

Korte uitleg: De tijd

Hoe gaat het systeem om met de datum en tijd. Eerst een paar begrippen.

De Hardware Clock van de PC

Wat in het algemeen de "hardware clock" word genoemd is een "real time clock" (die dus de werkelijke tijd bijhoudt) op het PC moederbord. Hij wordt gevoed door een batterij (ook bekend als de CMOS battery) zodat de klok ook doordraait als de PC uitstaat en de voedingskabel is losgemaakt. Deze klok houdt de datum en tijd bij.

Tijd in een Unix/Linux operating systeem

Een operating systeem heeft voor allerlei zaken de datum/tijd nodig. Bijvoorbeeld om tijdstempels toe te kennen aan gebeurtenissen zoals het aanmaken van een bestand en logging. Unix/Linux telt daarvoor de secondes sinds 1 januari 1970 om 0:00 uur UTC. UTC (Coordinated Universal Time) is voor ons gewone mensen hetzelfde als GMT (Greenwich Mean Time). Beiden zijn niet hetzelfde als de tijd die in Engeland wordt gebruikt (BST)! Dit tijdstip (1970-01-01 00:00:00 UTC) wordt de "epoch" genoemd.

Dit getal in secondes werd oorspronkelijk in een 32-bit integer met tekenbit opgeslagen. Dit betekent dat op 2038-01-19 03:14:08 UTC de 32-bit integer overloopt en dus negatief wordt (1901-12-13 20:45:52). Er is nog tijd genoeg om daar iets aan te doen, bijv. 64-bit gebruiken. Dan kan de tijd verder lopen dan de leeftijd van het universum nu is.

Je vindt die tijdstempels terug als o.a. het aanmaaktijdstip van een bestand:

```
henk@boven:~/test> ls -l --time-style=+%s file
-rw-r--r-- 1 henk wij 8 1441202940 file
henk@boven:~/test>
```

Ga er maar aan rekenen. Het is ergens gedurende 2015-09-02.

Je kunt ook de huidige stand van de teller opvragen:

```
henk@boven:~/test> date +%s
1478441630
henk@boven:~/test>
```

Uiteraard wil je als mens iets ander zien dan deze getallen van meer dan 1 miljard secondes.

Tijdzones

De mens heeft zijn tijd altijd verankerd aan het tijdstip dat de zon op een dag op zijn hoogst staat: lokale tijd. Zoals we weten is men er toe overgegaan om deze lokale tijd in zones samen te vatten, en wel 24 zones rond de aarde. De grenzen van deze zones zijn niet recht, maar lopen vaak langs landsgrenzen, zodat binnen (een deel van een) land toch dezelfde tijdzone wordt gebruikt. Ook zijn sommige landen ertoe overgegaan om gedurende een deel van het jaar de tijd met één (of meer) uur te wijzigen. De zogenaamde Zomertijd. Al deze afwijkingen van de zuivere 24 tijdzones worden door de politiek vastgesteld. Wijzigingen worden door de politiek het liefst op een zo laat mogelijk tijdstip bekend gemaakt.

Unix/Linux en de tijdzones

In Unix begreep men snel dat een mens, die ingelogd is op zijn systeem, graag een datum/tijd wil zien die overeenkomt met zijn plaatselijke kalender en klok. Omdat mensen vanuit de hele wereld kunnen zijn ingelogd op systemen verspreid over de hele wereld, moet het mogelijk zijn dit voor iedere gebruiker apart te regelen. Dat is iets voor een proces omgevingsvariable (zie *Korte uitleg: Environment variabelen (zoals PATH, DISPLAY, LANG, ...)*). Gekozen is voor de variabele met de naam `TZ`. Als de gebruiker die bij het inloggen goed zet (bijv. in `.profile` bij inloggen in de CLI), dan kunnen al zijn processen (als kind proces) die `TZ` variabele uitlezen en gebruiken.

