

SelfLinux-0.13.0



## Netzwerkzeitprotokoll



Autor: Steffen Dettmer ([steffen@dett.de](mailto:steffen@dett.de))  
Formatierung: Matthias Hagedorn ([matthias.hagedorn@selflinux.org](mailto:matthias.hagedorn@selflinux.org))  
Lizenz: GFDL

## Inhaltsverzeichnis

### 1 Einleitung

- 1.1 Über dieses Dokument
- 1.2 Zeitzonen
- 1.3 Weltzeit und Sommerzeit
- 1.4 Zeitverteilung
- 1.5 Server

### 2 Zeit Einstellen

### 3 Der XNTP Deamon

- 3.1 Installation
- 3.2 Konfiguration
- 3.3 Starten
- 3.4 Hierarchien aufbauen

### 4 Überwachung des Servers

- 4.1 Das Kontrollprogramm ntpdc verwenden
- 4.2 Partner Server anzeigen
- 4.3 Details über Server anzeigen
- 4.4 Systeminformationen anzeigen
- 4.5 Automatisierte Überwachung

### 5 Zeit für Windows

- 5.1 Samba Konfigurieren
- 5.2 LogOn Script anlegen
- 5.3 Einsatz

### 6 Zusammenfassung und Ausblick

# 1 Einleitung

Uhrzeiten (zusammen mit einem Datum) sind ein wichtiger Bestandteil des täglichen Lebens. Man verabredet sich für eine bestimmte Zeit oder erwartet eine Fernsehsendung zu einer bestimmten Zeit. Seit Jahrhunderten verwendet die Menschheit Uhren, um die Zeit möglichst genau zu messen. Mit der Kenntnis einer genauen Uhrzeit können auch längere Zeitspannen genau gemessen werden. Inzwischen ist man mit sogenannten Atomuhren in der Lage, mit viel technischem Aufwand die Zeit unheimlich genau zu messen: eine Atomuhr misst die Zeit auf mindestens 0.00000000001 Sekunden genau! Das bedeutet, dass eine Atomuhr nach zehn Jahren um maximal eine Millisekunde oder nach 10.000 Jahren nur um eine Sekunde falsch geht.

Es ist sicherlich einsichtig, dass es wichtig ist, dass auch jeder die gleiche Uhrzeit verwendet. Falsch gehende Armbanduhren können zu bösen Überraschungen beim ersten Rendezvous führen...


Auch bei Rechnersystemen und in Netzwerken spielt die genaue Uhrzeit eine wichtige Rolle: nur so können Protokoll- und Logdateien sinnvoll genutzt werden, Dateiänderungen korrekt dokumentiert und verglichen werden.

Dieser Text beschreibt, wie man es erreicht, dass ein Linuxsystem über das Netzwerkzeitprotokoll NTP über eine sehr genaue Zeit verfügt und diese einem Windowssystem über [Samba](#) zur Verfügung stellen kann. Netzwerkzeit ist ein Basisdienst eines Netzwerkes und sollte immer eingerichtet werden. Dies ist auch nicht sehr aufwendig.

Die Begriffe Uhrzeit und Zeit schließen im Folgenden das Datum grundsätzlich mit ein.

## 1.1 Über dieses Dokument

Dieses Dokument kann nur einen kleinen Überblick über das Thema geben. Viele Details werden nur angerissen oder überhaupt nicht erwähnt. Trotz aller Sorgfalt kann es leider passieren, dass einige der Informationen aus diesem Dokument fehlerhaft sind.

Der Autor  [Steffen Dettmer](#) freut sich natürlich über Kommentare und Anregungen zu diesem Text.

