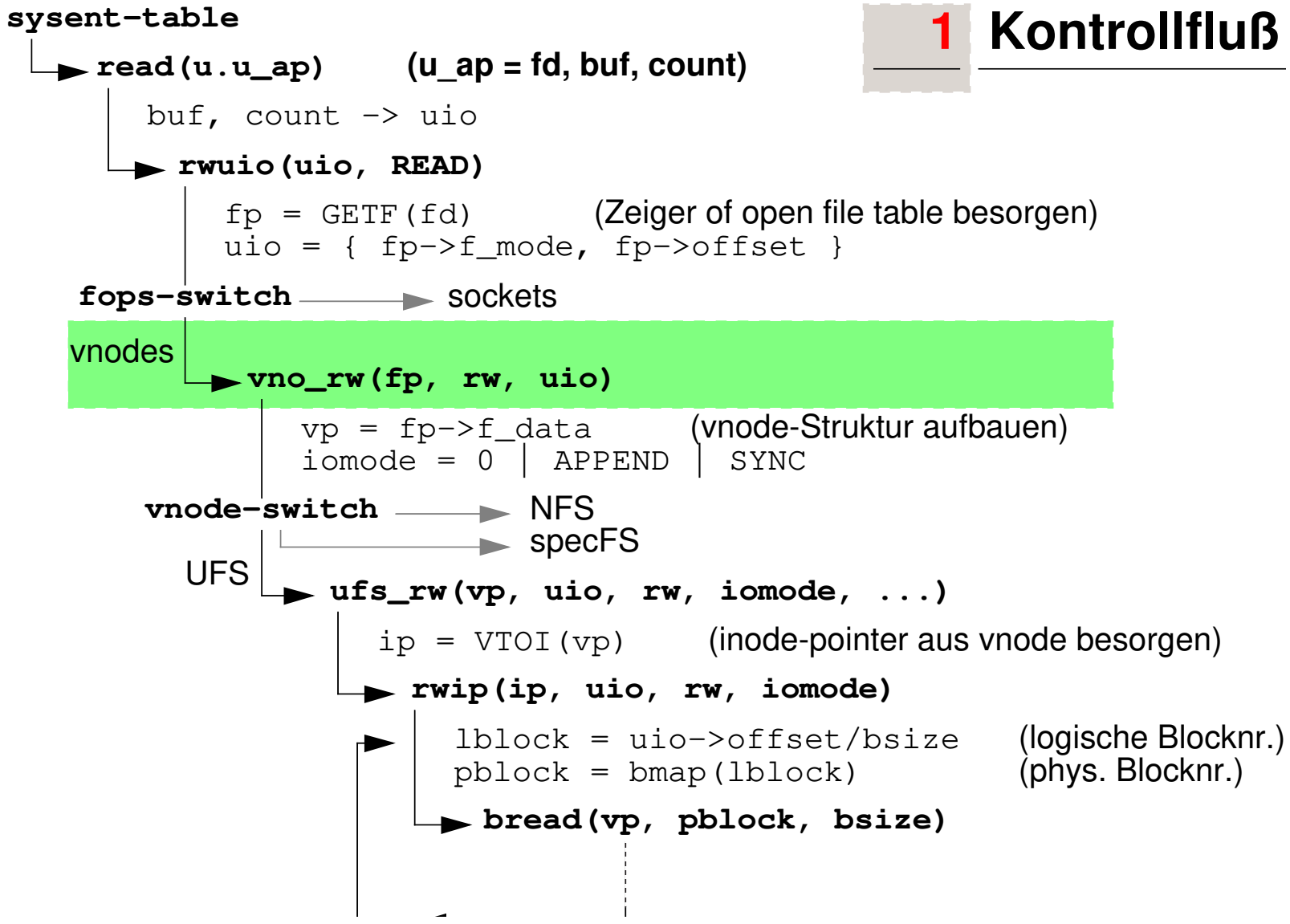
1 Kontrollfluß



Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

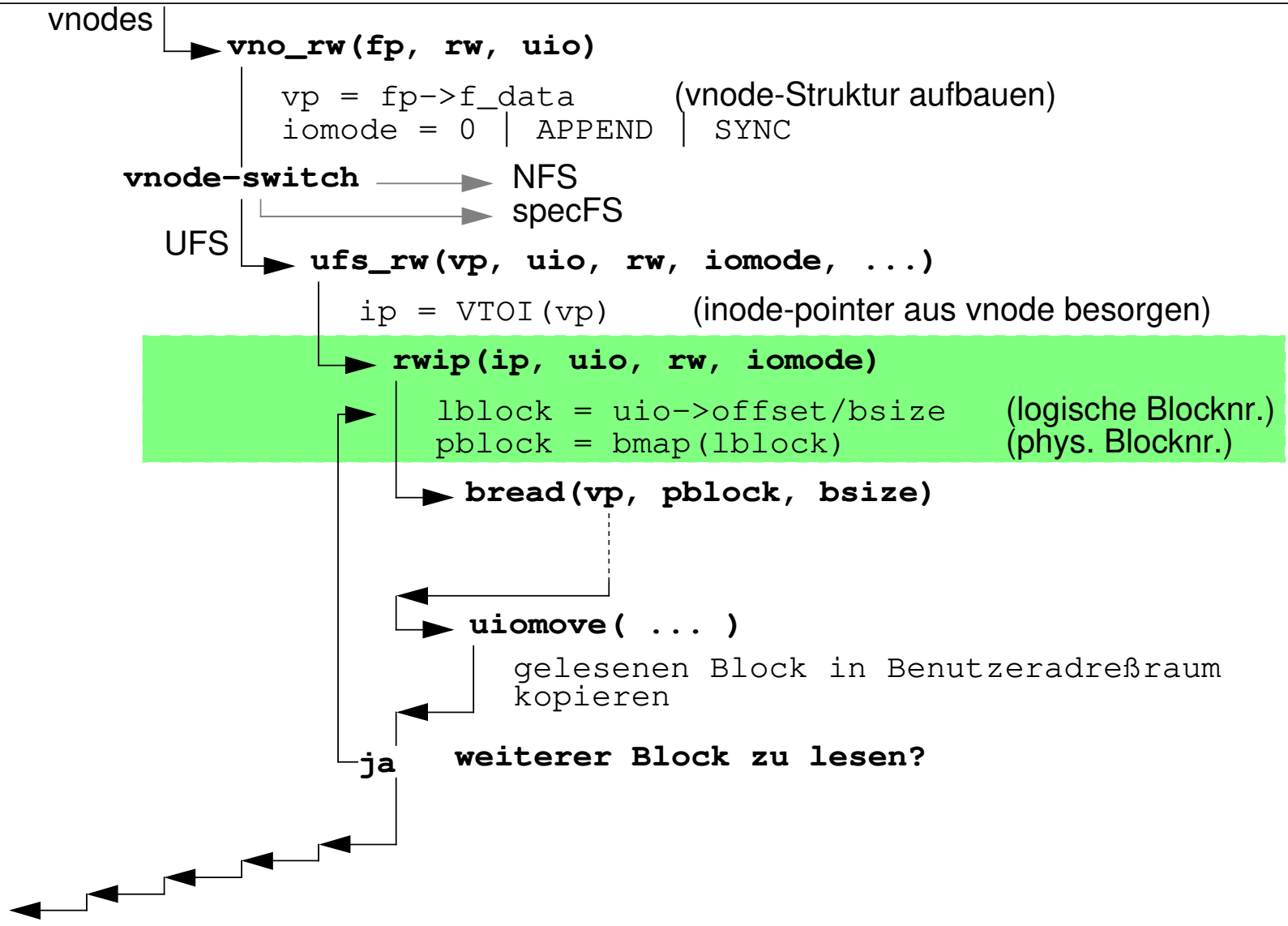
Dateisystem - Buffer Cache

1 Kontrollfluß



