

Systemprogrammierung

Grundlagen von Betriebssystemen

Teil B – V.3 Betriebssystemkonzepte: Dateien, Namensräume

12. Juni 2025

Rüdiger Kapitza, Jürgen Kleinöder

(© Wolfgang Schröder-Preikschat, Rüdiger Kapitza)



Lehrstuhl für Informatik 4
Systemsoftware



Friedrich-Alexander-Universität
Technische Fakultät

Einführung

Datei

Außensicht

Innensicht

Namensräume

Dateisystem

Zusammenfassung

Einführung

Datei

Außensicht

Innensicht

Namensräume

Dateisystem

Zusammenfassung

- **Datei** als Abstraktion von der tatsächlichen Speicherung von Daten
- **Hierarchische Namensräume** zur Benennung von Dateien und anderen Objekten
- Aufbau von **Dateisystemen**
- **Dateiverzeichnisse** zur Gruppierung von Dateien

Einführung

Datei

Außensicht

Innensicht

Namensräume

Dateisystem

Zusammenfassung

Datei als Betriebssystemabstraktion

- Datei (engl. *file*) Sammlung von Daten, eine...
 - zusammenhängende, abgeschlossene Einheit von Daten
 - meist „beliebige“ Anzahl eindimensional adressierter Bytes
- Abstraktion von der tatsächlichen Speicherung
 - Dauerhaftigkeit von Dateien ist eine Frage des Speichermediums:

nicht-flüchtige Datenträger	Platte, SSD, Band, CD, DVD, ...
flüchtige Datenträger	RAM

- Definition einer unabhängigen API für den Zugriff
 - z. B. für Geräte oder auf Kommunikationsmittel für kooperierende Prozesse

Datei

Außensicht

■ ausführbare Dateien: Binär- und Skriptprogramme

- von einem Prozessor ausführbarer **Programmtext**

Binär $\leadsto$ CPU, FPU, MCU, JVM, ..., Basic, Lisp, Prolog

Skript $\leadsto$ perl(1), python(1), {a,ba,c,tc}sh(1), tcl(n)

- der Prozessor liegt in Hard-, Firm- und/oder Software vor

■ nicht-ausführbare Dateien: Text-, Bild- und Tondaten

- von einem Prozessor verarbeitbare **Programmdaten**

– .{doc, fig, gif, jpg, mp3, pdf, tex, txt, wav, xls, ...}

– .{a, c, cc, f, F, h, l, o, p, r, s, S, y, ...}

- der Prozessor liegt in Form von Programmtext vor

- Dateien sind „von außen“ über symbolische Adressen erreichbar...
 - **benutzerdefinierter Name** von beliebiger aber maximaler Länge
 - Dateiname (engl. *file name*)
 - wird ggf. vom Betriebssystem (teilweise) interpretiert
- ...„nach innen“ besitzt jede Datei eine numerische Adresse
 - **systemdefinierte Kennung** einer Datenstruktur der Dateiverwaltung
 - identifiziert den sogenannten Dateikopf (engl. *file head*)

Tupel

- symbolische und numerische Dateiadresse bilden ein (festes) Paar

Dateinamensuffix (engl. *file extension*): eine meist durch einen Punkt vom Dateinamen abgegrenzte **symbolische Erweiterung** des Dateinamens

- liefert einen Hinweis auf das Dateiformat bzw. den Dateitypen

.doc	}	Textdokumente	{	MS-Word	
.tex				$\text{\LaTeX}$	latex(1)
.h	}	Programme	{	Präprozessor	cpp(1)
.c				Kompilierer	cc(1)
.s				Assembler	as(1)
.o				Binder	ld(1)

- ist Dienstprogrammen und/oder dem Betriebssystem bekannt
 - bei UNIX die Dienstprogramme, bei Windows das Betriebssystem

Datei

Innensicht

- „*inode*“: Dateikopf in UNIX (enthält Dateiattribute)
- Eintrag in der „*Master-File-Table*“: Dateikopf in Windows (enthält Attribute und ggf. auch Inhaltsdaten)
 - Beispiele für **Dateiattribute**
 - Eigentümer (*user ID*)
 - Gruppenzugehörigkeit (*group ID*)
 - Typ (reguläre/spezielle Datei)
 - Zugriffsrechte (lesen, schreiben, ausführen, ...)
 - Zeitstempel (letzter Zugriff, letzte Änderung [Typ, Zugriffsrechte])
 - Anzahl der Verweise („*hard links*“)
 - Größe (in Bytes)
 - Adresse(n) der Daten auf dem Speichermedium
- „*inode number*“ oder „*File Reference*“: Dateikopfnummer
 - Index in eine Tabelle von Dateiköpfen
 - die **numerische Adresse** der Datei (innerhalb des Dateisystems)

reguläre Datei (engl. *regular file, ordinary file*)