## 1.2 Zeitzonen

Mittags erwarten wir, dass die Sonne den höchsten Stand hat. Nun ist klar, dass dies regional unterschiedlich ist: ist es in Europa Mittag, so herrscht in Australien gerade tiefe Nacht. Also hat man verschiedene Uhrzeiten. Nun wäre es natürlich umständlich, in Berlin ausrechnen zu müssen, wie spät es gerade in Hamburg ist (vermutlich ein paar Sekunden früher), daher hat man sogenannte Zeitzonen eingeführt. Innerhalb einer Zeitzone verwendet man die gleiche Uhrzeit (auch wenn dann die Sonne erst ein paar Minuten nach zwölf Uhr im Zenit steht). Vereinfacht gesprochen hat man jeweils etliche Länder oder grössere Regionen in eine Zeitzone gesteckt, deren Zeit weniger als eine Stunde von einander abweicht. So erhält man 24 Zeitzonen (in der Praxis gibt es allerdings mehr). Im Internet findet man Seiten, die Karten der Zeitzonen zeigen.

## 1.3 Weltzeit und Sommerzeit

Beispielsweise im internationalen Schiffsverkehr, aber auch in internationalen Chaträumen ist es aufwendig, die Zeitzonen entsprechend ineinander umzurechnen. Daher hat man eine Weltzeit **Coordinated Universal Time** UTC definiert. Diese entspricht der Zeit am Nullmeridian und ist damit (ziemlich genau) gleich der **Greenwich Mean Time** (GMT). Die Zeit in Mitteleuropa ist im Winter eine Stunde, im Sommer zwei Stunden weiter (später).

Sommerzeit wurde eingeführt, um das Tageslicht besser ausnutzen zu können. Man kennt das: zweimal im Jahr werden die Uhren um je eine Stunde verstellt. Da man die Uhrzeit im Sommer eine Stunde vorstellt und die UTC keine Sommer- oder Winterzeit unterscheidet, hat man im Sommer also zwei Stunden Differenz.

Bei Zeitsynchronisation verwendet man UTC oder GMT. Die lokale Zeit wird dann vom System entsprechend der lokalen Zeitzone angezeigt. Die Umstellung von Sommer- auf Winterzeit und umgekehrt ist in diesem Modell auch keine Zeitemstellung, sondern lediglich eine Änderung der Zeitzone; so sind **CET** und **MET** Winterzeit, **CEST** und **MEST** sind Varianten dieser Zeitzonen in Sommerzeit.

Das Kommando `date` gibt die aktuelle Systemzeit und die lokale Zeitzone aus:

```
user@linux ~/ $ date
Sun Apr 13 16:08:15 CEST 2003
```

man erkennt, dass wir Sommerzeit haben. Die Weltzeit **UTC** kann das GNU `date` Kommando über die Option `--utc` anzeigen:

```
user@linux ~/ $ date --utc
Sun Apr 13 14:11:15 UTC 2003
```

Mitteuropäische Zeit ist eine Stunde Differenz, die zweite kommt durch die Sommerzeit - daher beträgt die Zeitdifferenz im Winter nur eine Stunde.

## 1.4 Zeitverteilung

Dass es super genaue Atomuhren gibt, hilft einem zunächst erstmal nicht sehr viel: man bräuchte einen grossen Keller und sehr preiswert sind diese Uhren auch nicht. Nun ist es aber möglich, die Atomzeit zu verteilen. Ein bekanntes Beispiel sind die Funkuhren, die die Uhrzeit auf ca. 20 Millisekunden genau über das **DFC77** Protokoll aus Frankfurt/Main per Funk empfangen. Mit dem Satelliten-Navigationssystem GPS ist es sogar möglich, die Uhrzeit auf eine Mikrosekunde genau zu erhalten, jedoch ist dies technisch deutlich aufwendiger.

Neben Funkverbindungen bietet sich das Internet an, um Uhrzeitdaten zu übermitteln. Leider ist die Signallaufzeit im Internet deutlich größer und vor allem ungenau, so dass man einige Tricks benutzen muß, um eine Genauigkeit besser als eine Sekunde zu erreichen. Es gibt Server im Internet, die die Uhrzeit auf Anfrage verschicken. Die physikalisch-technische Bundesanstalt beispielsweise betreibt eine Atomuhr und solche Zeitserver.

Es gibt etliche Programme, die eine genaue Zeitsynchronisation über das Internet ermöglichen.