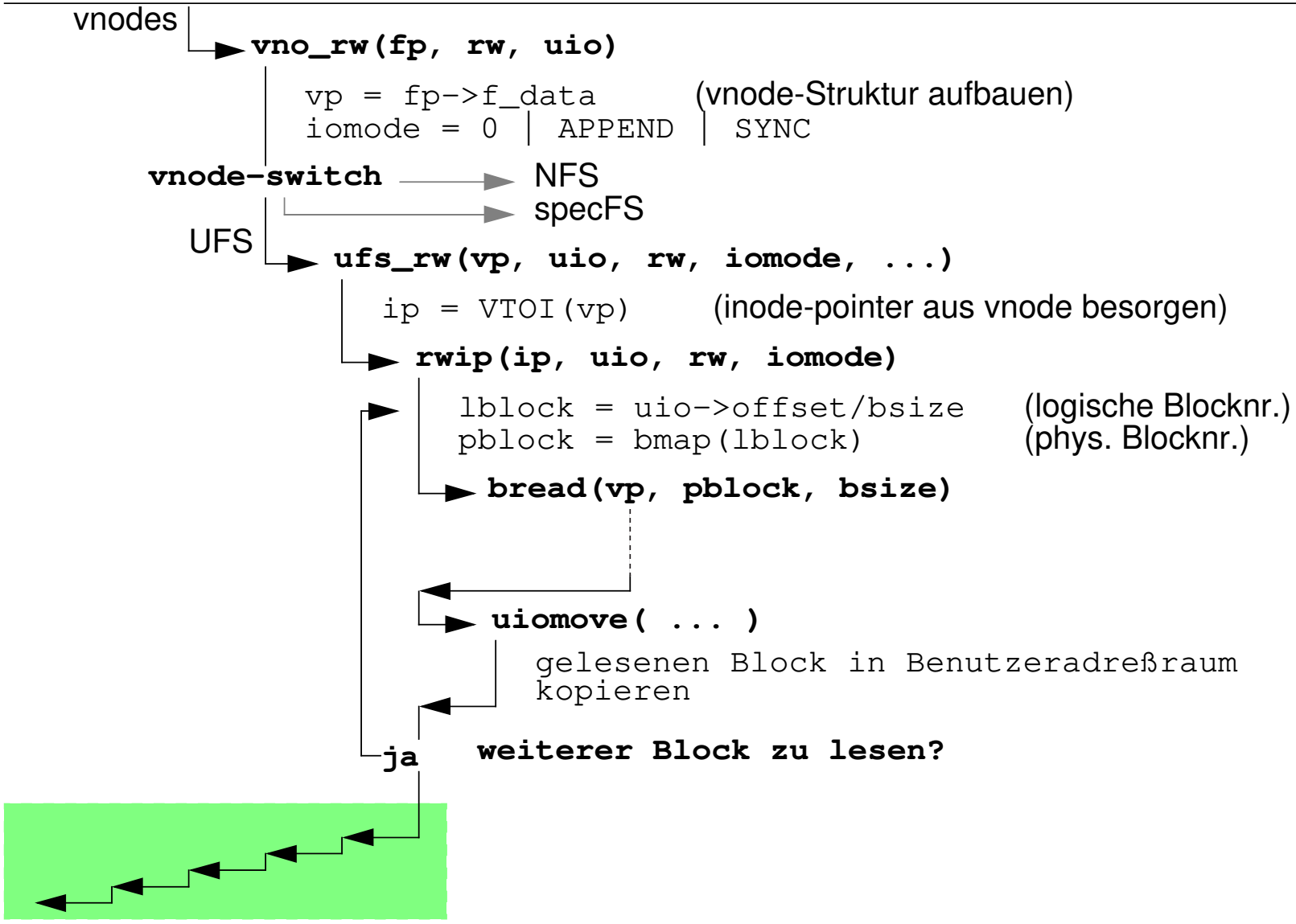
Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

I.9 Dateisystem - Buffer Cache



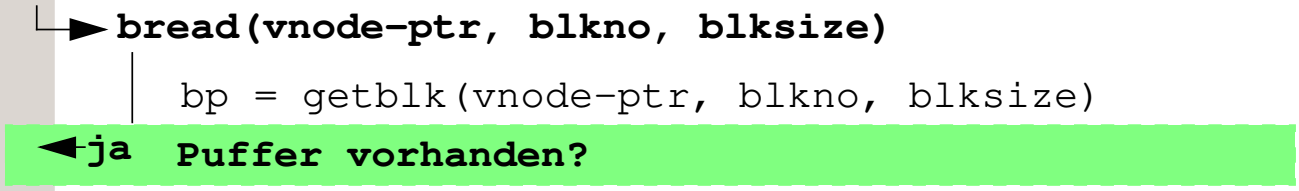
Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

