

Tutorat 9

Wahlfreier Zugriff, Hardlinks und Softlinks, Terminal Multiplexer

Organisatorisches

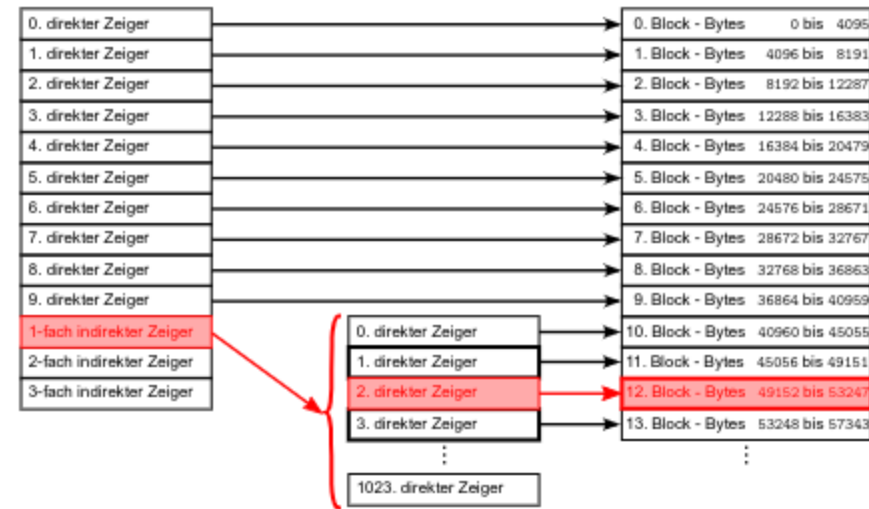
Mindestanforderungen Studienleistung

- denkt an die Anmeldung der **Studienleistung**
- **12** von **15** Übungsblättern sinnvoll bearbeitet
 - **"sinnvoll bearbeitet"**: eine Aufgabe in anständiger Weise bearbeitet
 - **Assistent**: Wenn es **nicht ganz 12** sind, die Person aber ansonsten bemüht war, passt das auch
- Regelmäßige **aktive Teilnahme** an den Tutoraten
 - aber **keine** Anwesenheitspflicht!
 - **Kommentar vom Tutor**: Da gibt es einigen Interpretationsspielraum und Studienleistung wird auch vergeben, wenn die Person immer **schöne Abgaben** hatte (viel **grün** in Blättern) aber nie im Tutorat war. Und vice versa auch, wenn die Person immer gut in den **Tutoraten mitgearbeitet** hat, aber bei den **Abgaben geschwächelt** hat

Übungsblatt

Aufgabe 1 freier Zugriff bei I-Nodes und bei FAT

a) I-NodedColodColor



- **Zeiger 0 bis 9:** $0 \text{ bis } 10 \cdot 4096 - 1 = 40959$

Übungsblatt

Aufgabe 1 - Wahlfreier Zugriff bei I-Nodes und bei FAT

a) I-Node

- **einfach indirekter Block:** $10 \cdot 4096 = 40960$ bis $10 \cdot 4096 + 1024 \cdot 4096 - 1 = 4235263$ (1 indirekter Block mit 1024 Zeigern auf 4KB Blöcke) ✓
- $\left\lfloor \frac{50001 - 40960}{4096} \right\rfloor = 2 \rightarrow$ **Zeiger Nr. 2 im einfach indirekten Block**, also der **Datenblock Nr. 12** (von 1 anfangend zu zählen 13)
 - **Datenblock 12:** $12 \cdot 4096 = 49152$ bis $13 \cdot 4096 - 1 = 53247$ ✓
- **Byte Nr. 50000:** $50001 - 12 \cdot 4096 - 1 = 84$