- problemorientiertes, eindimensionales Bytefeld

spezielle Datei ein „Sammelsurium“ von verschiedenen Objekten:

- **Verzeichnis** (engl. *directory*)
 - Katalog von regulären und/oder speziellen Dateien
- **Gerätefile** (engl. *device file*)
 - Zugang zu zeichen-/blockorientierten Geräte(treiber)n
- **symbolische Verknüpfung** (engl. *symbolic link*)
 - Abbildung eines Dateinamens auf einen Pfadnamen
- Kommunikationskanäle (*named pipe*) und Kommunikationsendpunkte (*socket*)

Einführung

Datei

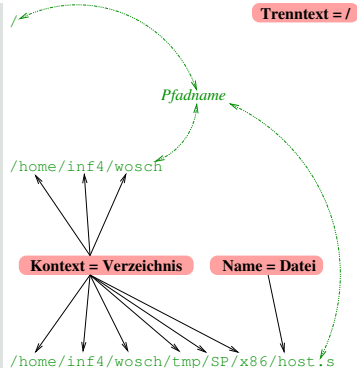
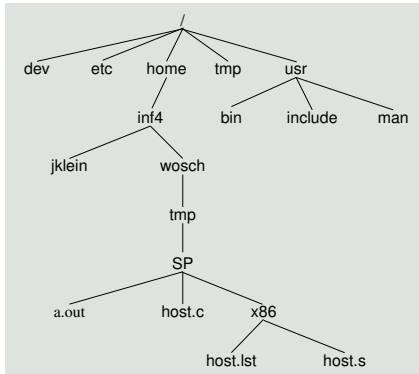
Außensicht

Innensicht

Namensräume

Dateisystem

Zusammenfassung



- ein **Kontext** repräsentiert einen Namensraum flacher Struktur
 - darin muss Eindeutigkeit mit der Namenswahl selbst gewährleistet sein
- im Gegensatz zu einem Namensraum hierarchischer Struktur (s. o.)
 - derselbe Name (`tmp`) kann in verschiedenen Kontexten definiert sein
 - durch den Pfadnamen wird sein jeweiliger Standort eindeutig gemacht

- fundamental für die Hierarchiebildung ist das **Verzeichnis**
 - es ordnet einen/mehrere Namen, auch Verzeichnisnamen, listenförmig
 - ist namentlich selbst in einem **Elternverzeichnis** (*parent directory*) gelistet
 - es gibt mehreren Namen ein gemeinsames Merkmal, denselben Kontext
 - bspw. Standort/Bezugspunkt innerhalb des Namensraums, Zugriffsrechte
 - es dient der Umsetzung von symbolischen in numerischen Adressen

Verzeichniseintrag (*directory entry*)

Speichert die Abbildung eines Namens auf eine Informationsstruktur.

- UNIX-artige Betriebssysteme bieten zudem vordefinierte Kontexte:
 - **Wurzelverzeichnis** (*root directory*) des Systems¹
 - Einstiegspunkt in, aber auch Anknüpfungspunkt für, den Namensraum
 - **Heimatverzeichnis** (*home directory*) eines autorisierten Benutzers
 - initiales Arbeitsverzeichnis nach erfolgter Anmeldung (*login*)
 - **Arbeitsverzeichnis** (*working directory*) eines Prozesses
 - gegenwärtiger, relativer Standort des Prozesses im Namensraum

¹Womit der Namensraum in einen bestehenden anderen Namensraum an einem Befestigungspunkt (*mount point*) gegebenenfalls eingebunden werden kann.

Einführung

Datei

Außensicht

Innensicht

Namensräume

Dateisystem

Zusammenfassung

Dateisystem (engl. *file system*)

- Teil des Betriebssystems, um ...
 - Dateien auf Speichersystemen zu organisieren und zu verwalten
 - Bereitstellung eines (hierarchischen) Namensraums
 - Dateiverzeichnisse zur Abbildung von Dateinamen auf interne Verwaltungsstrukturen (Dateikopf)
 - Realisierung der Schnittstelle zwischen Betriebssystem und (Hintergrund-)Speicher
 - Bereitstellung einer API für den Zugriff durch Anwendungsprogramme
 - Realisierung von Schutzfunktionen (Zugriffs- und Rechteverwaltung)

