

Dateisystem - Links

- **Dateien** bestehen unter UNIX aus **3 Komponenten**:

Name (z.B. gedicht)
Inode = Verwaltungseinheit (enthält z.B. Besitzer, tom, letzte Änderung: 1.1.1999 8:45, Zugriffsrechte, ...)
Inhalt (z.B. DIE MADE... HEINZ ERHARD)

- Abgelegt werden diese 3 Informationen im UNIX-Dateisystem an **3 getrennten Stellen**:
 - Namen in **Verzeichnissen**.
 - Inodes in **Inode-Tabellen**.
 - Inhalte in **Datenblöcken**.

Dateisystem - Links

- Ein **Verzeichnis** enthält eine **Liste von Dateinamen** und zugehöriger **Inode-Nummer**, z.B.

gedicht	123
erster	124
zweiter	130
esst	144

- Verzeichnisse sind also nur **spezielle Dateien**.
- Inode-Nr zu Dateien/Verz. erhält man mit `ls -li`.

Dateisystem - Links

- Ein **Inode (Index/Informations-Knoten)** enthält alle Datei-Eigenschaften (*Attribute*) außer den Dateinamen und den Datei-Inhalt:

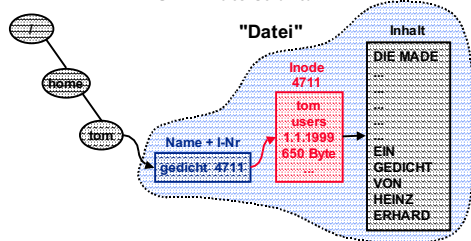
Datentyp
Zugriffsrechte
Hard-Link-Zähler
Besitzer + Besitzer-Gruppe
Dateigröße
Datum, ... (3 verschiedene)
Verweise auf die Datenblöcke

Dateisystem - Links

- Der **Datei-Inhalt** ist auf dem Datenträger (z.B. Festplatte) verstreut in **Datenblöcken** abgelegt.
 - Einige Einträge im Inode zeigen auf diese Datenblöcke.
- Diese etwas komplizierte Struktur wird vom UNIX-System **selbständig verwaltet**:
 - Die Anwender müssen sich nicht darum kümmern.
 - Zugriff auf Dateien erfolgt immer über **Namen**.
 - Das UNIX-System **übersetzt** die Namen in Inode-Nummern und Datenblock-Adressen.

Dateisystem - Links

UNIX-Dateistruktur



Dateisystem - Links

- Unter UNIX ist **mehr als ein Name** für eine Datei vergebbar.

- Dafür gibt es zwei prinzipielle Verfahren:

A) Hard-Links (physikalische Links): Verweise auf Inodes
B) Soft-Links (symbolische Links): Verweise auf Dateinamen

- **Hinweis:**
Bereits der "erste" **Dateiname** ist ein Hard-Link!

Dateisystem - Links

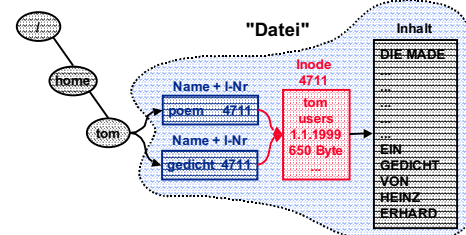
A) Hard-Link (Physikalischer Link)

- Weiterer Verzeichniseintrag zu einer Datei, der auf **denselben Inode** zeigt.
 - Kann **keine Partitions Grenzen** überschreiten (da Inode-Nummern nur pro Partition eindeutig).
 - Kann nur auf **normale Dateien** verweisen (nicht auf Verzeichnisse und Geräte).
 - **Alle Hard-Links einer Datei sind völlig gleichberechtigt.**
 - `ls -l` gibt die **Anzahl Hard-Links** in der 2. Spalte aus.

Die **Anzahl Hard-Links** auf eine Datei wird im **Inode** gezählt, erst beim Löschen des letzten wird ihr **Inhalt** freigegeben.

Dateisystem - Links

UNIX-Dateistruktur



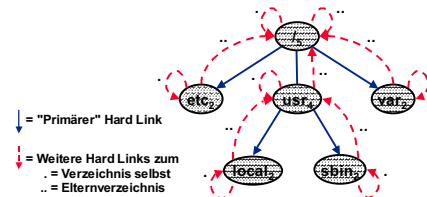
Dateisystem - Links

Hard-Links "." und ".." bei Verzeichnissen

- Werden von UNIX selbst verwaltet.
 - In jedem Verzeichnis vorhanden.
- Zeigt auf das Verzeichnis selbst
Zeigt auf sein Eltern-Verzeichnis
- D.h. jedes Verzeichnis hat mindestens **2 Namen**, sein Hard-Link-Zähler hat daher mindestens den Wert 2.
 - Für jedes **Unterverzeichnis** darin kommt ein Hard-Link für den Namen "." im Unterverzeichnis dazu.