Übungsblatt

Aufgabe 1 - Wahlfreier Zugriff bei I-Nodes und bei FAT

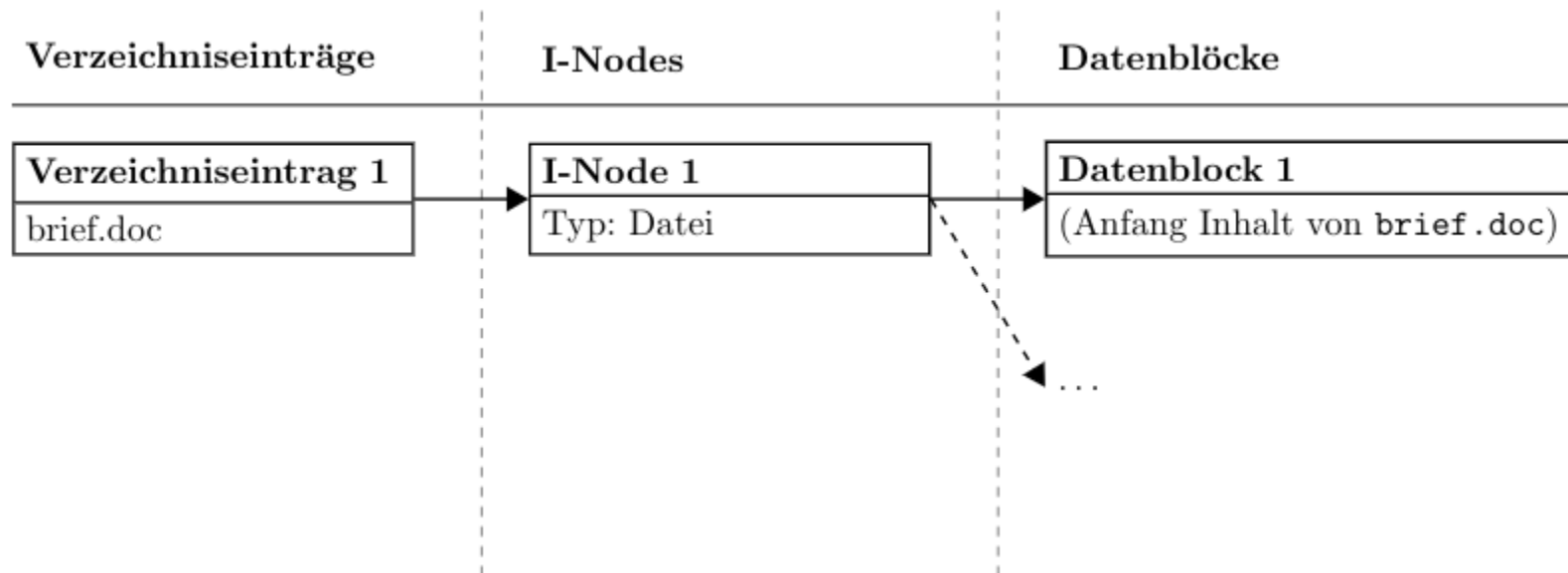
b) FAT32

- einfach verkettete Liste der Datenblöcke der Datei wird **sequentiell** gelesen
- **Anzahl N der verfolgten Verweise:** $N = \left\lceil \frac{n}{b} \right\rceil - 1$
 - danach dem gefundenen Verweis auf den Plattenblock folgen
- Zugriffszeit in Abhängigkeit von n asymptotisch **linear**: $O(n)$