## 1.5 Server

Häufig wird das Network Time Protocol (NTP) verwendet. Es gibt Server, die über eine externe Quelle, beispielsweise über einen GPS Empfänger, mit einer Atomuhr verbunden sind. Man bezeichnet diese als Stratum 1 Server. Diese können die Zeit mittels NTP zur Verfügung stellen. Server, die ihre Zeit über Stratum 1 Server beziehen, und damit **indirekt** von einer Atomuhr, nennt man Stratum 2 Server. So hat man ein Maß für die **Qualität** der Zeit. Auch Stratum 2 Server sind auf Sekundenbruchteile genau.

Manche Zeitserver im Internet dürfen nur nach Vereinbarung oder Erlaubnis benutzt werden. Insbesondere Stratum 1 Server sind oft stark belastet. Für die meisten Anwendungen ist ein Stratum 2 Server jedoch mehr als ausreichend (im Heimbereich ist eine auf hundertstel Sekunde genaue Uhrzeit sicherlich immer ausreichend).

## 2 Zeit Einstellen

Beim Systemstart, wenn der Kernel bootet, wird die aktuelle Uhrzeit aus der Echtzeituhr des Computers ausgelesen. Diese ist in einem Baustein auf dem Mainboard angebracht. Man erkennt diesen kleinen schwarzen Käfer oft an einer kleinen Uhrengrafik und der Beschriftung **RTC** (Real Time Clock, englisch für Echtzeituhr). Da dieser über eine Batterie gepuffert wird, stimmt bei einem Neustart also auch die Systemzeit in etwa.

Um die Zeit genau einzustellen, gibt es das Tool `ntpdate`. Dieses Programm nimmt Verbindung mit einem oder mehreren Timeservern auf und stellt die Systemuhrzeit ein. Dabei werden etliche Pakete verschickt und empfangen und anschließend wird das halbe Antwortzeitmittel von der Uhrzeit abgezogen. Der Aufruf ist denkbar einfach: lediglich ein paar Servernamen sind als Parameter notwendig. Bei den Einstellungen sind die Standardwerte meistens passend.

Um die Zeit einzustellen, muß man über root-Rechte verfügen. Man schreibt dann beispielsweise:

```
root@linux ~/ # ntpdate clock.isc.org ptbtime1.ptb.de
ntp1-0.cs.tu-berlin.de

12 Jan 21:12:38 ntpdate[2382]: adjust time server 192.53.103.103 offset
3.493504 sec
```

`ntpdate` hat die Linux-Kerneluhr eingestellt, und dabei die Uhrzeit um dreieinhalb Sekunden vorgestellt. Wenn man das Kommando jetzt noch ein paarmal ausführt, gewinnt man einen Eindruck, wie präzise dieser Mechanismus ist: in der Regel wird die Zeit dann nur noch um wenige Millisekunden verstellt, so genau funktioniert das! Um die Genauigkeit zu halten, könnte man `ntpdate` alle paar Stunden laufen lassen.

Nun kann man diese Uhrzeit noch in die BIOS-Echtzeituhr eintragen, damit man nach dem nächsten Neustart eine genauere Zeit hat. Dazu gibt es das Tool `hwclock`. Mit dem Parameter `--systohc` schreibt man die Systemzeit in die **hardware clock** (bei PCs also in den RTC des Mainboards). Der Aufruf sieht dann so aus:

```
root@linux ~/ # hwclock --systohc
```

Das dauert ein paar Sekunden, dann stimmt die Zeit in der Echtzeituhr.

Mit dieser Methode kann man also schon eine Genauigkeit erreichen, die für Heimanwender ausreichend ist. Hier hat man oft Internetwählverbindungen. Da bei Linux bei einer PPP Anwahl das Script `/etc/ppp/ip-up` ausgeführt wird, paßt hier ein Aufruf von `ntpdate` gut rein. Beispielsweise kann man hier einfügen:

```
Datei /etc/ppp/ip-up (Auszug)

#Ein paar Server, die zur Synchronisation verwendet werden
# können. Bitte nicht unbedingt diese verwenden, sondern andere
# aus http://www.eecis.udel.edu/~mills/ntp/ auswählen
/usr/sbin/ntpdate -bs \
                    clock.isc.org \
                    ptbtime1.ptb.de \
                    ntp1-0.cs.tu-berlin.de
```

Die Option `-s` gibt ab, dass man `syslog` für die Ausgaben verwenden möchte. Oft sieht man hier zusätzlich

den Schalter `-b`, dieser gibt an, dass hier mit der Zeit gesprungen werden darf und sollte angegeben werden, wenn man `ntpdate` nur selten startet.

