

zudem noch sehr reparaturfreundlich sind.

Hier ein Schnappschuss einer zerlegten „Babynova“. **Siehe Bild 18.** **Bild 19** zeigt die Werkstatt für die Waschmaschinenreparatur.

Alles in Allem kann ich die Einrichtung nur empfehlen, da dort ein praktischer Zugang für den Erwerb von Haushalts- und Unterhaltungsgeräten gepflegt wird. Ich habe mir sagen lassen, dass 10- bis 15-jährige Waschmaschinen bei denen der Renner im Verkauf sind. Günstig zu erwerben und für die Ewigkeit gebaut. Na dann...



Bild 18



Bild 19

Reboot

Unter Windows 10 sollte die Richtung mit den Updates schon bekannt sein. Jeden zweiten Dienstag im Monat kommen die normalen und alle sechs Monate ein „Creators Update“ (oder wie auch immer das heißt) auf das System. Die Wandlung von Windows10 zu einem „rolling release“ ist somit eingeläutet. Theoretisch sollte keine Neuinstallation mehr notwendig sein. Wohlgehemt, ich rede hier von Windows 10 als Clientsystem. Obwohl ich schon MS-Server gesehen habe, wo das Creators Update auch angeboten wurde. Anderes Thema.

Fakt ist, dass im Laufe der Zeit vieles komplexer geworden ist. Letzten Herbst griff eben Microsoft mit ihrem 1809er Update elegant ins Klo. Kann mich aber erinnern, dass vor zwei Jahren? auch mal was unter Debian Linux bei den Updates in die Hose ging. Wo Menschen arbeiten, gibt's eben auch immer welche, die nicht arbeiten :-).

Das Nachjustieren und Zurechtfinden nach so einem Riesenupdate mit mehreren Gigabyte ist zwar für den Endbenutzer ärgerlich, aber eben der Preis für die neue Updatepolitik unter Microsoft. Es geht nicht anders, wenn man immer das „Neueste“ haben will.

Der Zeitfaktor zum Einspielen der Updates ist da mitunter schon enorm, aber was soll man machen? Und wieso braucht das Microsoft System meist einen Reboot nach dem Einspielen von Updates? Wie erklärst Du das einem Laien?

Wieso verlangen Updates bei UNIX-Betriebssystemen nach keinem anschließenden Reboot? Und bei Windows immer (oder meist)? Dafür habe ich zwei Antworten parat. Einmal in Gestalt eines ausführlicheren Klugscheißermodus und dann noch mal trivial.

Fangen wir mit Ersterem an. Zur Begriffserklärung des Terminus „Inode“. Was ist das überhaupt?. Einen Inode kann man sich als Barcode oder ID-Nummer vorstellen. Beim Anlegen einer Datei wird demnach eine Nummer als Referenz auf die angelegte Datei benutzt. Diese Nummer heißt unter Linux „Inode“ und unter Windows „File-Record“. Ist aber im Prinzip dasselbe.

außer dem Dateinamen sind im Inode sämtliche Informationen einer Datei enthalten. Also Besitzer, Berechtigungen, Platz auf der Festplatte, die Anzahl der Hardlinks usw., die auf den Inode verweisen. Erst wenn das System auf den Inode schaut, weiß es, ob es sich um einen Ordner oder eine Datei handelt. Und weil das System nichts über Ordner weiß, hat jeder Ordner zwei Einträge. „.“ und „..“. **Siehe Bilder 20/21.**

Diese Punkte enthalten Informationen über den aktuellen Ordner (ein Punkt) und den jeweils übergeordneten (zwei Punkte).

In modernen Dateisystemen (NTFS, ext4, btrfs, xfs...) gibt es letztendlich keine feste Zuordnung zwischen Dateinamen und der Datei. Einfach weil eine Datei auch verschiedene Namen haben kann. Aber nur einen Inode oder File-Record. In Bild 22 siehst Du, in welchem Inode auf dem Dateisystem (ext4) die Datei „aaa“ gespeichert ist. Du ahnst es schon, es ist zugleich die Artikelüberschrift.

Der 56885659ste Inode beinhaltet die Infos über die Datei „aaa“. Das „total 4“ darüber stellt nur die verbrauchte Blockanzahl der Datei dar. Wichtiger für uns ist aber die „1“ vor „wachbirn“ in der Ausgabe. In dieser Spalte ist immer die Anzahl der Hardlinks, die auf diese Datei verwei-

sen, angegeben. Die Datei gibt es also genau ein mal.

Ein Hardlink ist eine Verknüpfung zwischen Dateinamen und Datei. Jede Datei unter Linux muss somit zumindest einen Eintrag haben, der auf den entsprechenden Inode verweist. Jeder Dateieintrag ist unter Linux ein Hardlink. Jeder. Darum muss in einem Ordner ein neu erzeugter Hardlink auch immer einen anderen Namen haben, da zwei identische Dateinamen sonst auf einen Inode treffen würden. Nicht gut. Ditto in Windows.

Somit kann eine Datei auch unter mehreren Namen am System existieren. Diese verschiedenen Namen treffen sich aber immer auf demselben Inode. **Siehe Bild 23.**

Der rote Pfeil weist unsere Datei aaa mit einem Hardlink aus. Der grüne Pfeil weist auf den Befehl zur Erzeugung eines weiteren Hardlinks hin. Sprich, dort erzeuge ich eine identische Datei mit dem Namen „bbb“. Der gelbe Pfeil zeigt uns nun, dass zwei Dateien im System existieren, die bis auf den Dateinamen identisch sind. Beide Dateien verweisen auf denselben Inode. **Siehe Bild 24.**

Die Datei gibt's also zwei mal. Wenn ich bei einer Datei etwas ändere, übernimmt die andere diese Änderungen auch. So

```
Verzeichnis von C:\Users\ich
11.03.2019  23:23  <DIR>  .
11.03.2019  23:23  <DIR>  ..
09.03.2019  08:53  <DIR>  3D Objects
09.03.2019  08:53  <DIR>  Contacts
11.03.2019  23:22  <DIR>  Desktop
```

Bild 20

```
wachbirn@antichrist:~/personal/pc_news$
häf ä neis dei-$ ls -la
total 20
drwxrwxr-x  2 wachbirn wachbirn 4096 Mär 12 21:17 ./
drwxrwxr-x 12 wachbirn wachbirn 4096 Mär 12 15:38 ../
-rw-r--r--  3 wachbirn wachbirn    2 Mär 12 15:38 aaa
-rw-r--r--  3 wachbirn wachbirn    2 Mär 12 15:38 bbb
```

Bild 21

```
wachbirn@antichrist:~/personal/pc_news$
häf ä neis dei-$ ls -li
total 4
56885659 -rw-r--r-- 1 wachbirn wachbirn 2 Mär 12 15:38 aaa
```

Bild 22