Übungsbaltt

Aufgabe 2 - Hardlinks und symbolische Links

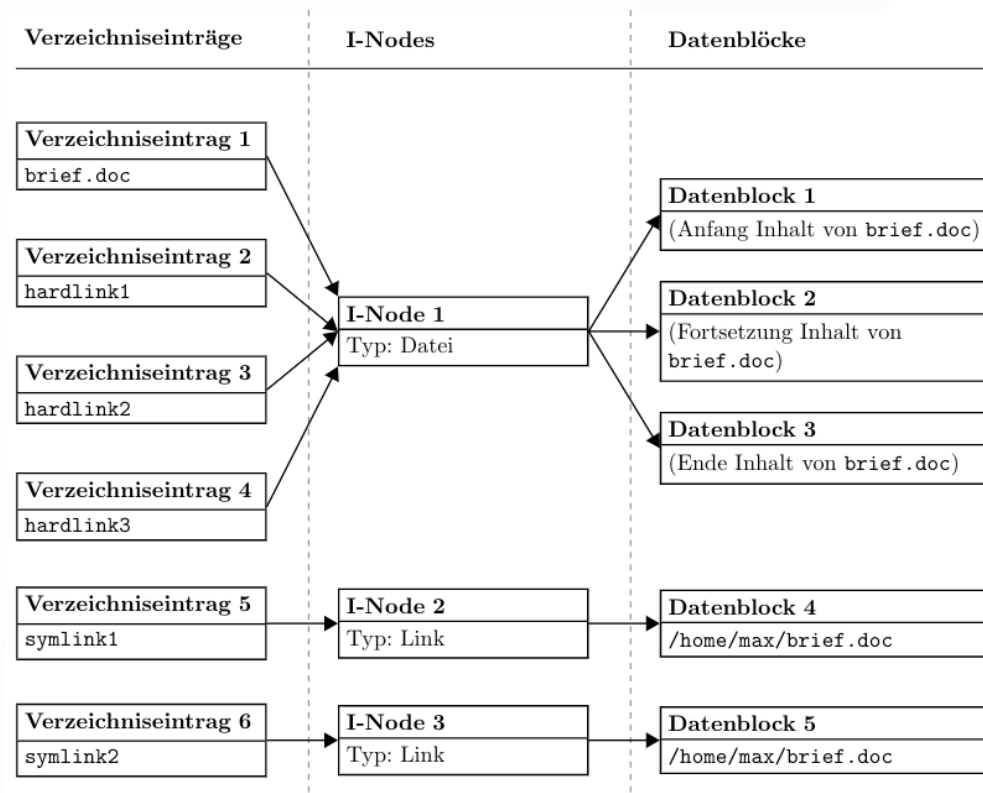
a)



Übungsbaltt

Aufgabe 2 - Hardlinks und symbolische Links

a)



Übungsblatt

Aufgabe 2 - Hardlinks und symbolische Links

a)

- Pfad in Datenblöcken 3/4 kann **relativ** oder **absolut** angegeben werde
- Datenblöcke 3 und 4 auch dürfen auch fehlen und der Pfad stattdessen in den I-Nodes 2 bzw. 3 angegeben sein

Übungsblatt

Aufgabe 2 - Hardlinks und symbolische Links

b)

- **Datei selbst:** 1 I-Node, 3 Datenblöcke, 1 Verzeichniseintrag (\Rightarrow 3 KB)
- **Symlinks:** jeweils 1 I-Node, 1 Datenblock für die Linkdatei und 1 Verzeichniseintrag ($\Rightarrow 2 \cdot 1$ KB)
- **Hardlinks:** jeweils 1 Verzeichniseintrag, 0 I-Nodes und 0 Datenblöcke
- **Insgesamt:** 5 Datenblöcke = 5 KB Speicherplatz, 3 I-Nodes, 6 Verzeichniseinträge

“ Die Verzeichniseinträge benötigen zusätzlich Speicherplatz in den Verzeichnistabellen der Elternverzeichnisse, in denen die Dateien liegen. Dieser Speicherplatz und der Speicherplatz für die I-Nodes selbst wird hier vernachlässigt. ”

Übungsblatt

Aufgabe 2 - Hardlinks und symbolische Links

c)

- die Zugriffsrechte aller **Hardlinks** ändern sich mit, da die Rechte im I-Node gespeichert werden und die Datei und alle Hardlinks auf das selbe I-Node zeigen
- die Zugriffsrechte aller **symbolischen Links** bleiben bei `1rwxrwxrwx`, da sie eigene I-Nodes haben

Übungsblatt

Aufgabe 2 - Hardlinks und symbolische Links

d)

- Die **Zugriffsrechte** des **symbolischen Links** können **nicht** geändert werden.
- Stattdessen wird der chmod-Befehl auf das **Linkziel** angewandt, insofern die entsprechenden Rechte bestehen
- Die angezeigten **Rechte der symbolischen Links** haben **keine Bedeutung** für den eigentlichen Zugriff auf die Zielfeile.
- Wäre das nicht der Fall, so könnte man einen **symbolischen Link** auf eine **fremde Datei** anlegen
 - Da man **selbst der Besitzer** des Links ist, konnte man die Rechte des Links nach Belieben ändern und sich somit **Zugriffsrechte verschaffen**.
 - Aus diesem Grund werden **symbolische Links** bei Benutzung immer **erst dereferenziert** und dann werden die **Rechte des Linkziels** ausgewertet um die **Zugriffsrechte zu bestimmen**