## 3 Der XNTP Daemon

Der `xntpd` ist ein Dienst, der die Zeit eines Linuxservers synchronisiert. Dazu verwendet er, wie `ntpdate`, Zeitserver im Internet. Der Dienst arbeitet jedoch noch aufwendiger. Zunächst sendet er alle paar Minuten (je nach Genauigkeit) Zeitpakete, um die Zeit ständig aktuell zu halten. Dies ist notwendig, da die Kerneluhr (und auch die Hardware- Echtzeituhr) nicht so genau sind und beispielsweise temperaturbedingter Schwankungen unterliegen.

Daher muß die Kerneluhr auch nachgestellt werden. Der `xntpd` macht das Nachstellen allerdings sehr geschickt: er setzt nicht einfach ruckartig die Zeit eine Sekunde zurück (oder natürlich vor), sondern er bremst die Kerneluhr ein kleines bißchen ab, bis die Zeiten gleich sind. Im Prinzip regelt der Dienst also die Geschwindigkeit der Kerneluhr (und gleicht so die temperaturabhängiger Schwankungen aus). So hat man - eine Internetverbindung vorausgesetzt - ständig eine genaue Zeit. Genau heißt hier, auf einen Sekundenbruchteil genau; beispielsweise auf eine Millisekunde genau. Das ist beachtenswert, denn die Antwortzeiten der Server betragen oft mehr als das hundertfache (also eine zehntel Sekunde).

Der Daemon sollte als `ntpd` oder als `xntpd` Paket in jeder Distribution zu finden sein.

### 3.1 Installation

Dieser Text erklärt nicht ausführlich, wie man das `ntpd` Paket installiert, da es in den großen Distributionen verfügbar ist und hier auf die gleiche Weise wie andere Softwarepakete einfach installiert werden kann. Meistens wird es soagr bereits installiert sein, da man diesen Basisdienst auf fast allen Server einsetzt. Man verwendet den Paketmanager oder das Distributionsspezifische Installationswerkzeug, um das Paket gegebenenfalls zu installieren.

Natürlich ist es auch möglich, den Quellcode downzuladen und selbst zu kompilieren. Diese Prozedur basiert auf GNU `autoconf` und ist daher sehr ähnlich zu anderen Softwarepaketen, die GNU `autoconf` verwenden.

### 3.2 Konfiguration

Eine einfache Konfigurationsdatei ist schnell erstellt. Im westentlichen werden ein paar Server eingetragen und der Zugriff darauf etwas beschränkt.

## Datei /etc/ntp.conf

```
#lokale Uhr
server 127.127.1.0
#stratum 10 bedeutet "schlechte Qualität"
fudge 127.127.1.0 stratum 10

#Nun ein paar Server, die zur Synchronisation verwendet werden
# können. Bitte nicht unbedingt diese verwenden, sondern andere
# aus http://www.eecis.udel.edu/~mills/ntp/ auswählen.
# Stratum 2 Server sind mehr als ausreichend.
server ntpsl-0.cs.tu-berlin.de prefer
server clock.isc.org
server ptbtime1.ptb.de
server ptbtime2.ptb.de

#Hat man mehrere NTP Server im LAN, sollte man diese als peer
# konfigurieren:
#peer ntps2-0.bedi.net # local time peer server

#Die ermittelte Laufzeitabweichung des Systems hier speichern:
driftfile /etc/ntp.drift

#Andere Server oder Clients dürfen die Zeit nur abfragen
restrict default notrust nomodify
#localhost darf alles (siehe "ntpd")
restrict 127.0.0.1
#Normalerweise vertrauen wir dem Partnerserver im LAN auch:
#restrict ntps2-0.bedi.net
```

### 3.3 Starten

Vor dem Starten sollte man `ntpdate` ausführen, um eine korrekte Ausgangszeit zu haben. Weicht die Zeit beim Start von `ntpd` stark ab, so wird die Zeit vom `ntpd` nicht automatisch korrigiert, da es sich möglicherweise um Fehler handelt, die manuell korrigiert werden sollten.

