Tutorübung Grundlagen: Betriebssysteme und Systemsoftware

Moritz Beckel

München, 27. Januar 2023

Freitag 10:15-12:00 Uhr Raum (00.11.038)

Zulip-Stream https://zulip.in.tum.de/#narrow/stream/1295-GBS-Fr-1000-A

Unterrichtsmaterialien findest du hier:

https://home.in.tum.de/~beckel/gbs

Folien wurden von mir selbst erstellt. Es besteht keine Garantie auf Korrektheit.

Dateiverwaltung

Contiguous Allocation

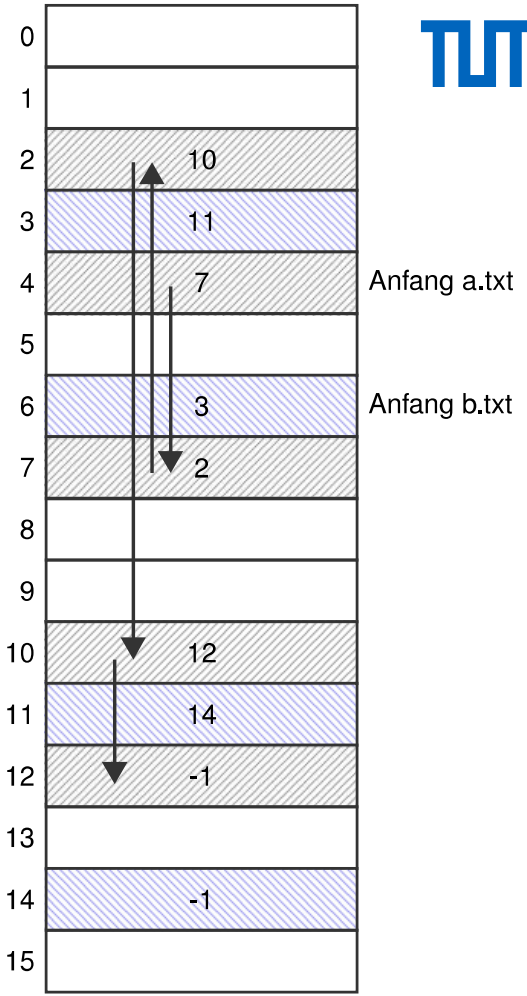
1. Dateien liegen hintereinander auf benachbarten Speicherblöcken
2. große externe Fragmentierung

A	A	A	B	B	B	B	B	C	C
C			E	E	E	F	F	F	F

Dateiverwaltung

Linked List Allocation (FAT)

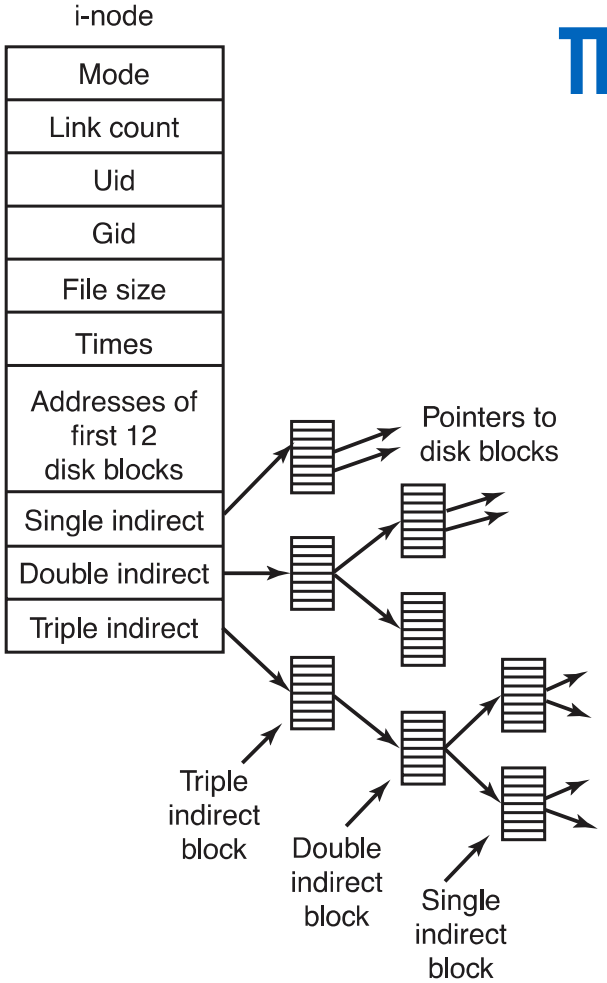
1. Speicherblöcke einer Datei werden innerhalb einer verketteten Liste verwaltet
2. weniger performant, da bei Random-Access von Block zu Block gesprungen werden muss um nächsten Block zu ermitteln
3. Keine externe Fragmentierung
4. Implementiert zum Beispiel durch File Allocation Table (FAT)



