



```

Direkte Verweise auf Datenblöcke
Indirekte Verweise auf Datenblöcke:
  Einfach indirekt
  Zweifach indirekt
  Dreifach indirekt

```

2c) Datei-Inhalt

Der Datei-Inhalt ist in Datenblöcken abgespeichert, die im Dateisystem verstreut liegen.

2d) Beispiel

- \* Aktuelles Verzeichnis ist "/home/tom/".
- \* Im Verzeichnis gibt es 2 Hard-Links namens "gedicht" und "gedicht2" auf die gleiche Datei.
- \* Weiterhin gibt es darin 1 Symbolischen Link namens "symlink", der auf "gedicht" zeigt.

		Verzeichnis-(Liste)		Inode-Tabelle		Daten-Blöcke		
		Name	Nr	Nr	Inode			
#=>	HL1	gedicht	175	...	...	DIE MADE...		
X								
X	HL2	gedicht2	175	175	Attr	...		
X								
X		test	...	...	...	/home/tom/gedicht ==#		
X								
X	SL1	symlink	231	231	Attr	...		
X								
X		doku	...	...	...	...		
X								
X				Hardlink	...	...		
X								
X								
X								
#	=====							#
						Symbolischer Link		

3) Einsatzzweck von Links

- \* Immer den gleichen Namen für ein Gerät zur Verfügung stellen, egal welche konkrete Hardware dahinter steckt (Bsp: "/dev/mouse").
- \* Für den schnellen Zugriff in Benutzer-Verzeichnissen eine "Kopie" einer (tief) im Dateisystem gespeicherten Datei ablegen.
- \* Plattenplatz sparen, wenn die gleiche Datei mehrfach im Dateisystem vorkommen soll (Verweise statt Kopien).
- \* Schreibfehler oder Namensänderungen wegen FHS (File Hierarchy Standard) abfangen (z.B. YaST = yast = zast, conf.modules = modules.conf).
- \* Konsistenz erhalten bei Versionierung, Updates, Konfigurationsdateien (Bsp: /usr/src/linux -> linux-2.2.5.SuSE).
- \* Eine von mehreren Versionen einer Datei oder eines Programms als Standard auswählen (z.B. /usr/bin/bash -> /usr/local/bin/bash-4.0)
- \* UNIX-Besonderheit: Das gleiche Programm/Skript kann unter verschiedenen Namen referenziert werden und verhält sich dann je nach Name, unter dem es aufgerufen wird, verschieden (Bsp: mv = cp = ln, mkfs.ext2 = mke2fs).

Beispiele für mehrere Namen für die gleiche Datei sind:

UNIX	Kommandos	Grund
SuSE-Linux	Yast2 yast2 zast yast1 yast	Schreibfehler abfangen
Solaris	cp mv ln	Programmierung erleichtern
Linux	gunzip gzip zcat	Programmierung erleichtern
Solaris	/usr/ucb/ps /usr/bin/ps	BSD + SYSV-Verhalten gleichz.

4) Link-Arten Hard-Link und Symbolischer Link

-----  
 UNIX unterscheidet 2 Arten von Links, die aber beide für den gleichen Zweck eingesetzt werden, nämlich zusätzliche Namen für eine einzelne Datei zur Verfügung zu stellen:

- 1) Hard-Link: Physikalischer Link (per I-Node)
- 2) Soft-Link: Symbolischer Link (per Name)

1) Hard-Links gibt es --- wie oben beschrieben --- für jede Datei mindestens einen, beliebig viele weitere können folgendermaßen angelegt werden:

```
ln DATEI HARDLINK      # Analog "cp DATEI KOPIE"
```

Ihre Verwendung ist in einigen Punkten eingeschränkt (z.B. nur innerhalb einer Partition einsetzbar).

2) Symbolische Links (auch Soft-Links) werden folgendermaßen angelegt:

```
ln -s DATEI SOFTLINK  # Analog "cp DATEI KOPIE"
```

und verweisen per (Pfad)Name auf eine andere Datei. Der Pfadname kann relativ oder absolut sein.

Man kann sich das so vorstellen, dass eine Datei als Wert den (Pfad)Namen einer anderen Datei enthält. Erst beim Zugriff auf einen Symbolischen Link wird festgestellt, ob seine Zielfeile wirklich existiert --- und wenn ja --- diese benutzt. Sie sind mit HTML-Links oder Windows-Verknüpfungen zu vergleichen (Verweis per URL = Name auf eine andere HTML-Seite).

Die beiden Link-Typen können folgendermaßen charakterisiert werden:

	Hard Physical	Soft Symbol.
Referenz per	I-Node	Name
Anzahl Links auf Datei bekannt	Ja	Nein
Über Partitionen hinweg möglich	Nein	Ja
Ziel kann ein Verzeichnis sein	Nein	Ja
Rekursion möglich (Verz. enthält sich selbst)	Nein	Ja
Ziel muß existieren (kann nicht ins Leere zeigen)	Ja	Nein
Geschwindigkeit	schnell	langsam
Unter WINDOWS vorhanden (Verknüpfung)	Nein	Ja

5) Vergleich mit Windows

-----  
 Symbolische Links gibt es unter Windows nur in Form der "Desktop-Icons" = Verweise auf Programme, Daten oder Verzeichnisse. Allerdings können diese wirklich nur auf dem Desktop liegen, innerhalb von Verzeichnissen sind sie nicht möglich. D.h. dieser Link-Typ ist unter Windows nur eingeschränkt verfügbar.

Hard-Links gibt es unter Windows erst mit dem Dateisystem NTFS. Zu ihrer Verwaltung gibt es aber keine grafische Möglichkeit im Windows Explorer, sondern sie sind nur per Kommandos nutzbar.