

Linux

Sommerzeit/Winterzeit - Windows vs Linux

Hier mal eine Beispiel, wie man die Verwaltung der Uhrzeit in einem Computer-Betriebssystem NICHT umsetzen sollte.

Leser-Zielgruppe: Microsoft-Lästerer und Unix-Einsteiger

Ich habe neulich, *vor* Beginn der Sommerzeit, auf dem Windows-2000-System an meinem Arbeitsplatz eine Datei angelegt, es war so gegen 12:46 Uhr (habe ich mir in die Datei reingeschrieben). Gestern habe ich sie mir mit 'dir' anzeigen lassen:

```
> C:\sommerzeit>dir sommerzeit.txt
> Datenträger in Laufwerk C: ist Systemplatte
> Datenträgernummer: F85B-FBEF
>
> Verzeichnis von C:\sommerzeit
>
> 24.03.2003  13:46          64 sommerzeit.txt
>     1 Datei(en)          64 Bytes
>     0 Verzeichnis(se), 677.343.232 Bytes frei
```

Man beachte: 13:46 Uhr!

Derselbe Versuch unter Linux (daheim) ergibt, daß die Datei immernoch mit Erstellungszeitpunkt 12:46 angezeigt wird.

Zunächst: Welches Verhalten ist richtig?

Das von Linux natürlich. Wers nicht glaubt, stelle sich folgendes vor: Man fährt zu schnell und wird von der Polizei geblitzt. Das bemerkt man sogar, schaut auf die Uhr, es ist 12:46 Uhr. Einige Tage später erhält man einen Brief mit Bußgeldbescheid, in dem steht, man sei um 13:46 Uhr zu schnell gefahren - was würde man davon wohl halten? Aber genau so arbeitet Windows 2000.

Jetzt fragt man sich natürlich, wie es dazu kommen kann.

Zunächst die Funktionsweise von Linux (bzw. Unix allgemein) (prinzipiell, ohne technische Details): Im Dateisystem wird der Zeitstempel einer Datei *absolut* festgehalten, damit meine ich, in einer Schreibweise, die von Ort (lokaler Zeit und lokaler Zeitzone) und von Sommerzeit unabhängig ist. Dazu verwendet man die Anzahl Sekunden seit Ende 1969 GMT, also in einer bestimmten Zeitzone, die auch keine Sommerzeit kennt. Und zwar immer, egal ob man bei der Installation seines Linux Zeitzone Deutschland oder sonstwas angegeben hat. In dem Augenblick, wo man sich die Datei anzeigen läßt, muß diese Zeit in die gewünschte lokale Zeit umgerechnet werden. Dazu sind Tabellen installiert, in denen steht, wie die gewünschte lokale Zeit (z.B. die deutsche Zeit) zu jedem Zeitpunkt von GMT abweicht. Wenn ich mir also die oben beschriebene Datei mit `ls -l` anzeigen lasse, so wird der

Linux

Zeitstempel genommen (soundsoviel Sekunden seit Ende 1969 GMT), dann wird nachgeschaut, wie die deutsche Zeit zu jenem Zeitpunkt davon abgewichen ist (in diesem Fall war gerade Winterzeit, deshalb eine Stunde Abweichung), und dann kann ls die Uhrzeit entsprechend anzeigen. Es ist deshalb völlig unerheblich, *wann* ich ls -l aufrufe: Die Ausgabe wird immer gleich aussehen.

Auf Windows 2000 läuft die Berechnung offenbar anders, ich vermute wie folgt: Der Datei-Zeitstempel wird auch in einer absoluten Zeit (angenommen: wie bei Unix) abgespeichert. Bei der Anzeige fragt der Befehl dir das System, wie denn die eingestellte Zeitzone von der absoluten Zeit abweicht, und erhält die Antwort "2 Stunden", weil ja gerade Sommerzeit gilt. Daß zum Zeitpunkt der Anlage der Datei die Abweichung nur 1 Stunde betrug, bleibt unberücksichtigt. Somit hängt die Ausgabe von dir davon ab, wann man es aufruft.

Nebenbei: Windows 2000 verhält sich richtig, wenn die Datei in einem Verzeichnis liegt, das per SaMBa von einem Linux-Rechner exportiert wird. Windows 2000 verhält sich falsch, wenn die Datei in einem Verzeichnis liegt, das von einem anderen Windows 2000-Rechner exportiert wird. Der Windows-Explorer zeigt dasselbe an wie der Kommandozeilen-Befehl dir.

Wertung:

Ich halte das Verhalten für typisch für viele Microsoft-Produkte: Oberflächlich toll und Linux oft überlegen. Aber bei genauerem Hinsehen stellt man fest, daß oft die grundlegenden Konzepte völlig mißraten sind (bzw. daß man sich offenbar überhaupt kein Konzept überlegt hat). *)

Für einen Server mit ernstzunehmenden Anwendungen halte ich so ein Verhalten, wie es Windows 2000 zeigt, für blamabel und völlig unakzeptabel. (Oops: das waren natürlich nur die Workstation-Versionen, auf denen ich das ausprobiert habe - vielleicht machen es die Server-Versionen ja besser?! (Ich glaube es nicht.) Dummerweise laufen einige wichtige Anwendungen in meiner Firma auf Win-2000-Workstation-Rechnern.)

*) Andere Anekdote gefällig? Man kann das ganze Repository auf einem SourceSafe-*Server* (SourceSafe ist ein Versionsverwaltungssystem von MS, also sowas wie CVS) ruinieren, indem man aus- und eincheckt und dabei die Uhrzeit des *Client*-Rechners geeignet verstellt.

- -

Microsoft sieht das anders. Das Verhalten wurde mit Windows NT eingeführt und ist offensichtlich bekannt und beabsichtigt. (siehe z.B. Knowledgebase-Artikel Q158588 [1]).

Linux

- > Dazu Verwendet man die Anzahl
- > Sekunden seit Ende 1969 GMT, also in einer bestimmten Zeitzone, die
- > auch keine Sommerzeit kennt.

Eigentlich nimmt man heute UTC.

- > Auf Windows 2000 läuft die Berechnung offenbar anders,

Es kommt auch darauf an, ob man FAT- oder NTFS-Partitionen benutzt. Unter FAT wird die lokale Zeit gespeichert, unter NTFS wird UTC genommen. Das Problem tritt offensichtlich immer auf, wenn eine Datei auf einem NTFS-Volume gespeichert ist und man die Mod.zeit dann mit der C-Lib-Funktion `stat()` ausliest - wie es z.B. `'dir'` macht.

Ich habe zufaelligerweise gerade erst einen einen Artikel dazu gelesen.[2]

[1] <http://support.microsoft.com/?kbid=158588>

[2] <http://www.codeproject.com/datetime/dstbugs.asp>

Eindeutige ID: #0

Verfasser: Markus Stein

Letzte Änderung der FAQ: 2006-02-27 23:14