Dateiverwaltung

i-Nodes

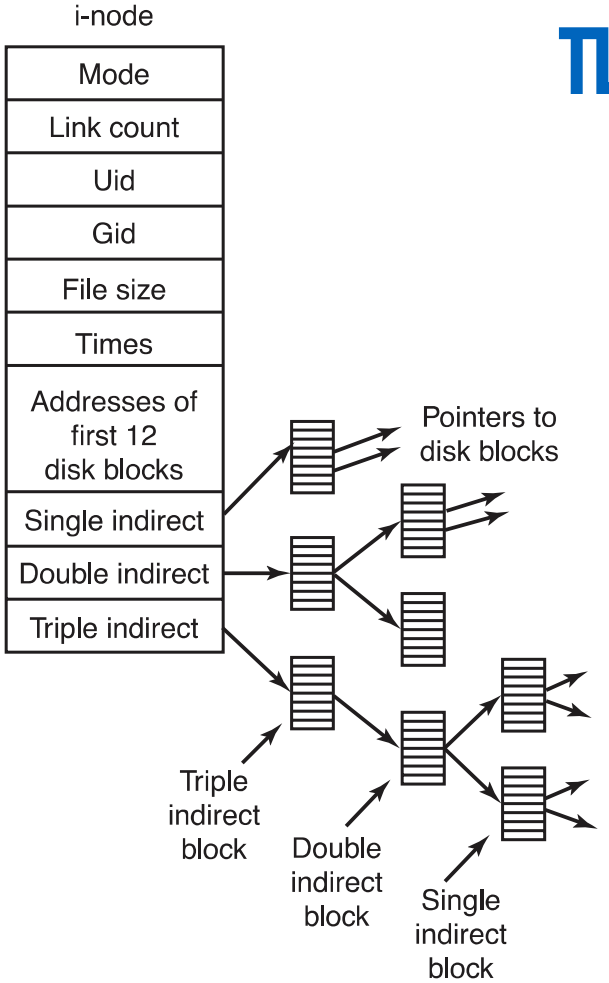
1. Eigene Datenstruktur für Dateien, mit Dateieigenschaften und Verweisen auf Datenblöcke
2. Direkte, Indirekte, doppelt indirekte und dreifach indirekte Verweise möglich auf Datenblöcke
3. Nur i-nodes von aktuell verwendeten Dateien im RAM gehalten



Aufgabe 2

Der Hintergrundspeicher ist in Blöcke aufgeteilt. Nehmen Sie im Folgenden an, ein Block habe eine Größe von 1 KiB und die Adresse eines Blocks benötige 32 Bit.

Welche Größe (in Byte) kann eine Datei maximal haben, wenn im i-node der Datei neben den single-, double- und triple-indirect-Verweisen noch 12 Datenblöcke direkt referenziert werden?



Der Hintergrundspeicher ist in Blöcke aufgeteilt. Nehmen Sie im Folgenden an, ein Block habe eine Größe von 1 KiB und die Adresse eines Blocks benötige 32 Bit. Welche Größe (in Byte) kann eine Datei maximal haben, wenn im i-node der Datei neben den single-, double- und triple-indirect-Verweisen noch 12 Datenblöcke direkt referenziert werden?

Aufgabe 2

Der Hintergrundspeicher ist in Blöcke aufgeteilt. Nehmen Sie im Folgenden an, ein Block habe eine Größe von 1 KiB und die Adresse eines Blocks benötige 32 Bit. Welche Größe (in Byte) kann eine Datei maximal haben, wenn im i-node der Datei neben den single-, double- und triple-indirect-Verweisen noch 12 Datenblöcke direkt referenziert werden?

- ⌊1024B / 4B⌋ = 256 Adressen auf einem Block
- Direct 12 Blöcke
- Single indirect 256 Blöcke
- Double indirect $256 \cdot 256 =$ 65 536 Blöcke
- Triple indirect $256 \cdot 256 \cdot 256 =$ 16 777 216 Blöcke
- Gesamt 16 843 020 Blöcke
- Maximale Größe $16843020 \cdot 1\ \text{KiB} \approx 16448\ \text{MiB} \approx 16{,}06\ \text{GiB}$

Aufgabe 3

Auf einer 16 GiB-Festplatte werden Dateien unter UNIX mittels i-nodes gespeichert. Eine i-node verfüge im Folgenden über 12 direkte Referenzen auf Datenblöcke, einen einfach indirekten Verweis und einen zweifach indirekten Verweis. Die Blockgröße betrage 1024 Bytes und die Adresslänge 32 Bit.

a) Wie viele Blöcke Verwaltungsaufwand benötigt eine 250 KiB-Datei?

Aufgabe 3

Auf einer 16 GiB-Festplatte werden Dateien unter UNIX mittels i-nodes gespeichert. Eine i-node verfüge im Folgenden über 12 direkte Referenzen auf Datenblöcke, einen einfach indirekten Verweis und einen zweifach indirekten Verweis. Die Blockgröße betrage 1024 Bytes und die Adresslänge 32 Bit.

a) Wie viele Blöcke Verwaltungsaufwand benötigt eine 250 KiB-Datei?

- 250 KiB Nutzdaten benötigen 250 Datenblöcke
- 12 Blöcke direkt über die i-node adressiert (1 Block)
- restlichen 238 Blöcke mittels single-indirect adressiert (1 Block)
- 2 Blöcke Verwaltungsaufwand und 250 Blöcke Nutzdaten

Aufgabe 3

Auf einer 16 GiB-Festplatte werden Dateien unter UNIX mittels i-nodes gespeichert. Eine i-node verfüge im Folgenden über 12 direkte Referenzen auf Datenblöcke, einen einfach indirekten Verweis und einen zweifach indirekten Verweis. Die Blockgröße betrage 1024 Bytes und die Adresslänge 32 Bit.

b) Welche Größe würde eine FAT (File Allocation Table) für die obige Festplatte haben? Verwenden Sie die kleinstmögliche Anzahl von Bits für die Blocknummern.

Aufgabe 3

Auf einer 16 GiB-Festplatte werden Dateien unter UNIX mittels i-nodes gespeichert. Eine i-node verfüge im Folgenden über 12 direkte Referenzen auf Datenblöcke, einen einfach indirekten Verweis und einen zweifach indirekten Verweis. Die Blockgröße betrage 1024 Bytes und die Adresslänge 32 Bit.

b) Welche Größe würde eine FAT (File Allocation Table) für die obige Festplatte haben? Verwenden Sie die kleinstmögliche Anzahl von Bits für die Blocknummern.