Übungsblatt

Aufgabe 3 - Prozesse und Signale

a)

- Mehrere Möglichkeiten, z.B.
 - `ps aux` (BSD syntax)
 - `ps -ef` (standard syntax)
 - `e`: Select all processes
 - `f`: Do full-format listing

Übungsblatt

Aufgabe 3 - Prozesse und Signale

b)

- Es existieren eine Reihe von möglichen Befehlen, z.B.
 - `kill -s <signal value> <pid>`
 - `kill -<signal value> <pid>`
 - `kill -<option1> <pid>`
 - `kill -<option2> <pid>`

	signal value	option1	option2
SIGSTOP	19 (für x86-Architektur)	SIGSTOP	STOP
SIGCONT	18 (für x86-Architektur)	SIGCONT	CONT
SIGTERM	15	SIGTERM	TERM
SIGKILL	9	SIGKILL	KILL

Übungsblatt

Aufgabe 3 - Prozesse und Signale

b)

- `$(pidof -x counter.sh)` anstelle von `<signal value>`
 - `-x`: Scripts too - this causes the program to also return process id's of shells running the named scripts

```
> $ pidof counter.sh  
  
> $ pidof -x counter.sh  
131047
```

- Um die Ermittlung der **PID** zu umgehen, kann statt `kill [...] <pid>` der Befehl `killall [...] counter.sh` mit der selben Syntax verwendet werden
 - If no signal name is specified, `SIGTERM` is sent

Übungsblatt

Aufgabe 3 - Prozesse und Signale

c)

- `man 7 signal`

Signal	Bedeutung
SIGSTOP	Blockiert den Prozess.
SIGCONT	Setzt einen gestoppten Prozess fort [als Hintergrundprozess].
SIGTERM	Fordert den Prozess auf sich zu beenden, kann ignoriert werden.
SIGKILL	Erzwingt das Beenden des Prozesses.

- **Unterschied SIGTERM und SIGKILL** : SIGTERM kann vom Prozess **ignoriert** werden oder der Prozess kann noch seine **Ressourcen aufräumen** und dann terminieren. Das Verhalten wird vom **Programmierer vorgegeben**. SIGKILL kann vom jeweiligen Prozess **nicht ignoriert** werden. Der Prozess wird mit SIGKILL beendet, **ohne** dass er auf das Signal **reagieren** kann.

Übungsblatt

Aufgabe 3 - Prozesse und Signale

d)

- 1) Screen starten: „`screen`“ oder „`screen -S <sessionname>`“
- 2) Counter starten: „`./counter.sh`“
- 3) Detach: Ctrl-a d
- 4) Ausloggen und wieder einloggen
- 5) Session suchen: „`screen -ls`“
- 6) Re-attach: „`screen -r <sessionname>`“

Alternativen:

- screen starten und sofort detachen: „`screen -d -m ./counter.sh`“
- Wenn nur eine Session läuft, kann „`screen -r`“ ohne Session-Name aufgerufen werden. Dieser Befehl re-attached die letzte Session.

Ergänzungen

Herausfinden, wer den Prozess aufgerufen hat

- `ps auxf`
 - `a`: list all processes with a terminal (tty), or to list all processes when used together with the `x` option
 - `x`: list all processes owned by you (same EUID as ps), or to list all processes when used together with the `a` option
 - `u`: Display user-oriented format
 - `f`: ASCII art process hierarchy (forest)
- `pstree`

Ergänzungen

Weitere interessante command-line Argumente

- **ps** :
 - **T** : Select all processes associated with this terminal. Identical to the **t** option without any argument
 - **r** : Restrict the selection to only running processes
 - **c** : Show the true command name. Command arguments and any modifications to them are thus not shown
 - **H** : Show threads as if they were processes