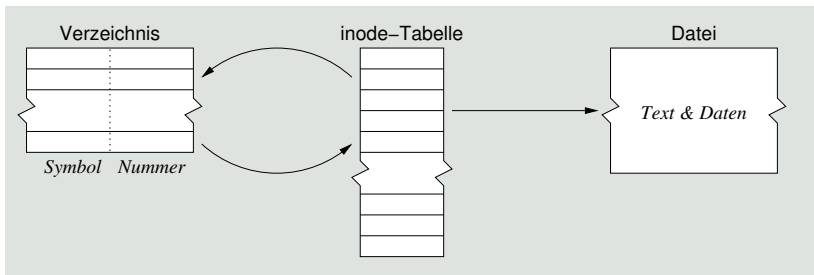
Katalog (engl. *catalogue*, *directory*) von symbolischen Namen

- definiert einen gemeinsamen **Kontext**
 - symbolische Adressen sind nur innerhalb ihrer Kontexte eindeutig
- implementiert eine „**Umsetzungstabelle**“:



- speichert die Abbildung Dateiname $\mapsto$ Dateikopfnummer

Zusammenhang der Datenstrukturen im Namensraum



- die **inode-Tabelle** (*inode table*) ist ein statisches Feld (*array*) von *inodes* und die zentrale Datenstruktur
 - ein *inode* ist **Deskriptor** insb. eines Verzeichnisses oder einer Datei
- das **Verzeichnis** (*directory*) ist eine **Abbildungstabelle**, es übersetzt symbolisch repräsentierte Namen in *inode*-Nummern
 - eine von der Namensverwaltung des Betriebssystems definierte Datei
- die **Datei** (*file*) ist eine abgeschlossene Einheit zusammenhängender Daten beliebiger Repräsentation, Struktur und Bedeutung

Verzeichniseintrag I

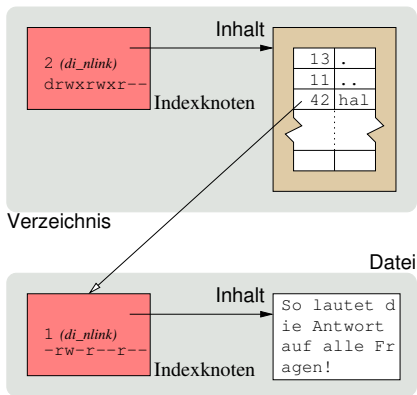
- die **feste Verknüpfung** (*hard link*) von einem Dateinamen mit einer *inode*-Nummer (UNIX V7, *dir.h*):

```
1  typedef unsigned short ino_t;
2
3  #define DIRSIZ 14
4
5  struct direct {
6      ino_t d_ino;
7      char  d_name[DIRSIZ];
8  };
```

- eine als **Wertepaar** gespeicherte **surjektive Abbildung**
- mehrere Paare können auf den selben *inode* (*d_ino*) zeigen
- im selben Verzeichnis jedoch mit verschiedenen Namen (*d_name*)
- in einem *inode* ist die Anzahl der auf ihn verweisenden Wertepaare desselben Namensraums gespeichert (*reference counter*)
- alle *inodes* eines Namensraums sind in einer **inode-Tabelle** in diesem Namensraum gespeichert
 - *d_ino* ist der **Indexwert** eines Verzeichniseintrags für diese Tabelle
- Anlegen/Löschen erfordert **Schreibzugriffsrecht** auf das Verzeichnis
- unabhängig von den Zugriffsrechten auf die referenzierte Datei

Verzeichniseintrag II

- ein Namenverzeichnis ist eine **spezielle Datei** der Namensverwaltung



- das selbst einen Namen hat, der auf einen *inode* verweist
- über eine Verknüpfung erreichbar ist aus einem anderen Verzeichnis
- Namen getrennt von eventuellen Dateiinhalten speichert

*Verknüpfungen anlegen/löschen zu können, ist eine **Berechtigung**, die sich nur auf das Verzeichnis der betreffenden Verknüpfungen bezieht!*