Man könnte den `xntpd` einfach von der Kommandozeile starten, aber das kann die verwendete Distribution auch automatisch beim Start über die sogenannten `rc`-Skripte. Bei SuSE-Distributionen kann man beispielsweise über das Konfigurationswerkzeug den Dienst aktivieren lassen und auch per Hand starten:

```
root@linux ~/ # /etc/rc.d/xntpd start
```

Bei anderen Distributionen kann der Name geringfügig abweichen und beispielsweise `ntpd` oder `ntp` lauten.

### 3.4 Hierarchien aufbauen

Hat man viele Server im LAN, so muss man natürlich nicht mit jedem Server die Zeit aus dem Internet heraus synchronisieren. Man kann auch Stratum 2 Server verwenden, um zwei (oder mehrere) interne Stratum 3 Server zu betreiben. Diese sollten untereinander als `peer` konfiguriert sein, um bei Internetausfall höhere Genauigkeit zu bieten.

Die anderen lokalen Server können die Uhrzeit dann von den eigenen Stratum 3 Servern beziehen. Dieses Verfahren ermöglicht immernoch Zeiten auf hundertstel Sekunde genau, was meistens ausreichend sein sollte. Man kann die lokalen Server auch alle untereinander als `peer` konfigurieren, beziehungsweise die mit ausreichend genauen Uhren (manche PCs haben sehr ungenaue Systemuhren). Dies ist jedoch ein grosser

Aufwand, der nicht unbedingt durch Nutzen überwogen wird.

## 4 Überwachung des Servers

Betrieibt man `ntp` Server, so sollte man diese hin und wieder kontrollieren. Weicht die Zeit einmal erheblich ab, so stellt der `ntpd` die Zeit beispielsweise nicht mehr automatisch nach.

### 4.1 Das Kontrollprogramm `ntpd` verwenden

Zur Überwachung gibt es das Werkzeug `ntpd`. Dieses baut eine Verbindung zum Server auf und kann über Kommandos Einstellungen und Statusanzeigen abfragen und setzen. Daher sollte auch die `restrict` Option im Server gesetzt werden, um zu verhindern, daß jeder Optionen ändern kann.

`ntpd` verbindet per Voreinstellung zu `localhost`. Möchte man zu einem anderen Server, so gibt man dessen Name einfach auf der Kommandozeile an.

Nach dem Start befindet sich das Werkzeug im interaktiven Modus, was durch ein Prompt angezeigt wird, und erwartet ein Kommando.

Eine Hilfe gibt es mit dem Kommando `help`.

### 4.2 Partner Server anzeigen

Ein Überblick gibt eine Tabelle, in der die verwendeten Server und Informationen darüber angezeigt werden. Diese Tabelle kann man sich mit dem Kommando `peers` anzeigen lassen:

```
ntpd> peers
      remote                local          st poll reach  delay  offset  disp
=====
*ntp1.ptb.de      193.178.163.162  1 1024  377 0.03285 -0.001386 0.01483
=LOCAL(0)        127.0.0.1        10  64   377 0.00000  0.000000 0.00093
=hora.cs.tu-berl 193.178.163.162  2 1024  377 0.03644 -0.001299 0.01486
+secu.bedi.net   193.178.163.162  2 1024  376 0.11745 -0.003707 0.02843
=ntp2.ptb.de     193.178.163.162  1 1024  377 0.03380 -0.001047 0.01485
=clock.isc.org   193.178.163.162  2 1024  377 0.21767 -0.012278 0.01485
=ntp-nasa.arc.na 5.0.0.0          16 1024   0 0.00000  0.000000 0.00000
=ntp0-rz.rrze.un 5.0.0.0          16 1024   0 0.00000  0.000000 0.00000
0.00000
```

Hier erkennt man, dass mit fünf Servern die Zeit synchron läuft. Zusätzlich werden Informationen über die lokale Systemuhr (LOCAL) angezeigt. Die unteren beiden Einträge stammen von Servern, die vermutlich nicht erreichbar sind und deuten auf eine ungünstige Konfiguration hin (beispielsweise eine Firewall, die diese Pakete abweist).