Ergänzungen

Sort keys

- `ps -eo pid,ppid,user,priority,size,vsize,pcpu,cmd,start_time`

KEY	LONG	DESCRIPTION
c	cmd	simple name of executable
C	pcpu	cpu utilization
f	flags	flags as in long format F field
g	pgrp	process group ID
G	tpgid	controlling tty process group ID
j	cutime	cumulative user time
J	cstime	cumulative system time
k	utime	user time
m	minflt	number of minor page faults
M	majflt	number of major page faults
n	cminflt	cumulative minor page faults
N	cmajflt	cumulative major page faults
o	session	session ID
p	pid	process ID
P	ppid	parent process ID
r	rss	resident set size
R	resident	resident pages
s	size	memory size in kilobytes
S	share	amount of shared pages
t	tty	the device number of the controlling tty
T	start_time	time process was started
U	uid	user ID number
u	user	user name
v	vsize	total VM size in KiB
y	priority	kernel scheduling priority

Ergänzungen

Tmux

- **Überblick:**
 - **Gute config file:** <https://github.com/gpakosz/.tmux>
 - **Tutorials:** <https://github.com/rothgar/awesome-tmux>
 - **Cheatsheet:** <https://gist.github.com/MohamedAlaa/2961058>

Schnelle Lösung:

```
export TERM=xterm  
echo $TERM
```

Ergänzungen

Tmux

Langfristige Lösung:

- **Informationen über das verwendete Terminal rüberschicken:**

```
infocmp > alacritty.terminfo
# Pool-Account
scp alacritty.terminfo XY123@login.uni-freiburg.de:~/
# oder Uni-Account
# scp alacritty.terminfo <pool-login>@login.informatik.uni-freiburg.de:~/
```

- **Informationen über das Terminal in Datenbank abspeichern:**

```
tic -x alacritty.terminfo
```


Ergänzungen

Tmux

- **oneliner:**

- `infocmp | ssh "XY123.uni-freiburg.de" 'tic -x /dev/stdin'`
- `infocmp | ssh "<pool-login>.informatik.uni-freiburg.de" 'tic -x /dev/stdin'`

Wichtige Commands

- `ctrl+a,?`
- `ctrl+a,e` zum öffnen von `~/.tmux.conf.local`
- `tmux new -As default`
- `tmux ls`
- `tmux kill-session -t default`
- `tmux kill-server` oder `killall "tmux: server"` oder `ctrl+a,:kill-server`

Ergänzungen

Tmux

Nützliche Einstellungen

- `tmux source-file ~/.tmux.conf`
 - `set-option -g status-position top`
 - `bind-key < swap-window -t -1`, executed with `ctrl+a,<`
 - `bind-key > swap-window -t +1`, executed with `ctrl+a,>`
 - `default theme` kommentieren und `default theme (ansi)` auskommentieren
 - `tmux_conf_copy_to_os_clipboard=true` auskommentieren
 - `set -g status-keys vi` und `set -g mode-keys vi` auskommentieren

Ergänzungen

Tmux

Plugin für Wiederherstellung über Neustart hinweg

- <https://github.com/tmux-plugins/tmux-resurrect>
- `set -g @plugin 'tmux-plugins/tmux-resurrect'` auskommentieren

```
# Prozesse wiederherstellen:  
set -g @resurrect-processes '~mocp ~htop ~nvim ~emacs ~neomutt ~vifm ~man ~less'  
# alle Prozesse: set -g @resurrect-processes ':all:'  
# Zeichen in Panes speichern:  
set -g @resurrect-capture-pane-contents 'on'  
# Session.vim berücksichtigen:  
set -g @resurrect-strategy-nvim 'session'
```

Ergänzungen

Tmux

Plugin für Wiederherstellung über Neustart hinweg

- `->` to specify a command to be used when restoring a program
- `~` to restore a program whose process contains target name
- gesichert mit Datum unter: `~/.tmux/resurrect/`, `ln -sf <file_name> last`

Ergänzungen

Tmux

Shh mit Tmux

```
Host uni-tmux
  HostName login.uni-freiburg.de
  # oder HostName login.informatik.uni-freiburg.de
  User XY123 # oder <pool-login>
  RequestTTY yes
  RemoteCommand tmux new -As default
```

- `ssh uni-tmux`

Ergänzungen

Prozess suspenden

- `ctrl+z`, `bg`, `fg`, `jobs`
 - `bg/fg %<job-number>`
 - `bg/fg %<string-beginning>`
 - z.B. `fg ./co`
 - `bg/fg %?<some_string-somewhere>`