- 2^{24} Blöcke der Größe 1 KiB auf der Festplatte
- 2^{24} Einträge der Größe 24 Bit in der FAT
- $2^{24} \cdot 24$ bit = $2^{24} \cdot 3$ Bytes = 48 MiB

Aufgabe 3

Auf einer 16 GiB-Festplatte werden Dateien unter UNIX mittels i-nodes gespeichert. Eine i-node verfüge im Folgenden über 12 direkte Referenzen auf Datenblöcke, einen einfach indirekten Verweis und einen zweifach indirekten Verweis. Die Blockgröße betrage 1024 Bytes und die Adresslänge 32 Bit.

c) Welches der beiden Verfahren (i-nodes, FAT) ist bei der Verwaltung von wenigen kleinen Dateien im Vorteil? Warum? Begründen Sie Ihre Antwort (denken Sie an den Hauptspeicherbedarf).

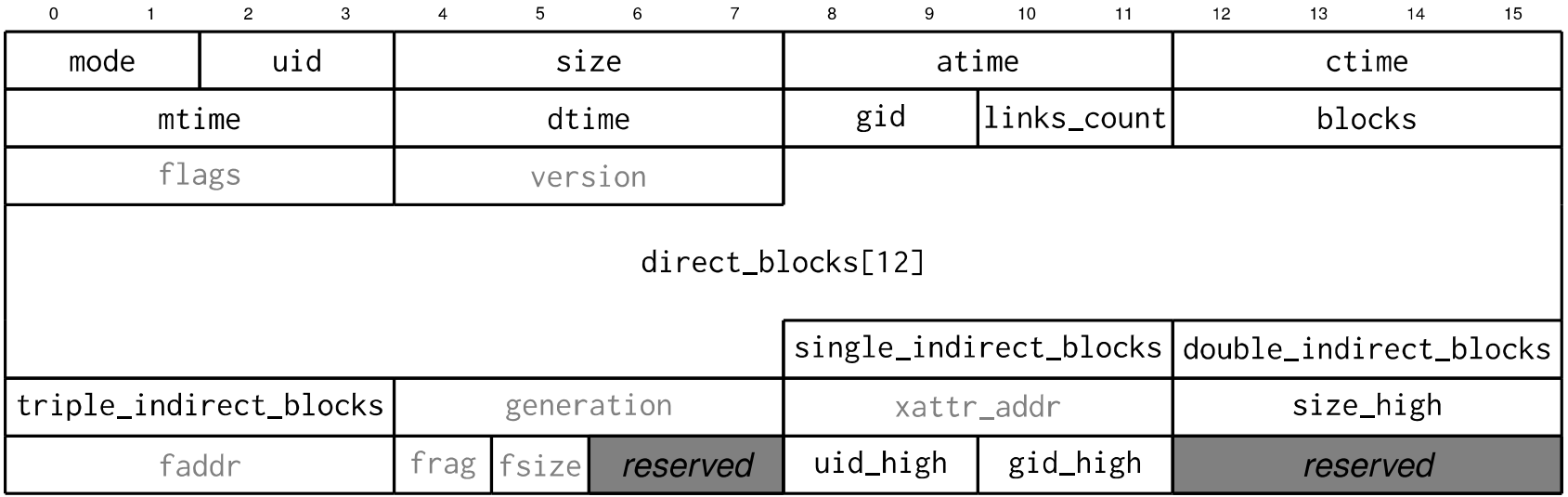
Aufgabe 3

Auf einer 16 GiB-Festplatte werden Dateien unter UNIX mittels i-nodes gespeichert. Eine i-node verfüge im Folgenden über 12 direkte Referenzen auf Datenblöcke, einen einfach indirekten Verweis und einen zweifach indirekten Verweis. Die Blockgröße betrage 1024 Bytes und die Adresslänge 32 Bit.

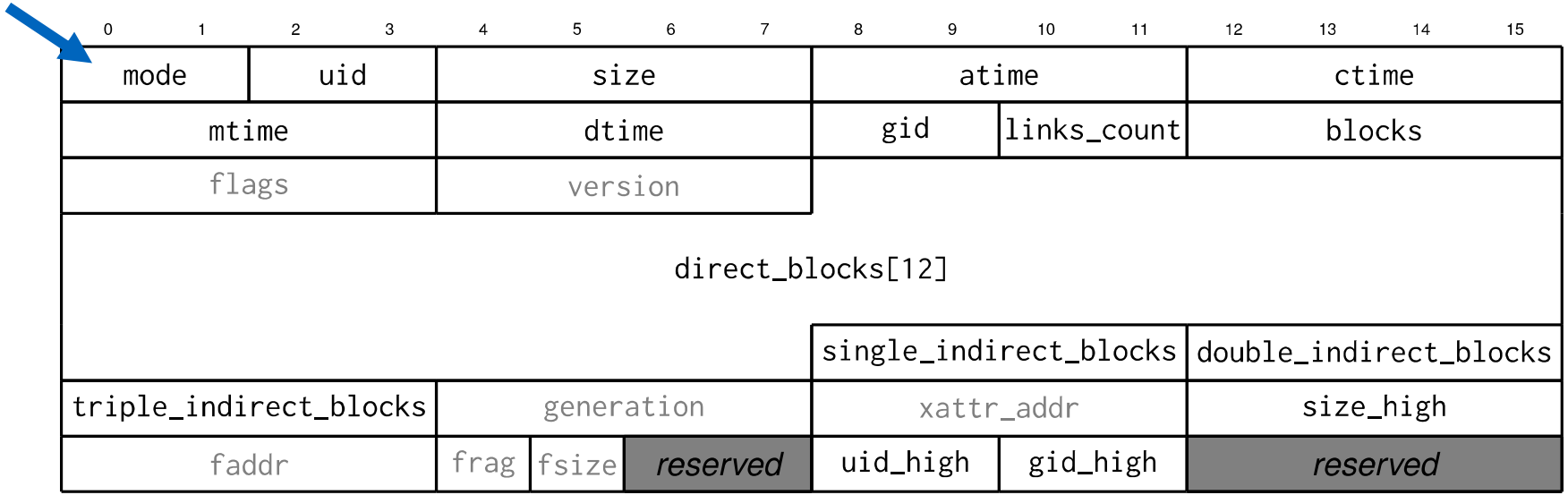
c) Welches der beiden Verfahren (i-nodes, FAT) ist bei der Verwaltung von wenigen kleinen Dateien im Vorteil? Warum? Begründen Sie Ihre Antwort (denken Sie an den Hauptspeicherbedarf).