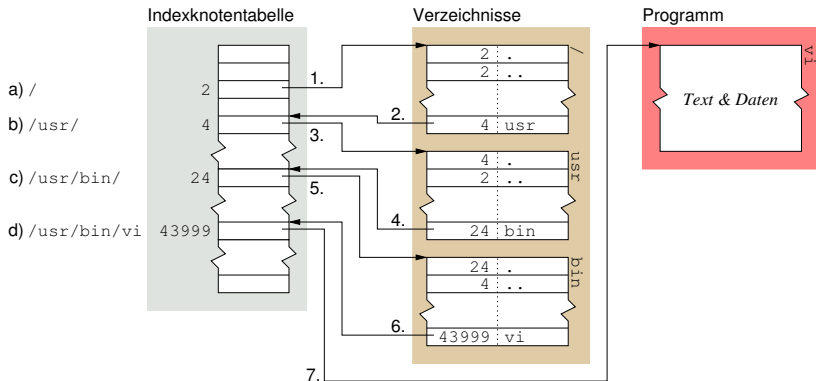
- Selbstreferenz („dot“, 13) und Elternverzeichnis („dot dot“, 11) geben wenigstens zwei Verweise in einem Verzeichnis
 - auch wenn das Verzeichnis selbst sonst keine weiteren Namen enthält

- **Namensbindung** (*name binding*) kommt zuerst, also die Abbildung der symbolischen Adresse in eine numerische Adresse
 - einen Pfadnamen mit einem *inode* assoziieren: `creat(2)`, `link(2)`
 - geschieht zum **Erzeugungszeitpunkt** eines Datei-/Verzeichnisnamens
 - diesen mit einem freien/belegten *inode* verknüpfen und
 - dann in ein Namensverzeichnis eintragen
- **Namensauflösung** (*name resolution*) kommt später, die Umsetzung der symbolischen Adresse in eine numerische Adresse
 - einen *inode* anhand eines Pfadnamens lokalisieren: `open(2)`
 - geschieht zum **Benutzungszeitpunkt** eines Datei-/Verzeichnisnamens
 - Verzeichnisse für jeden einzelnen Namen im Pfad durchsuchen und
 - schließlich den Dateinamen (Blatt) auffinden

Hinweis

Der *inode* besitzt eine Adresse (*a*) in der Ablage und (*b*) im Arbeitsspeicher. Diese ist eine Art **virtuelle Adresse**, über die Inhalte von Dateien speicherabgebildet im virtuellen Adressraum indirekt zugänglich werden. Die Ladestrategie operiert vorausschauend.

Auflösung am Beispiel von /usr/bin/vi



- a) Wurzelverzeichnis des Namensraums öffnen
- b) darin Namenseintrag `usr` suchen, ist ein Verzeichnis, öffnen
- c) darin Namenseintrag `bin` suchen, ist ein Verzeichnis, öffnen
- d) darin Namenseintrag `vi` suchen, ist ein Programm, öffnen

- *inode*-Nummern gelten in einem bestimmten **Namensraum**
 - der Zugriff auf einen anderen Namensraum ist darüber nicht möglich
 - sie teilen sich damit dieselbe Eigenschaft wie logische/virtuelle Adressen
 - sie sind ein **Tabellenindex**, wie der *p*- bzw. *s*-Anteil solcher Adressen
- der hierarchische Namensr. ist ein **gerichteter azyklischer Graph**
 - um den **Wurzelbaum** zu erhalten, scheiden feste Verknüpfungen aus
 - feste Verknüpfungen zu Verzeichnissen zerstören die azyklische Struktur
 - das Elternverzeichnis (..) eines Verzeichnisses wäre dann uneindeutig
 - Gefahr von Endlosschleifen bei der Namensraumabsuche
- Abbildung $f : N_{symbolisch}^d \mapsto N_{symbolisch}^z$ hat diese Merkmale nicht
 - feste Verknüpfungen sind ununterscheidbar, symbolische nicht: **Dateityp**
 - symbolische Verknüpfungen haben einen eigenen *inode*, feste nicht
- Ursprung ist der **symbolische Name** (*symbolic name*) in Multics [1]
 - zur dynamischen Bindung von Namen an (besondere) E/A-Geräte

Einführung

Datei

Außensicht

Innensicht

Namensräume

Dateisystem

Zusammenfassung

- **symbolische Bezeichnungen** abstrahieren von konkreten Adressen
 - Programmtext- und -datenstellen, Datei-/Pfadnamen, ..., URL
 - üblich ist die mehrstufige statische und auch **dynamische** Auflösung
- Datei-/Pfadnamen repräsentieren Adressen im Ablagesystem
 - der Namens-/Adressraum ist (nicht nur hier) hierarchisch organisiert
 - Orientierung am Dateisystem: Datei, Verzeichnis, Indexknoten, Verknüpfung
- das Zusammenspiel statischer und dynamischer Verfahren ist typisch in Rechensystemen
 - nämlich Abbildungsfunktionen, die vor und zur Laufzeit greifen

Zusammenfassung

Bibliographie

[1] FEIERTAG, R. J. ; ORGANICK, E. I.:

The Multics Input/Output System.