I.9 Dateisystem - Buffer Cache



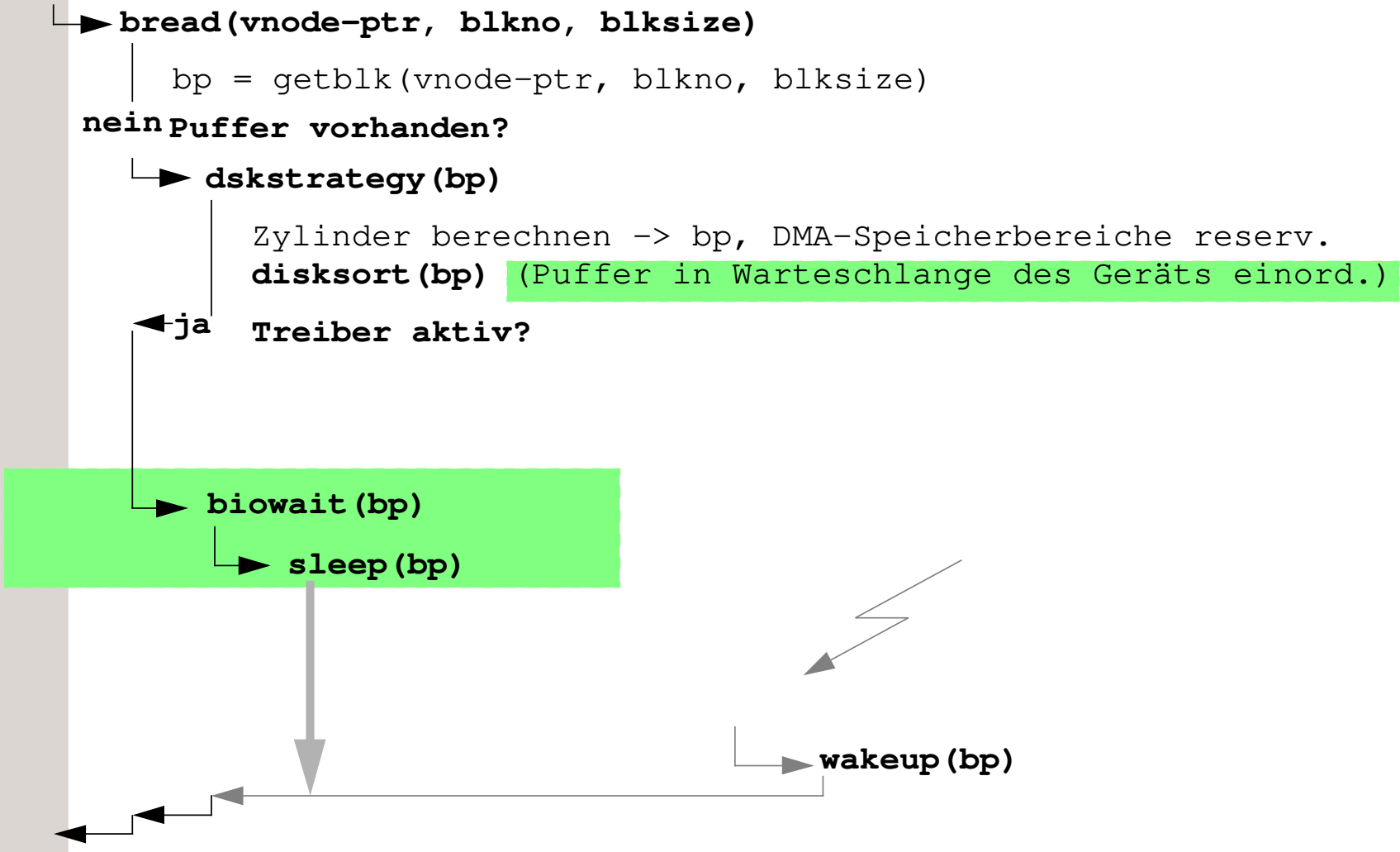
Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