Hoewel het er in eerste instantie op lijkt dat een eenvoudige waarde als `TZ=MET` (voor Midden Europese Tijd) daarvoor genoeg is, is dit niet zo. Een tijdzone van een land kan een zomertijd hebben of niet, en die zomertijd was er misschien niet altijd en hij begon en eindigde misschien niet altijd op hetzelfde tijdstip in ieder jaar. In Linux heeft de `TZ` variabele een waarde die verwijst naar een Time Zone Database. De database bestaat uit binaire bestanden in `/usr/share/zoneinfo`. Sommige bestanden daar zijn gegroepeerd in directories, zoals `/usr/share/zoneinfo/Europe`. Daarin bevindt zich o.a. `/usr/share/zoneinfo/Europe/Amsterdam`.

Als de `TZ` variabele er niet is wordt `/etc/localtime` gebruikt:

```
henk@boven:~/test> l /etc/localtime
lrwxrwxrwx 1 root root 36 16 jul 09:51 /etc/localtime -> /usr/share/zoneinfo/Europe/Amsterdam
henk@boven:~/test>
```

Zoals je ziet verwijst dat met een symbolische link naar het bestand dat op dit systeem bij verstek gebruikt wordt.

Network Time Protocol

Door het ontstaan van netwerken (en het internet) werd het steeds belangrijker dat systemen synchroon lopen. Die netwerken maken het ook mogelijk om de tijd van een ander systeem, dat beter bij de tijd is, te halen. Zo'n ander systeem kan bijvoorbeeld direct aan een atoomklok hangen. Hievoor werd het Network Time Protocol (NTP) ontwikkeld. Het protocol en de

erachter liggende ideeën zijn nogal ingewikkeld (er wordt zelfs rekening gehouden met de vertraging in het netwerk), maar het gebruik is zeer eenvoudig.

Met YaST > Netwerkservices > NTP-configuratie is de NTP daemon in te stellen en te starten. Het enige dat je eigenlijk moet invullen is een Server. Liefst netwerksgewijs zo dichtbij mogelijk. Als je provider er één beschikbaar stelt, gebruik die dan. Anders is nl.pool.ntp.org een goed alternatief.

Als de NTP daemon (`ntpd`) loopt, is hij niet alleen Client, maar ook Server. je kunt dus andere Client systemen er naar laten verwijzen. Bij sommige bedrijfsnetwerken is het handig om een centraal systeem voor dit soort zaken in te richten, waar alle andere systemen gebruik van maken.

Hoe blijft het systeem bij de tijd in de praktijk

Tijdens het booten wordt in een vroeg stadium de tijd door de Kernel van de Hardware Clock uitgelezen en gebruikt als systeemklok. Zodra het netwerk gebruikt kan worden en `ntpd` is gestart wordt de klok gecorrigeert. Tijdens het draaien van het systeem corrigeert `ntpd` de systeemklok regelmatig.

Bij het stoppen van het systeem wordt de tijd naar de Hardware Clock geschreven, zodat deze weer zo correct mogelijk is.

Als je een multi-boot systeem hebt, zullen het (de) andere syste(m)en hetzelfde doen. Zolang dat ook Linux systemen zijn is dat geen probleem. Echter Microsoft Windows schrijft niet de tijd in UTC, maar de lokale tijd naar de Hardware Clock en bij het lezen wordt natuurlijk verondersteld dat de Hardware Clock lokale tijd levert. Er is een optie om openSUSE ook de Hardware Clock met lokale tijd te laten behandelen. Indien nodig, gebruik dat dan. Overigens schijnen de nieuwste Microsoft Windows systemen ook met UTC te kunnen werken. Controleer dus eerst wat er gebeurt en of je dat in kunt stellen.