Die Spalte **offset** gibt an, wie synchron die Uhren laufen. Die Spalte **disp** gibt die Dispersion, also Streuung, der Zeitwerte an. Eine Dispersion größer als 1 deutet auf grobe Fehler hin, ist diese jedoch kleiner als 0.1, sind die Werte in Ordnung.

Beim Start des Servers ist die Dispersion zunächst sehr groß und pegelt sich im Laufe der Zeit auf einen kleineren Wert ein. Dies kann jedoch etliche Minuten dauern!

### 4.3 Details über Server anzeigen

Im Beispiel sah man, das der Server [clock.isc.org](http://clock.isc.org) gute, der Server [ntp.nasa.gov](http://ntp.nasa.gov) ([ntp-nasa.arc.nasa.gov](http://ntp-nasa.arc.nasa.gov)) jedoch keine sinnvollen Werte übermittelt. Das Kommando `pstats` gibt nähere Informationen. Zunächst für den ersten Server:

```
ntpdc> pstats clock.isc.org

remote host:          clock.isc.org
local interface:     62.154.193.196
time last received:  778s
time until next send: 247s
reachability change: 3079649s
packets sent:        19408
packets received:    18658
bad authentication:  0
bogus origin:        0
duplicate:           0
bad dispersion:      11395
bad reference time:  0
candidate order:     3
```

Man sieht, das vor 778 Sekunden Daten empfangen wurden und in 247 Sekunden das nächste Mal ein Datenaustausch stattfinden soll. Der Server ist seit 35 Tagen (3079649 Sekunden) ununterbrochen erreichbar gewesen.

```
ntpdc> pstats ntp.nasa.gov

remote host:          ntp-nasa.arc.nasa.gov
local interface:     7.0.0.0
time last received:  3209635s
time until next send: 10s
reachability change: 3209635s
packets sent:        3150
packets received:    0
bad authentication:  0
bogus origin:        0
duplicate:           0
bad dispersion:      0
bad reference time:  0
candidate order:     0
```

Man erkennt, dass der Server seit 37 Tagen (3209635 Sekunden) nicht erreichbar ist. Möglicherweise ist dieser Server nicht mehr frei verfügbar.

### 4.4 Systeminformationen anzeigen

Für globale Serverinformationen gibt es die Kommandos `sysinfo` und `sysstats`. `sysinfo` gibt den aktuellen Serverstatus aus:

```
ntpdc> sysinfo
```

```
system peer:          ntp1.ptb.de
system peer mode:     client
leap indicator:       00
stratum:              2
precision:           -17
root distance:        0.07309 s
root dispersion:      0.01715 s
reference ID:         [192.53.103.103]
reference time:       c243ecfe.50fd08d4  Sun, Apr 13 2003
16:04:46.316
system flags:         auth monitor ntp kernel stats kernel_sync
jitter:              0.001297 s
stability:           0.003 ppm
broadcastdelay:      0.003998 s
authdelay:           0.000000 s
```

Man erkennt, welcher Server momentan zur Synchronisation verwendet wird (hier [ntp1.ptb.de](http://ntp1.ptb.de)), und das im Clientmode gearbeitet wird. Hat man mehrere NTP Server im LAN, die untereinander auch `peer` sind, arbeitet ein System bei Internetzugangsproblemen im `peer` Modus. Der Beispielserver befindet sich aber im Clientmodus, so dass die Internetverbindung offensichtlich funktioniert.


Die Ausgaben von `systats`:

```
ntpdc> sysstats
system uptime:        19681383
time since reset:     19681383
bad stratum in packet: 0
old version packets: 21927
new version packets: 525853
unknown version number: 0
bad packet length:   0
packets processed:   114444
bad authentication:  0
limitation rejects:  0
```

zeigen beispielsweise, wie lange (in Sekunden) das System schon läuft (hier also 19.681.383 Sekunden, das entspricht ca. 227 Tagen) und das 114.444 Pakete bearbeitet wurden.