3 Kontrollfluß



Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

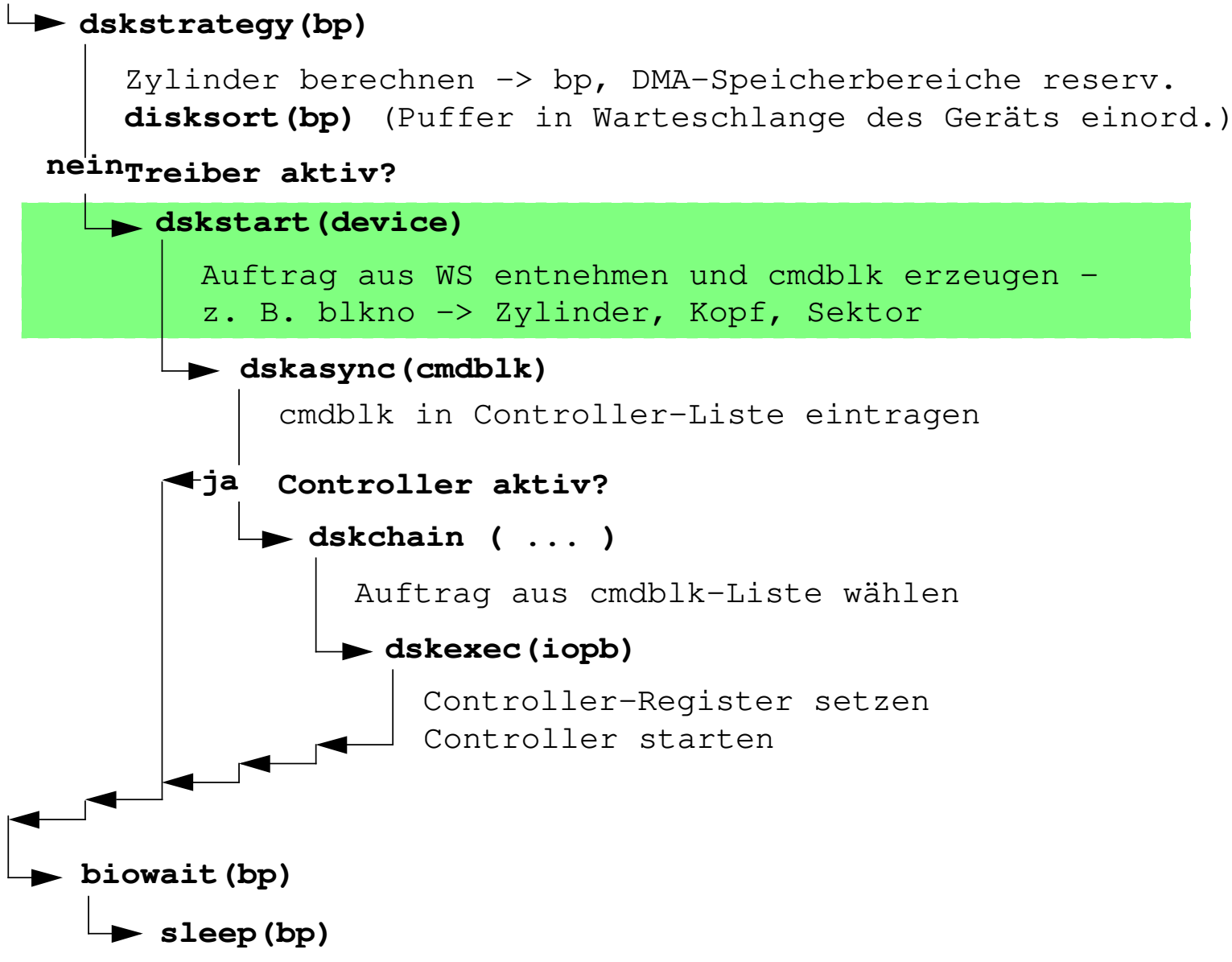
3 Kontrollfluß



© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1998

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

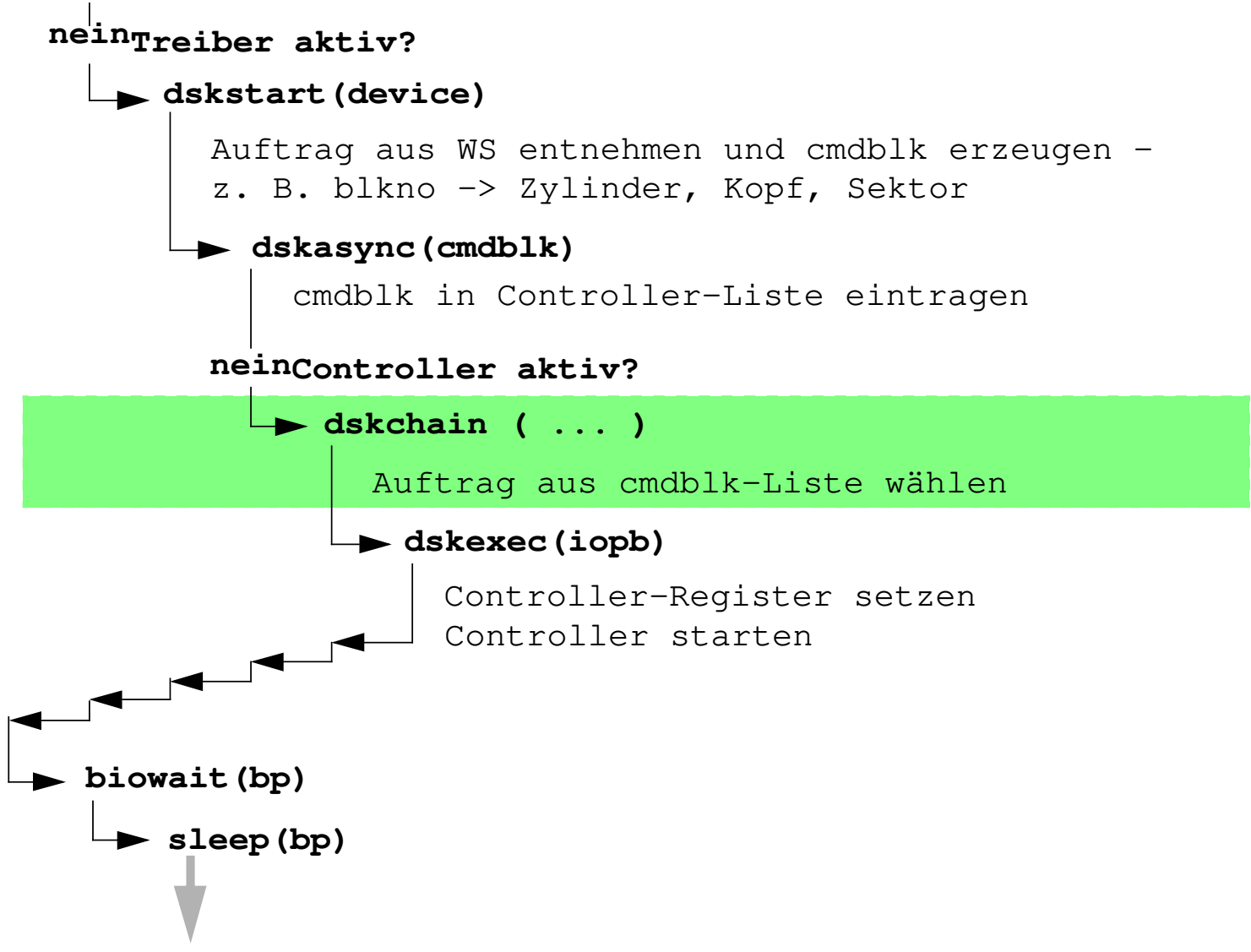
3 Kontrollfluß



© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1998

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

3 Kontrollfluß



© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1998

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

3 Kontrollfluß

dskstart(device)