- Es müssen nur die i-nodes der aktuell geöffneten Dateien im Speicher gehalten werden
- Für Dateien mit weniger als 12 KiB muss nur die entsprechende i-node geladen werden
- Bei FAT immer die komplette File Allocation Tabelle

Aufgabe 4

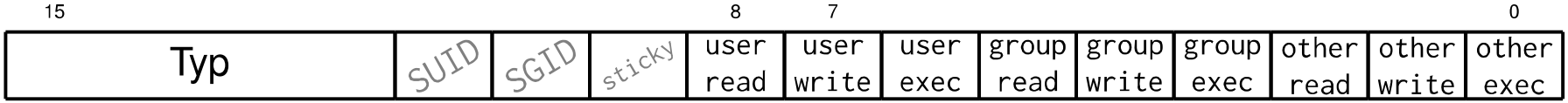

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
mode
uid
size
atime
ctime
mtime
dtime
gid
links_count
blocks
flags
version
direct_blocks[12]
single_indirect_blocks
double_indirect_blocks
triple_indirect_blocks
generation
xattr_addr
size_high
faddr
frag
fsize
reserved
uid_high
gid_high
reserved

Aufgabe 4



mode: Art der Datei (reguläre Datei: 8, Directory: 4, symbolic Link: 10, named Pipe: 1 etc.) + Rechte: r (read), w (write) und x (execute) je für den Besitzer, die Nutzer der Gruppe und alle anderen

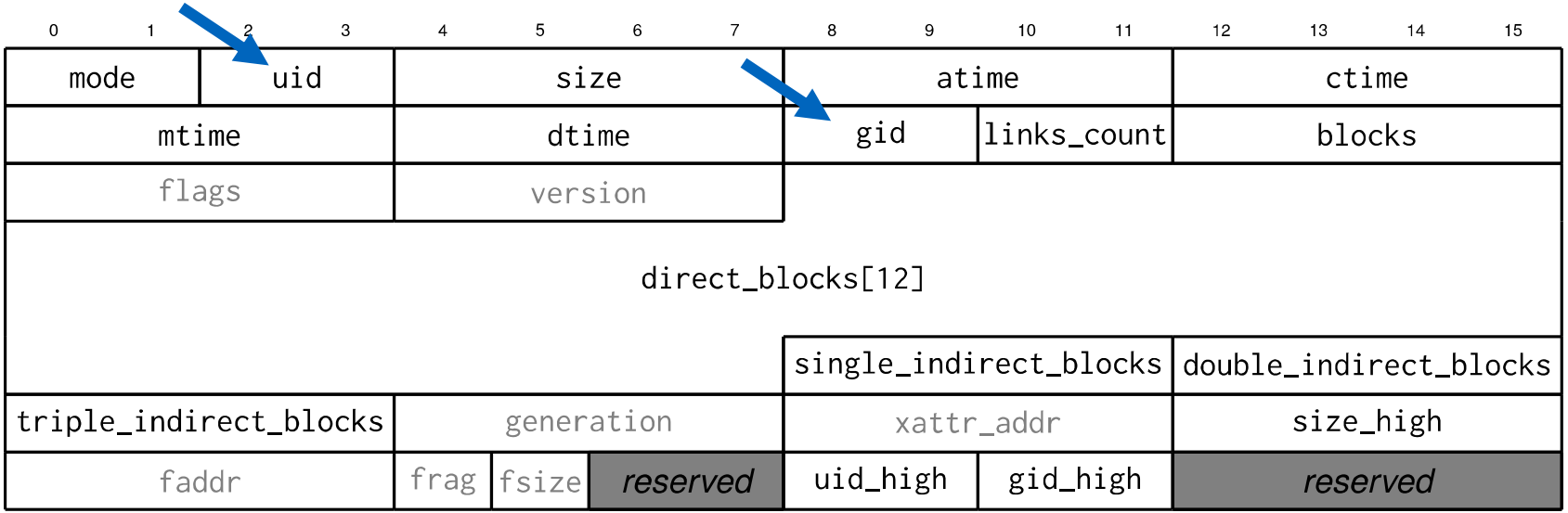
Aufgabe 4



mode: Art der Datei (reguläre Datei: 8, Directory: 4, symbolic Link: 10, named Pipe: 1 etc.) + Rechte: r (read), w (write) und x (execute) je für den Besitzer, die Nutzer der Gruppe und alle anderen

