



Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

Configuración del reloj II (práctica) (16388 lectures)

Per **Antonio Tejada Lacaci**, [Wildfred](#) ()

Creado el 16/11/1999 00:00 modificado el 16/11/1999 00:00

Complemento práctico a la [explicación teórica del funcionamiento del reloj en linux](#)⁽¹⁾. Está tomado de un mensaje de Conrado Badenas en [debian-user-es](#).

Antes que nada hay que saber si el reloj del sistema usa GMT (o UTC). Si en /etc/default/rcS la variable GMT es "" entonces la hora es local, pero si GMT es "-u" entonces hay que usar la opción -u en los comandos en que se escriba [-u]. Además, si la hora del sistema es local, hay que asegurarse de que el sistema conoce el timezone. El timezone viene definido en el fichero /etc/timezone, y/o en la variable de entorno TZ.

Además has de tener los comandos hwclock (en el paquete base/util-linux) y netstd (en el paquete net/netstd).

Las fuentes externas (extenas al sistema) de hora pueden ser: el reloj hardware, un reloj que no pueda leer directamente sino a través de la información que le proporcione una persona que mira ese reloj (un reloj de pulsera, o de servicio telefónico, o el del teletexto), y un reloj en un ordenador servidor de tiempo accesible via internet. Por ello, según tus posibilidades tienes que elegir alguna de estas soluciones:

a) Un ordenador completamente aislado del mundo:

a.1: Tras instalar el paquete admin/adjtimex, lo primero es borrar todos los ficheros antiguos con datos generados automáticamente (quieres tener pleno control):

```
# cat <<EOF >/etc/adjtime
> 0.0 0 0.0
> EOF
# rm /var/log/clocks.log
```

a.2: Cógete un buen reloj (de pulsera, de pared, de lo que sea) o pon el teletexto de alguna cadena de televisión o llama a información horaria.

Tecllea "adjtimex [-u] -w <enter>" (la opción -u sólo hay que ponerla si trabajas con hora universal, y no con hora local) y vuelve a pulsar <enter> cuando sepas la hora exacta (el segundero del reloj marca exactamente 00, por el teléfono suena la señal horaria "pip", ...) entonces di qué hora era cuando has pulsado <enter> y qué error adjudicas a este tiempo (cuanto tiempo pasa desde que es la hora hasta que pulsas <enter>: yo pongo 0.1 segundos: para estimar tus reflejos y capacidad y anticipación te sugiero que pongas un cronómetro en marcha e intentes pararlo cuando marque 10 segundos exactos).

Cada cierto tiempo (media hora, una hora) vuelve a hacer "adjtimex [-u] -w" (-u si el reloj del sistema va en UTC, sin -u si es hora local) para darle información de la hora que es, y entonces "adjtimex -r" para mostrar las estimaciones (ajuste por mínimos cuadrados) de los errores sistemáticos del reloj del sistema y del reloj hardware.

a.3: Entonces, cuando tenga suficientes datos buenos, en la información que te da "adjtimex -r" aparecerá el "suggested tick" y el "suggested freq". Imagina que te sale:

least-squares solution:

cmos_error = -215 +- 11 ppm suggested adjustment = 18.5921 sec/day

sys_error = 785 +- 11 ppm suggested tick = 9992 freq = 963288

note: clock variations and unstated data errors may mean that the least squares solution has a bigger uncertainty than estimated here



entonces, edita el fichero `/etc/rc.boot/adjtimex` y cambia las líneas

```
TICK=10000
```

```
FREQ=0
```

por las líneas

```
TICK=9992
```

```
FREQ=963288
```

para que las variables del kernel tengan los valores apropiados cuando arranques de nuevo. Además puedes fijarte en cuál de los dos relojes es mejor: en este caso el de hardware (cmos) tiene un error menor (215 partes por millón = 19 segundos al día) que el del sistema (785 partes por millón = 68 segundos al día).