Auftrag aus WS entnehmen und cmdblk erzeugen –
z. B. blkno -> Zylinder, Kopf, Sektor

dskasync(cmdblk)

cmdblk in Controller-Liste eintragen

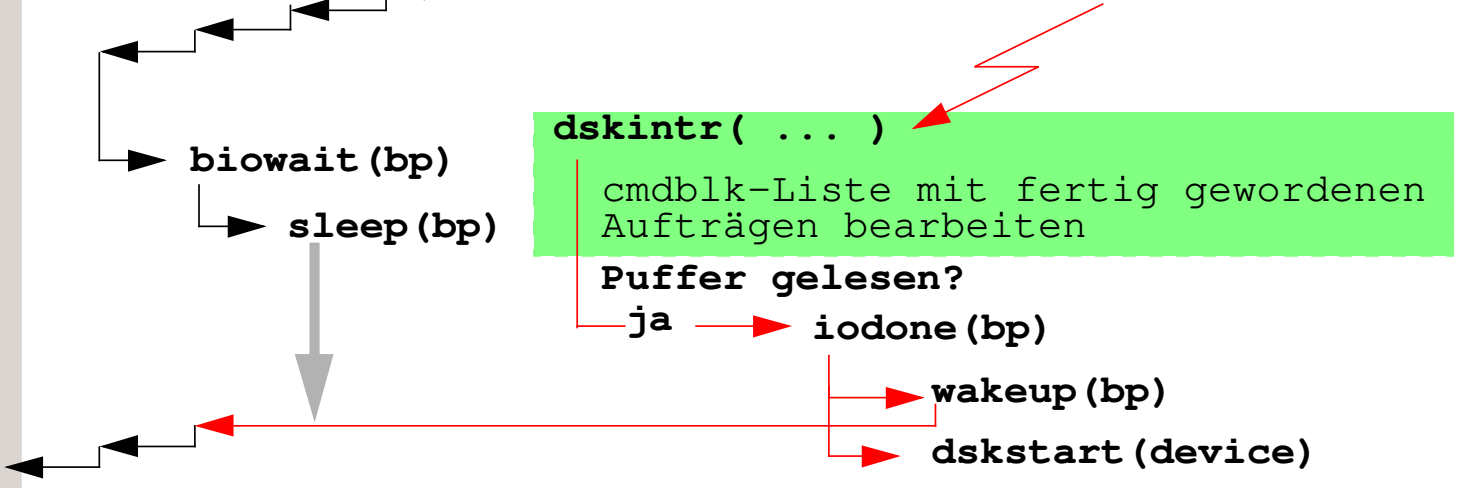
Controller aktiv?

dskchain (...)

Auftrag aus cmdblk-Liste wählen

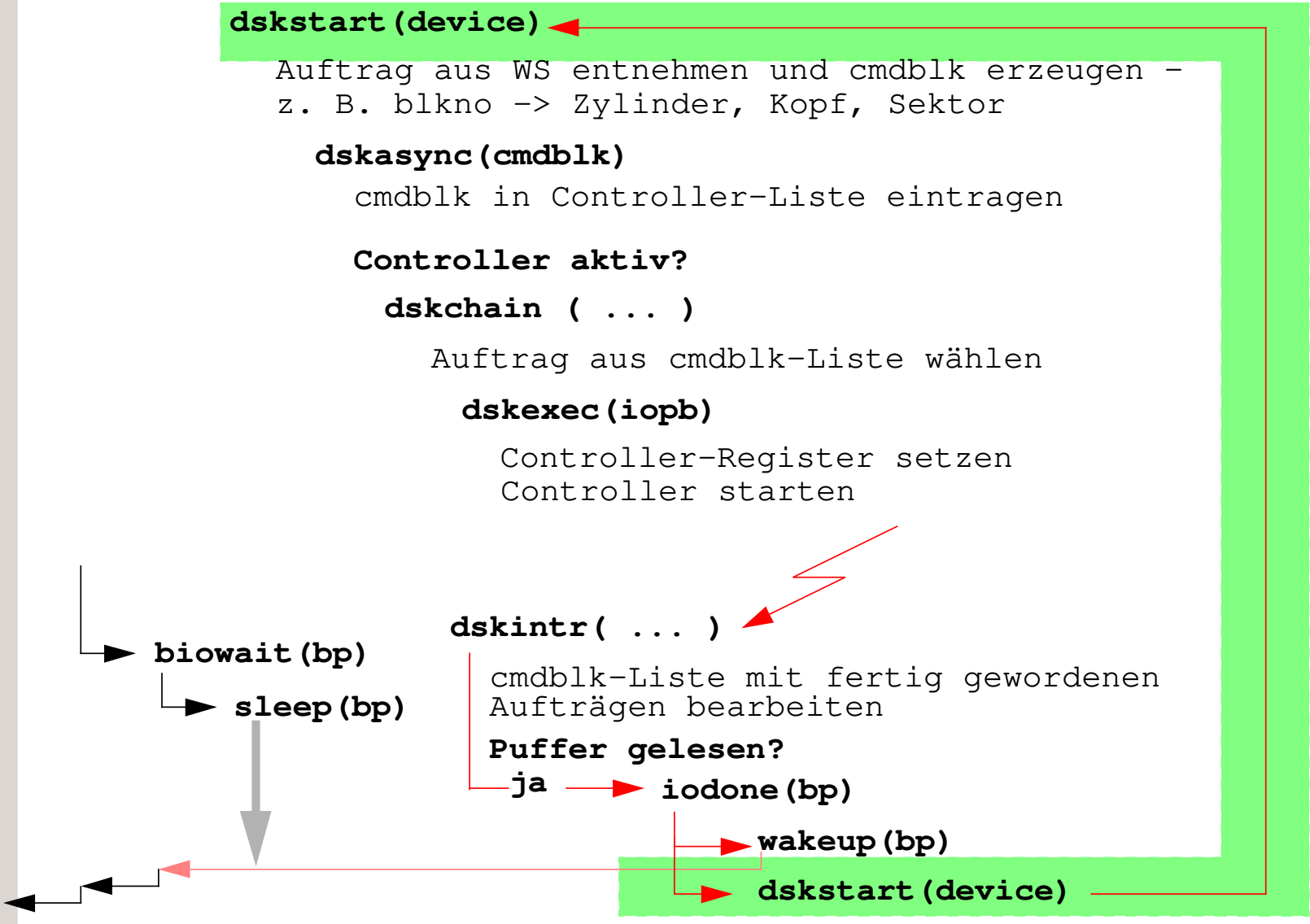
dskexec(iopb)