Ergänzungen

Addition binär und dezimal

011011 (27)	17718
+ 011101 (29)	+ 6524
11111	11 1
=====	=====
111000 (56)	24242

00 + 00 = 00	00 + 00 (+ 01) = 01
00 + 01 = 01	00 + 01 (+ 01) = 10
01 + 00 = 01	01 + 00 (+ 01) = 10
01 + 01 = 10	01 + 01 (+ 01) = 11

Ergänzungen

Subtraktion binär und dezimal (nicht empfohlen,
dient Vergleich mit nächster Folie)

(1)	
0111000 (56)	24242
- 0011011 (27)	- 17718
11111	11 1
=====	=====
0011101 (29)	6524

10 - 00 = 10	10 - 00 (- 01) = 01
10 - 01 = 01	10 - 01 (- 01) = 00
11 - 00 = 11	11 - 00 (- 01) = 10
11 - 01 = 10	11 - 01 (- 01) = 01

Ergänzungen

Subtraktion binär und dezimal (funktioniert immer, egal was für Vorzeichen Zahlen haben)

```
(2)
  0111000 (56)
+ 1100101 (27) (0011011 negiert und +1)
  11
  =====
  0011101 (29)
```

- **Zweierkomplement Negation:** 11011 -> 011011 -> 100100 -> 100101
 - 0 en hinzufügen bis **Minuend** und **Subtrahend** beide gleiche Länge haben und Platz für ihr **Vorzeichenbit** ist und dieses korrekt gesetzt ist
 - **1er Komplement Negation** und +1 nicht vergessen für den **Subtrahenden**

Ergänzungen

Multiplikation binär und dezimal

1101 x 1001 (13 * 9)

1101

0000

0000

1101

=====

1110101 (117)

1304 x 12

48

+ 0

+ 36

+12

=====

15648

- **Verschiebung** ist aufgrund der 0en, die hier ausgelassen sind

Ergänzungen

Division binär

```
1110101 / 1011 (117 : 11) = 1010 (10) Rest: 111 (7)
- 1011 |||
===== |||
   111 ||
-      0 ||
   ===== ||
   1110 |
-  1011 |
   ===== |
     111
-      0
   =====
     111
```

Ergänzungen

Division dezimal

15658 / 12 = 1304,833...

12 |||

== |||

36 ||

36 ||

== ||

05 |

0 |

== |

58

48

...

Ergänzungen

Division dezimal

```
==  
10|0   oder Rest: 10  
 9 6  
===  
 40  
 36  
  ==  
  40  
  36  
   ==  
   4...
```

Ergänzungen

Division binär

- bei **binärer Division** gibt es nur **2 Zustände** (**1** oder **0**), dementsprechend wird entweder die Zahl so übernommen ($\text{Zahl} \cdot 1$) oder die Zahl ist **0** ($\text{Zahl} \cdot 0$)

Division allgemein

- nach jeder Addition ein Zahl runterholen, bis keine mehr runtergeholt werden kann \rightarrow dann Ende (bei **ganzzahliger Division**). Was unten stehen bleibt ist der **Rest**
- bei Division mit Nachkommastellen, 0en runterbringen, bis einmal **kein Rest** mehr rauskommt oder Grenze setzen bis zu der man weiter macht \rightarrow dann Ende
- ist der **Dividend** trotz runtergebrachter weiter Stelle (weil einmal kein Rest übrig blieb) immernoch kleiner als der **Divisor**, so ist der **Quotient** 0, weil nur durch $\cdot 0$ rechnen kann der **Divisor** noch kleiner sein als der **Dividend**

Quellen

Wissenquellen

- https://wiki.archlinux.org/title/Alacritty#Terminal_functionality_unavailable_in_remote_shells
- <https://stackoverflow.com/questions/21523815/tmux-open-terminal-failed-missing-or-unsuitable-terminal-xterm-256color>
- <https://github.com/alacritty/alacritty/issues/2487>
- <https://linux.die.net/man/1/tic>

Quellen

Bildquellen

- 

**Vielen Dank für
eure
Aufmerksamkeit!**