## 4.5 Automatisierte Überwachung

Es bietet sich an, den Dienst automatisch überwachen zu lassen: so kann man es nicht vergessen und wird informiert, sobald ein Fehler auftritt.

Man kann das `netsaint` Paket verwenden, das man über  <http://www.netsaint.org> downloaden kann. `Netsaint` hat neben vielen anderen Tests auch ein `check_ntp`, welches NTP Server über das Netzwerk kontrolliert. Antwortet ein Server nicht, oder hat seine Uhrzeit schlechte Qualität, kann ein Information verschickt werden. Der Administrator bekommt dann eine eMail mit einer entsprechenden Fehlermeldung. Über ein Webfrontend kann er sich Details anschauen, beispielsweise die Verfügbarkeit des Dienstes als Diagramm.

## 5 Zeit für Windows

Meistens hat man viele Windows Arbeitstationen im LAN. Es ist natürlich wichtig, auch diesen eine genaue Zeit einzustellen. Windows verfügt von Haus aus über keinen genauen, präzisen Mechanismus, jedoch ist für Workstations im LAN eine nur sekundengenaue Uhrzeit mehr als ausreichend.


Dieser Abschnitt beschreibt, wie man die Uhrzeit über einen Samba-Server an Windows verteilen kann.

### 5.1 Samba Konfigurieren

/etc/smb.conf (Auszug)

```
; Zeitsynchronisation mit Server. Optionen:  
; time server, logon server, logon script, netlogon share  
  
; (Samba's) Time Server aktivieren  
time server = yes  
  
; LogOn Server für logon script aktivieren  
logon server = yes  
  
; Das Script, relativ zur LogOn-Share  
logon script = scripts/logon.bat  
  
; Und die Share mit dem Script  
[netlogon]  
comment = Netlogon Service  
browseable = no  
read only = yes  
public = yes  
path = /var/spool/samba/netlogon
```

### 5.2 LogOn Script anlegen

Das **LogOn**-Script benötigt nur eine Kommandozeile zusätzlich (ein Aufruf von `net time`). Hat man bereits eines, zu fügt man diesen Aufruf einfach dazu. Sicherheitshalber hier ein Beispiel-Script (danke an  [Carsten Heinrici](#)):

netlogon/scripts/logon.bat

```
rem sync time with server on logon  
net time /set /yes
```

### 5.3 Einsatz

Wenn man alles nochmals überprüft hat, kann man den **Samba**-Server neustarten. Loggt sich ab jetzt ein Windows-Benutzer auf diesem Server ein, so wird automatisch das Logon-Script `logon.bat` ausgeführt. Damit wird die Zeit automatisch beim Einloggen gesetzt.

Mit Windows 95/98/ME ist das schon alles. Bei Windows 2000 muß man jedoch davon ausgehen, dass der Benutzer nicht berechtigt ist, die Uhrzeit zu setzen. Hier muß das Windowssystem entsprechend konfiguriert

werden.

## 6 Zusammenfassung und Ausblick

Die Werkzeuge `ntpdate` (insbesondere für Dial-Up-Verbindungen) und `ntpd` (insbesondere für Server mit permanenter Internetanbindung) für die Zeitsynchronisation mit NTP wurden hier kurz vorgestellt. Dieses Protokoll und der Dienst ist sehr mächtig, hier wurde nur auf die wichtigsten Optionen eingegangen.

Es gibt einfache Werkzeuge, wie `netdate` und `rdate`, die auf ganz einfache Weise Zeiten synchronisieren, jedoch wesentlich ungenauer sind und daher nicht so gerne benutzt werden. Weiterhin gibt es das Paket `chrony`, das auch NTP verwendet und damit sehr genau ist.

Im Internet finden sich viele Informationen zu Zeiten, beispielsweise auf der Homepage des NTP Projektes:

 <http://www.ntp.org/>

David L. Mills pflegt eine Seite mit Informationen rund um das Thema:

 <http://www.eecis.udel.edu/~mills/ntp/>

die oft referenziert wird und beispielsweise Listen öffentlicher Zeitserver enthält.