Dateisystem - Links

Hard-Links "." und ".." bei Verzeichnissen



Dateisystem - Links

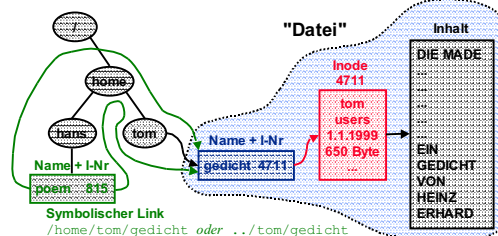
B) Symbolischer Link (Soft-Link)

- Weiterer Verzeichniseintrag zu einer Datei, der über einen **Pfadnamen** (absolut / relativ) auf sie verweist.
 - Kann **Partitions Grenzen** überschreiten.
 - Kann auf **alle Dateitypen** verweisen (insb. Verzeichnisse), d.h. im Dateibaum sind "**Rekursionen**" möglich.
 - **Ziel** muß nicht existieren (kann ins "Leere" zeigen).
 - Entspricht **Windows-Verknüpfung** oder **HTML-Link**.

Die **Anzahl Soft-Links** auf eine Datei wird **nicht** gezählt.

Dateisystem - Links

UNIX-Dateistruktur



Dateisystem - Links

Zweck von Links

Einfacher Dateizugriff vom Heimat-Verzeichnis aus.
Plattenplatz sparen (*nur eine gemeinsame Datei*).
Konsistenz bei Datenupdates (*analog*).
Schreibfehler / geänderte Konventionen abfangen.
Gültige Version aus mehreren vorhandenen auswählen.
Immer gleichen Namen für ein Gerät festlegen.
Programme / Skripte können sich abhängig vom Aufruf-
Namen verschieden verhalten (*UNIX-Besonderheit*).

Dateisystem - Links

Eigenschaft	Hard	Symbolic
Referenz per	Inode-Nr	Name
Anzahl Links bekannt	✓	–
Über Partitionen hinweg	–	✓
Ziel muß existieren	✓	–
Ziel kann Verzeichnis sein	–	✓
Rekursion erlaubt	–	✓
Geschwindigkeit	schnell	langsam
Platzbedarf	keiner	etwas
Unter Windows vorhanden	–	Verknüpf.
Einfach erkennbar	–	✓

Dateisystem - Links

- `ln` (**link**) legt **Links** auf Dateien an (**symbolic**):

```
ln /PFAD/ZU/DATEI LINKNAME (Hard Link)
ln -s /PFAD/ZU/DATEI LINKNAME (Symbolic L.)
```

- **Tipp:** Form analog `cp` (*kopiert nur Dateinamen*).

- `rm` (**remove**) löscht zunächst nur Links:

```
rm [OPTIONEN] LINK...
```

Die "eigentliche" Datei wird erst dann gelöscht,
wenn der **letzte Hard-Link** entfernt wird.

Dateisystem - Links

Tipps zu symbolischen Links

- **In das Verzeichnis gehen**,
in dem ein symbolischer Link angelegt wird.
- **Absolute Pfade** beim Anlegen verwenden,
wenn man sich unsicher ist.
- **Relative Pfade** bieten zwar Vorteile, sind aber
wesentlich schwerer korrekt zu formulieren!
 - Z.B. ist ein Verzeichnisbaum mit relativen symbolischen
Links **verschiebbar**, ohne sie ungültig zu machen.
 - **UNIX-interne** symbolische Links sind meist relativ.

Dateisystem - Links

Hinweis

- Nach dem Anlegen eines **symbolischen Links** mit

```
ln -s /home/tom/datei symlink
```

zeigt `ls -l symlink` folgendes an:

```
symlink->> /home/tom/datei
```

- D.h. bei `ln` lautet die "**Richtung**" des Links:

Datei → Link-Name (*klassisch: Quelle → Ziel*)

Bei `ls -l` wird **anders herum** dargestellt:

Link-Name → Datei

Dateisystem - Links

Zusammenfassung

```
ls -i Inode-Nummer einer Datei ausgeben.
ln Hard-Link auf eine Datei anlegen (link).
ln -s Symbolischen Link auf eine Datei anlegen.
rm Hard- oder Symb. Link löschen (remove).
```

Die **Anzahl Soft-Links** auf eine Datei wird **nicht** gezählt.