Controller-Register setzen
Controller starten



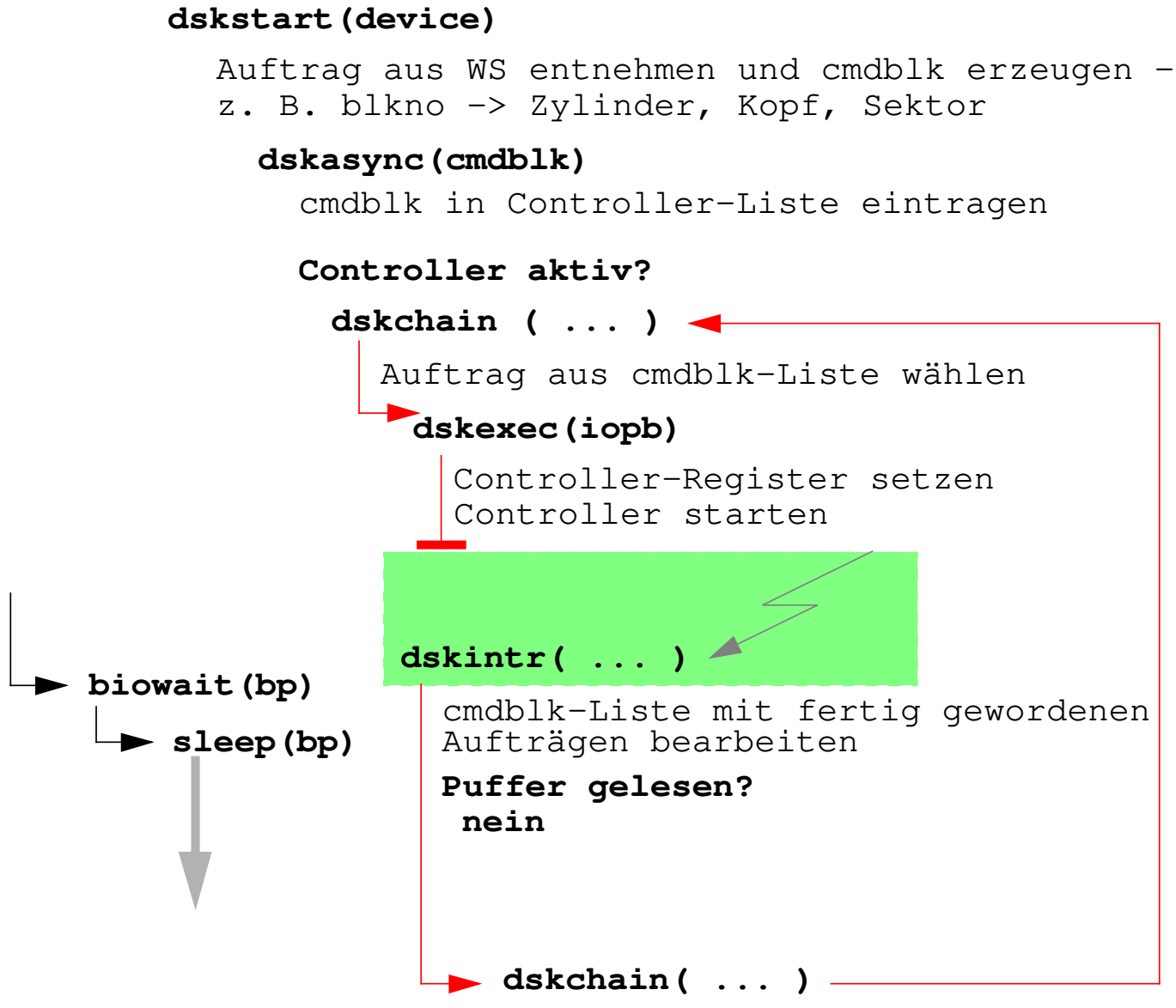
Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