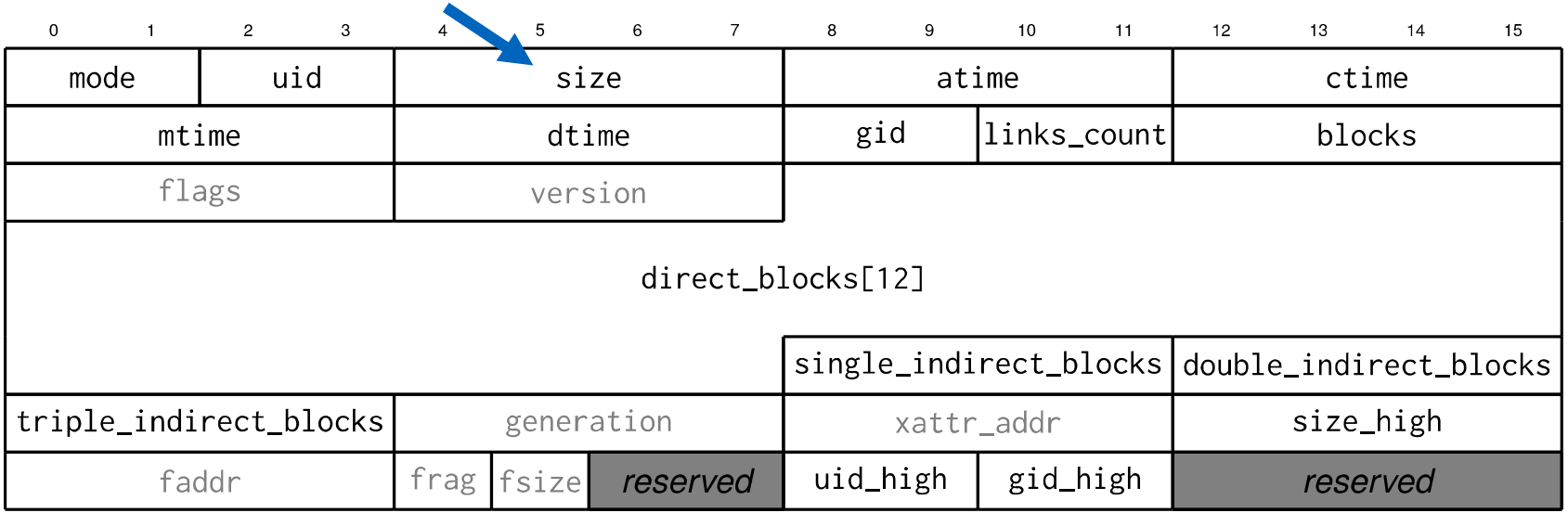
Bei Verzeichnissen haben die Rechte-Bits eine andere Bedeutung: r: Auflisten der Dateien; w: Hinzufügen/Umbenennen/Entfernen von Dateien; x: Nutzen der Dateien im Verzeichnis

Aufgabe 4



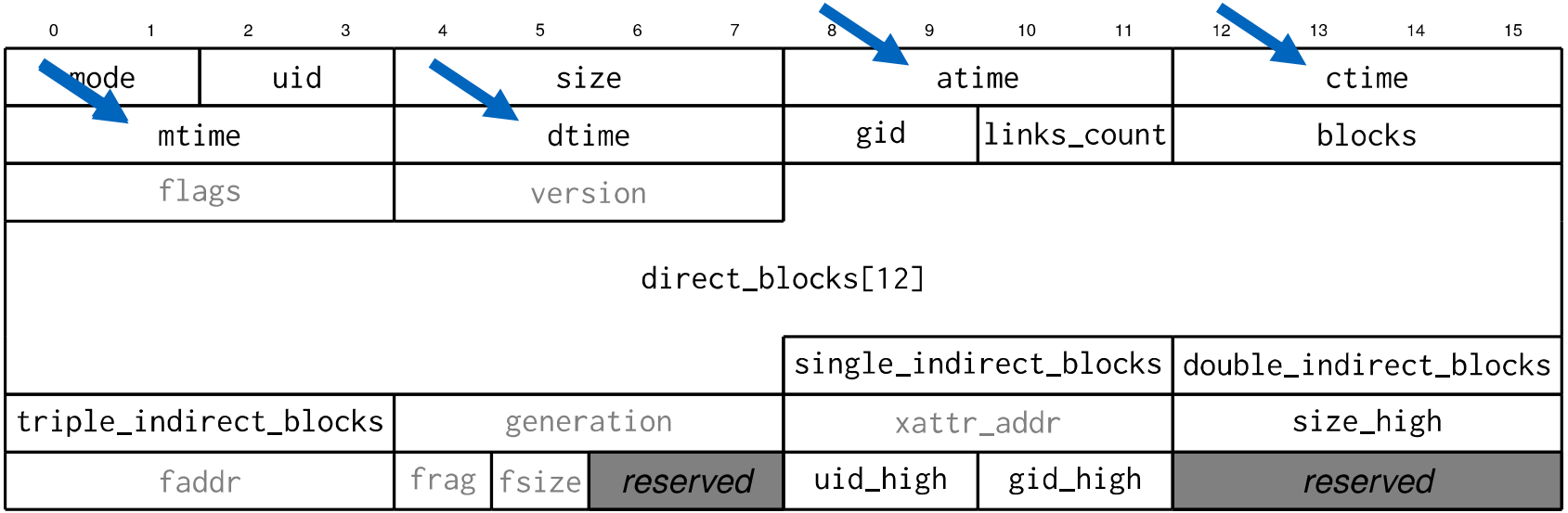
uid, gid: Nutzer-ID des Besitzers, Gruppen-ID