Configuratie van tijd en tijdzone

Tijdens installatie, of als je YaST > Systeem > Datum en tijd aanroept, krijg je een venster met een wereldkaart en daaronder uitklapmenu's. Kies eerst in het linker menu (Region) en daarna in het rechtermenu (Time_zone), bijvoorbeeld Europa en Nederland. Linksonder staat: Hardware Clock Set to UTC. Dat is normaal aangevinkt. Als je multi-boot met MS Windows moet dat misschien uit. Rechtsonder staat datum en tijd in de lokale tijdzone. Tijdens installatie is het goed om dat zo goed mogelijk te zetten. Als NTP eenmaal draait (en dat wordt hier ook aangegeven) is dat niet zo nodig. NTP zal dat snel genoeg bijwerken.

De tijdzone die je hier hebt ingevuld, wordt de verstek tijdzone van het systeem. D.w.z. YaST maakt de link van `/etc/localtime` aan (zoals hierboven uitgelegd).

Bijwerken van de Time Zone Database

Zoals hierboven al uitgelegd is, wijzigen autoriteiten de tijdzones onregelmatig: zomertijd wordt ingevoerd of afgeschaft, de tijd waarop die ingaat of eindigt wordt gewijzigd, landsdelen gaan naar een andere zone, enz. Dit levert dus wijzigingen in de database op. Die worden aangeboden als Recommended in de Update repos. Ze zijn niet altijd op tijd omdat de politiek kennelijk vindt dat 48 uur genoeg is om zo'n wijziging door te voeren.

De gebruiker

De gebruiker van een multi-user systeem kan natuurlijk, al dan niet tijdelijk, een andere tijdzone willen gebruiken dan de verstekwaarde van het systeem.

In het algemeen is de `TZ` variabele niet gezet in de gebruikersomgeving en dus wordt de hierboven ingestelde tijdzone (de symlink van `/etc/localtime`) gebruikt.

Indien de gebruiker een andere tijdzone wil gebruiken kan hij de `TZ` variable zetten. Bijvoorbeeld in `~/.profile`:

```
export TZ=Asia/Kathmandu
```

Dat zet de tijdzone voor alle sessies (uiteeraard de CLI, maar ook voor KDE). Het kan uiteraard ook voor één commando:

```
henk@boven:~> date +%s ; TZ=UTC date ; date ; TZ=Asia/Kathmandu date
1478509363
ma nov  7 09:02:43 UTC 2016
ma nov  7 10:02:43 CET 2016
ma nov  7 14:47:43 NPT 2016
henk@boven:~>
```

De eerste regel geeft het aantal secondes sinds de epoch.

De tweede regel geeft dat omgerekend in UTC.

De derde regel geeft dat voor Nederland (Central European Time), dat loopt één uur voor op UTC.

De vierde regel geeft dat Nepal vier uur en drie kwartier voorloopt op Nederland.

Een andere:

```
henk@boven:~/test> ls -l --time-style=%s spsp ;\
> TZ=UTC ls -l --time-style=long-iso spsp ;\
> ls -l --time-style=long-iso spsp ;\
> TZ=Asia/Kathmandu ls -l --time-style=long-iso spsp
-rw----- 1 henk wij 3 1364750999 spsp
-rw----- 1 henk wij 3 2013-03-31 17:29 spsp
-rw----- 1 henk wij 3 2013-03-31 19:29 spsp
```

```
-rw----- 1 henk wij 3 2013-03-31 23:14 spsp  
henk@boven:~/test>
```

We zien hier de tijd van aanmaken van het bestand (de `ctime`, zie *Korte uitleg: Directories, meta-data van bestanden, inodes*). Het getal in de eerste regel is ook echt de waarde die in de inode van het bestand (zie *Korte uitleg: Directories, meta-data van bestanden, inodes*) staat.

Aangezien het bestand tijdens de Nederlandse zomertijd is aangemaakt is het verschil met UTC hier twee uur en het verschil met Nepal (dat geen zomertijd kent) drie uur en drie kwartier.