3 Kontrollfluß



Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

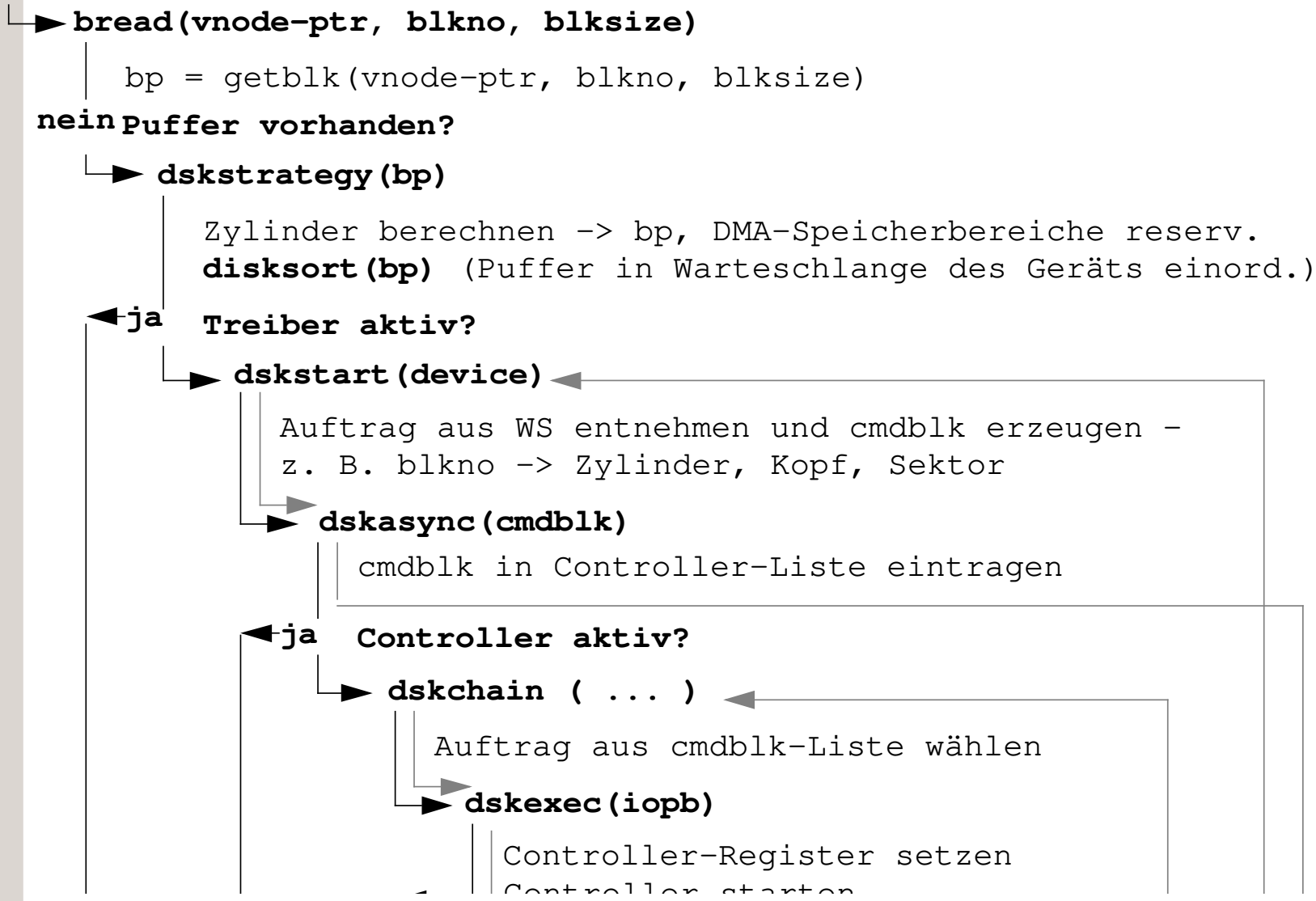
3 Kontrollfluß



Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

3 Kontrollfluß



© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1998

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.