Aufgabe 4



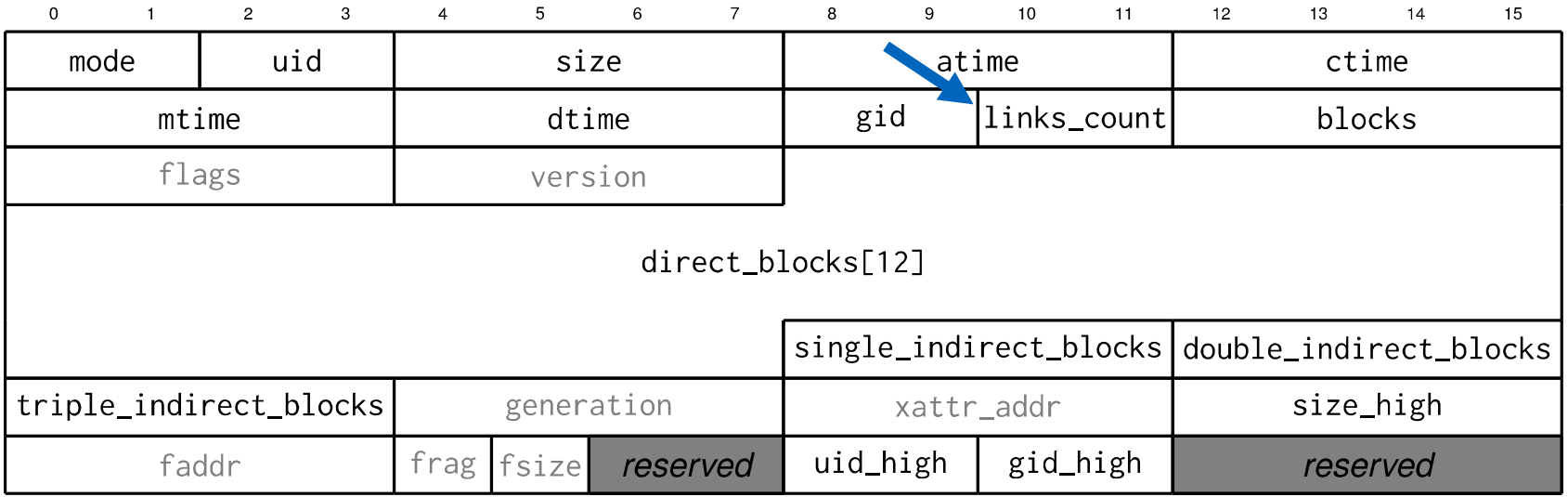
size: Größe der Datei in Bytes

Aufgabe 4



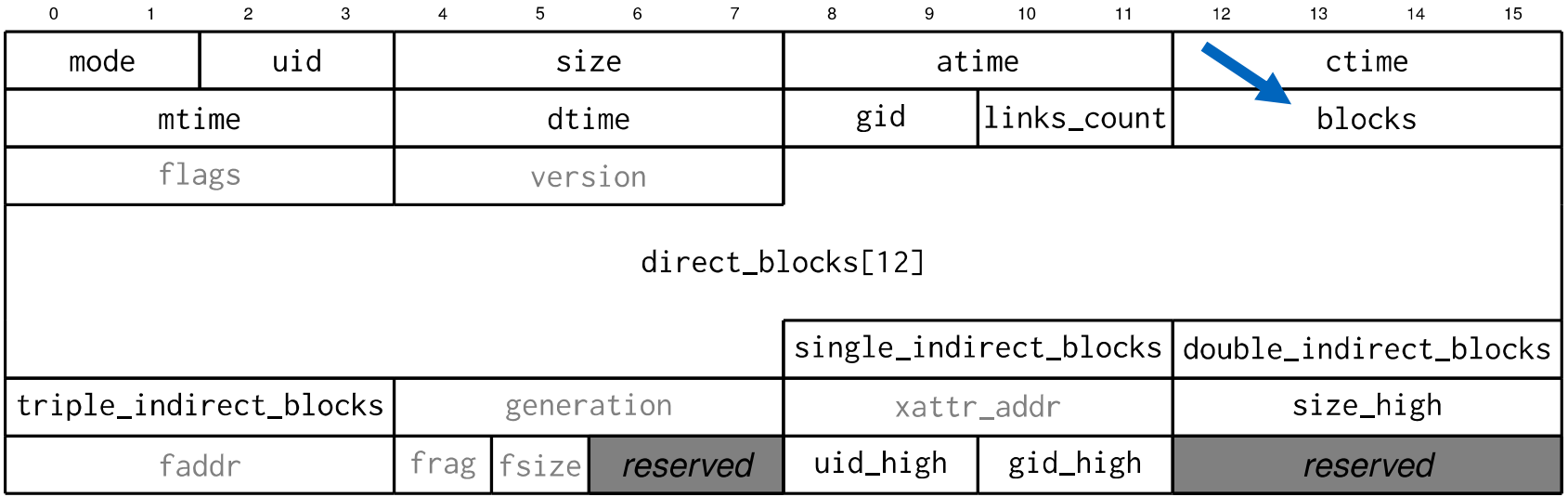
atime, ctime, mtime, dtime: Access-, Change-, Modification- und Deletion-Time. mtime gibt an, wann zuletzt Daten geändert wurden; ctime gibt an, wann zuletzt Inode geändert wurde (außer atime)

Aufgabe 4



links_count: Anzahl der Hardlinks auf Inode

Aufgabe 4



blocks: Anzahl der allozierten 512-Byte Blöcke

(Dieser Wert kann tatsächlich kleiner sein als (Dateigröße / 512 Byte). Fehlende Blöcke werden implizit als genullt gelesen. Solche Dateien heißen sparse.)