In: *Proceedings of the Third ACM Symposium on Operating System Principles (SOSP 1971)*, October 18–20, 1971, Palo Alto, California, USA, ACM, 1971, S. 35–41

Anhang

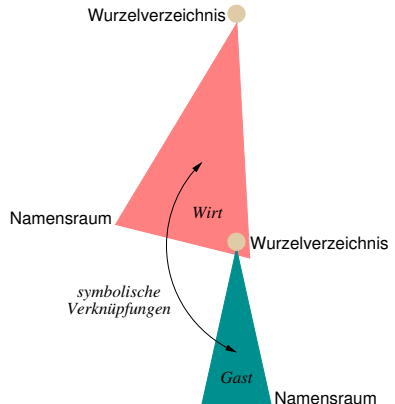
Pfadnamen

- feste Verknüpfung
 - nicht zu Verzeichnissen oder Dateien anderer Dateisysteme
 - überdauert die Umplatzierung einer Datei
 - bleibt bestehen, nur solange es noch Referenzen gibt
 - hat keinen eigenen Indexknoten
- symbolische Verknüpfung
 - auch zu Verzeichnissen und Dateien anderer Dateisysteme
 - ungültig nach Umplatzierung einer Datei
 - bleibt bestehen, auch wenn es keine Referenzen gibt
 - hat einen eigenen Indexknoten

„Nicht alles, was glänzt, ist Gold“ (Shakespeare, 1600)

```
wosch@lorien 1$ mkdir -p Laptop/faiu43w; cd Laptop; ln -s faiu43w lorien; ls -l
total 8
drwxr-xr-x  2 wosch  wosch   68 29 Apr 13:01 faiu43w
lrwxr-xr-x  1 wosch  wosch    7 29 Apr 13:02 lorien -> faiu43w
wosch@lorien 2$ cd lorien
wosch@lorien 3$ cd ..; rmdir faiu43w; cd lorien
-bash: cd: lorien: No such file or directory
wosch@lorien 4$ ls -l
total 8
lrwxr-xr-x  1 wosch  wosch    7 29 Apr 13:02 lorien -> faiu43w
wosch@lorien 5$ mkdir faiu43w; cd lorien
wosch@lorien 6$ ln -s Fata\ Morgana SP1
```

- Namensräume können an einem **Befestigungspunkt** (*mount point*) miteinander verbunden werden
 - ohne damit jedoch eine Erweiterung des Namensraums vorzunehmen
- der Punkt ist ein **Verzeichnis** im Wirtsnamensraum
 - Einhängen (*mount*) blendet den Inhalt des Wirtsverzeichnisses aus
 - der Wurzelverzeichnisinhalt des Gastnamensraums erscheint
 - Aushängen (*unmount*) macht den alten Verzeichnisinhalt sichtbar
- ein **Pfadname** kann dann Wirts- und Gastnamensraum abdecken
 - einerseits streng hierarchietreu, von oben nach unten (*top down*)
 - andererseits quer verweisend, durch **symbolische Verknüpfung**



Anhang

Internetadressen

Definition (Nummerung (DIN 6763))

Bilden, Erteilen, Verwalten und Anwenden von Nummern.

- die **Eindeutigkeit** der Speicher- und Standortadressen ist begrenzt
 - Indexknotennummern durch den **Namensraum** ihres Dateisystems
 - reale, logische und virtuelle Adressen durch ihren (Prozess-) **Adressraum**
 - Prozesskennungen durch den **Nummernraum** ihres Rechensystem
- die **Internetprotokolladresse** (IP-Adresse) ist weltweit eindeutig
 - IPv4** ■ 32 Bit: vier Blöcke zu jeweils drei Dezimalstellen (8 Bit)
 - IPv6** ■ 128 Bit: acht Blöcke zu jeweils vier Hexadezimalstellen (16 Bit)
 - vom ARP (*address resolution protocol*) aufgelöst und umgewandelt in die **Netzwerkadapteradresse** (MAC, *media access control*)
 - rechensystemlokal wird das Adressenpaar in der ARP-Tabelle hinterlegt
- verallgemeinerte Form ist der/die **URL** (*universal resource locator*)
 - neben Adressinformationen ist zusätzlich die **Zugriffsmethode** enthalten, die durch `://` von den schemaspezifischen Angaben getrennt ist
 - die Internetadresse identifiziert dabei den **Wirt** (*host*) einer Webanfrage