Además tendrás que hacer `"adjtimex -tick 9992 -frequency 963288"` para eliminar las variaciones sistemáticas del reloj del sistema hasta que no lo apagues de nuevo.

a.4: Una vez eliminadas las variaciones sistemáticas (systematic drift) hay que eliminar el error sistemático constante.

Para ello, puedes hacerlo con:

```
# cat <<EOF > /etc/adjtime
```

```
> 18.5921 0 0.0 (el 18.5921 aparecía haciendo "adjtimex -r")
```

```
> EOF
```

```
# date [-u] --set "22:32:50" (la hora que sea cuando teclees <enter>)
```

```
# hwclock [-u] --systohc
```

a.5: De esta forma tendrás los dos relojes siempre sincronizados a tu fuente externa y sincronizados entre sí. Si detectas que la sincronización falla al cabo de unos días, vuelve a hacerlo todo desde el punto a.1.

b) Un ordenador que tiene una conexión intermitente a internet (por ejemplo, si usas modem y sólo lo enchufas unos pocos minutos al día: al empezar la jornada laboral, y al terminarla para leer el correo electrónico)

b.1: Exactamente igual que en a.1.

b.2: Es igual que en el apartado a) salvo que en vez de usar una fuente externa manual usas un servidor de tiempo. Yo en la facultad uso el servidor del campus que está en la IP 147.156.1.3. Entonces, en vez de `"adjtimex [-u] -w"` de antes usas `"adjtimex [-u] --host 147.156.1.3"`.

Ejecutas este comando un par de veces con una separación de unas horas, y `"adjtimex -r"` te dará unos valores muy precisos del systematic drift (variación sistemática) de los relojes hardware y del sistema.

b.3: Exactamente igual que en a.3

b.4: El error sistemático constante de ambos relojes se elimina muy fácilmente con

```
# cat <<EOF > /etc/adjtime
```

```
> 18.5921 0 0.0 (el 18.5921 aparecía haciendo "adjtimex -r")
```

```
> EOF
```

```
# netdate 147.156.1.3
```

```
# hwclock [-u] --systohc
```

b.5: Los relojes hardware y del sistema están sincronizados entre sí y con el reloj del servidor de tiempo. Si al cabo de unas semanas ves que se producen variaciones sistemáticas, vuelve a empezar desde el punto b.1.

c) Un ordenador con una conexión permanente a internet (como lo que tengo en la facultad: una red local en todo el campus con la que estoy siempre conectado)

c.1: Ahora no se necesita el paquete `adjtimex`, pues es mejor usar el demonio `xntpd`. Para ello instala el paquete `net/xntp3`. Cuando lo configures (le tendrás que decir cuál es tu servidor de tiempo, por ejemplo 147.156.1.3 para los ordenadores de la Universidad de Valencia) dile que quieres ejecutar el comando `ntpdate` antes de ejecutar el demonio `xntpd` cuando el sistema arranca, con lo que siempre tendrás una buena hora de sistema desde el momento del arranque.

c.2: Para ver como funciona todo el cotarro sin tener que esperar al siguiente arranque puedes hacer:

```
# /etc/init.d/xntp3 start
```

```
# hwclock [-u] --systohc
```



c.3: Y ya no tendrás que preocuparte nunca más de que el reloj del sistema tenga una variación sistemática, salvo que el servidor de tiempo se caiga o te corten el cable Ethernet, o alguna otra desgracia.

c.4: Además, el script `/etc/init.d/hwclock.sh`, que se ejecuta al arrancar y al apagar el sistema te asegura que al apagar el sistema el reloj hardware se actualizará con la hora del sistema (que es perfecta gracias a `xntpd`), y otros SOs que usan este reloj (como Mirdows 9x y TOS) se beneficiarán de la potencia de Linux. Fíjate que al apagar el sistema aparece un mensaje como éste:

CMOS clock updated to Mon Nov 15 23:43:44 CET 1999.

Lista de enlaces de este artículo:

1. <http://bulma.net/body.phtml?nIdNoticia=153&nIdComentario=-1>
-

E-mail del autor: wildfred_ARROBA_teleline.es

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=154>