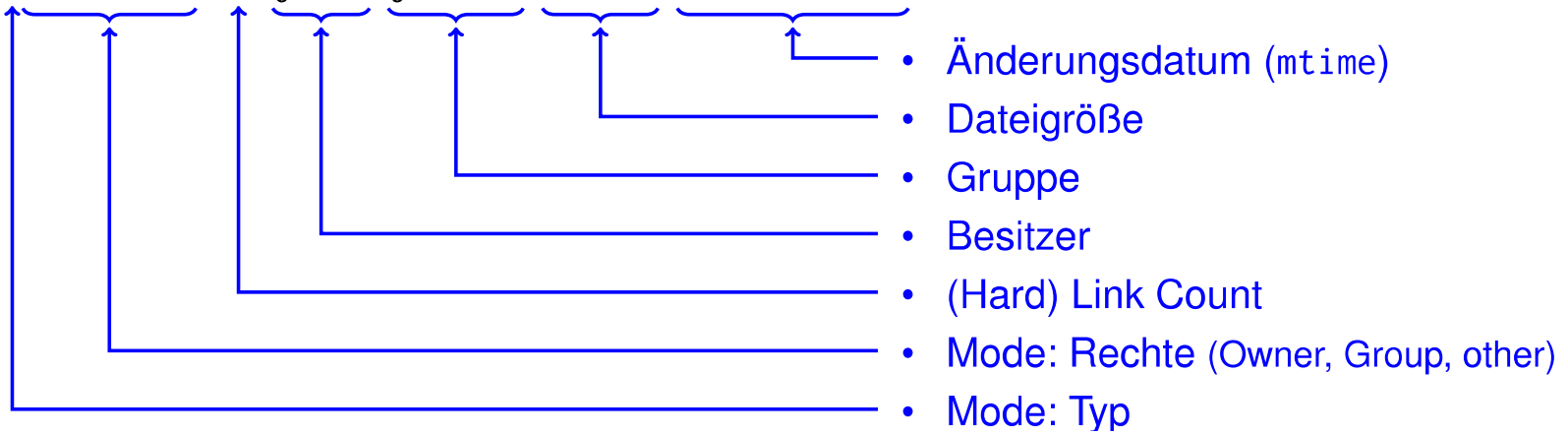
Aufgabe 4

a) Gegeben sei folgender Output von ls -la. Zu welchen Feldern der Inode korrespondieren die Spalten jeweils?

```
total 150K
drwxr-xr-x  3 josef josef     1024 Dec 31 16:36 .
drwxr-xr-x 27 root  root      1024 Dec 31 16:38 ..
-rw-r--r--  2 josef josef       13 Dec 31 15:54 file.txt
-rw-rw----  1 josef gbs_uel 145013 Dec 31 15:57 gbs-endterm-solution-ws2022.pdf
-rwx------  1 josef josef    14866 Dec 31 16:37 generate_valid_itsec_flags.sh
drwxr-xr-x  5 josef josef     3072 Dec 31 16:39 Memes
```

Aufgabe 4

```
total 150K
drwxr-xr-x  3 josef josef     1024 Dec 31 16:36 .
drwxr-xr-x 27 root  root      1024 Dec 31 16:38 ..
-rw-r--r--  2 josef josef       13 Dec 31 15:54 file.txt
-rw-rw----  1 josef gbs_uel 145013 Dec 31 15:57 gbs-endterm-solution-ws2022.pdf
-rwx------  1 josef josef    14866 Dec 31 16:37 generate_valid_itsec_flags.sh
drwxr-xr-x  5 josef josef     3072 Dec 31 16:39 Memes
```



Aufgabe 4

b) Wer könnte in die Datei file.txt schreiben?

```
total 150K
drwxr-xr-x  3 josef josef     1024 Dec 31 16:36 .
drwxr-xr-x 27 root  root      1024 Dec 31 16:38 ..
-rw-r--r--  2 josef josef       13 Dec 31 15:54 file.txt
-rw-rw----  1 josef gbs_uel 145013 Dec 31 15:57 gbs-endterm-solution-ws2022.pdf
-rwx------  1 josef josef    14866 Dec 31 16:37 generate_valid_itsec_flags.sh
drwxr-xr-x  5 josef josef     3072 Dec 31 16:39 Memes
```

Aufgabe 4

b) Wer könnte in die Datei file.txt schreiben?

```
total 150K
drwxr-xr-x  3 josef josef     1024 Dec 31 16:36 .
drwxr-xr-x 27 root  root      1024 Dec 31 16:38 ..
-rw-r--r--  2 josef josef       13 Dec 31 15:54 file.txt
-rw-rw----  1 josef gbs_uel 145013 Dec 31 15:57 gbs-endterm-solution-ws2022.pdf
-rwx------  1 josef josef    14866 Dec 31 16:37 generate_valid_itsec_flags.sh
drwxr-xr-x  5 josef josef     3072 Dec 31 16:39 Memes
```

- Sofern Zugriff auf alle parent directories besteht, kann der Benutzer josef in die Datei schreiben

Aufgabe 4

c) Worin besteht der Unterschied zwischen sym- und hardlinks?

```
total 150K
drwxr-xr-x  3 josef josef     1024 Dec 31 16:36 .
drwxr-xr-x 27 root  root      1024 Dec 31 16:38 ..
-rw-r--r--  2 josef josef       13 Dec 31 15:54 file.txt
-rw-rw----  1 josef gbs_uel 145013 Dec 31 15:57 gbs-endterm-solution-ws2022.pdf
-rwx------  1 josef josef    14866 Dec 31 16:37 generate_valid_itsec_flags.sh
drwxr-xr-x  5 josef josef     3072 Dec 31 16:39 Memes
```

Aufgabe 4

c) Worin besteht der Unterschied zwischen sym- und hardlinks?

- Bei Hardlinks verweisen mehrere Einträge in Verzeichnissen gleichermaßen auf eine gemeinsame Inode. Damit sind neben den Nutzdaten auch die ganzen Metadaten geteilt.

- Ein Symlink hingegen beinhaltet lediglich einen Pfad auf eine weitere Datei.
- Wird der Link nun geöffnet, so liest der Kernel den Pfad aus und öffnet stattdessen die entsprechende Datei, auf die verwiesen wird. Sobald die Zieldatei aber gelöscht wird, zeigt der Link auf eine nicht-existente Datei und ist somit invalide.

Aufgabe 4

d) Wo steht der Dateiname?

Aufgabe 4

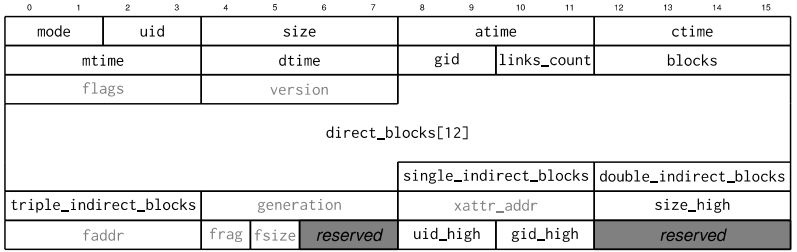
d) Wo steht der Dateiname?

- Der Dateiname steht im Verzeichnis. Damit können verschiedene Hardlinks verschiedene Namen haben, obwohl sie sich eine Inode teilen.
- Außerdem ist der lookup einer Datei im Verzeichnis somit effizienter, da alle Dateinamen in den Blöcken des Verzeichnis selbst stehen und nicht alle Inodes der Dateien des Verzeichnis (welche quer über die Festplatte verteilt sein könnten) gelesen werden müssen.

Aufgabe 4

```
total 150K
drwxr-xr-x  3 josef josef     1024 Dec 31 16:36 .
drwxr-xr-x 27 root  root      1024 Dec 31 16:38 ..
-rw-r--r--  2 josef josef       13 Dec 31 15:54 file.txt
-rw-rw----  1 josef gbs_uel 145013 Dec 31 15:57 gbs-endterm-solution-ws2022.pdf
-rwx------  1 josef josef    14866 Dec 31 16:37 generate_valid_itsec_flags.sh
drwxr-xr-x  5 josef josef     3072 Dec 31 16:39 Memes
```

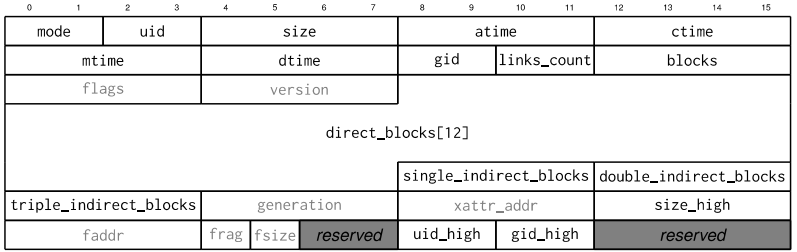
e) Zu welcher Datei im ls-Output der Teilaufgabe a) gehört wohl die Inode im folgenden Hexdump? (Achtung: little-Endian wird verwendet)




```
00003a00  a4 81 e8 03 0d 00 00 00  20 4d b0 63 20 4d b0 63  |........ M.c M.c|
00003a10  20 4d b0 63 00 00 00 00  e8 03 02 00 02 00 00 00  | M.c............|
00003a20  00 00 00 00 08 00 00 00  c2 07 00 00 00 00 00 00  |................|
00003a30  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00003a40  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00003a50  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00003a60  00 00 00 00 3f 16 45 f0  00 00 00 00 00 00 00 00  |....?.E.........|
00003a70  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00003a80  20 00 00 00 c0 05 65 89  c0 05 65 89 c0 05 65 89  | .....e...e...e.|
```

Aufgabe 4

```
total 150K
drwxr-xr-x  3 josef josef     1024 Dec 31 16:36 .
drwxr-xr-x 27 root  root      1024 Dec 31 16:38 ..
-rw-r--r--  2 josef josef       13 Dec 31 15:54 file.txt
-rw-rw----  1 josef gbs_uel 145013 Dec 31 15:57 gbs-endterm-solution-ws2022.pdf
-rwx------  1 josef josef    14866 Dec 31 16:37 generate_valid_itsec_flags.sh
drwxr-xr-x  5 josef josef     3072 Dec 31 16:39 Memes
```



e) Zu welcher Datei im ls-Output der Teilaufgabe a) gehört wohl die Inode im folgenden Hexdump? (Achtung: little-Endian wird verwendet) – file.txt (Größe 13 Byte)

```
00003a00  a4 81 e8 03 0d 00 00 00  20 4d b0 63 20 4d b0 63  |........ M.c M.c|
00003a10  20 4d b0 63 00 00 00 00  e8 03 02 00 02 00 00 00  | M.c............|
00003a20  00 00 00 00 08 00 00 00  c2 07 00 00 00 00 00 00  |................|
00003a30  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00003a40  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00003a50  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00003a60  00 00 00 00 3f 16 45 f0  00 00 00 00 00 00 00 00  |....?.E.........|
00003a70  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00003a80  20 00 00 00 c0 05 65 89  c0 05 65 89 c0 05 65 89  | .....e...e...e.|
```

Aufgabe 4

f) Welche Felder ändern diese Befehle jeweils in der Inode? cat, chmod, chown, touch, ln

Aufgabe 4

f) Welche Felder ändern diese Befehle jeweils in der Inode? cat, chmod, chown, touch, ln

- cat: atime
- chmod: Mode: Rechte, ctime
- chown: Besitzer, Gruppe, ctime
- touch: atime, mtime, ctime
- ln: links_count, ctime