



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

Configuración del reloj I (teoría) (36964 lectures)

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

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

Este artículo es una explicación exhaustiva del funcionamiento del reloj bajo Linux, parte teórica. Está tomado de un mensaje de Conrado Badenas en [debian-user-es](#). También puedes ver la [parte práctica](#)⁽¹⁾ de la explicación.

En Linux hay dos relojes: el de la BIOS (o el de CMOS, o RTC, o reloj hardware: tiene muchos nombres) y el del sistema. El que importa es el del sistema, pero al arrancar el kernel no sabe qué hora es. Para que el sistema sepa la hora el primer programa que lanza el kernel (el programa `init`) se la pregunta al reloj hardware inmediatamente después del arranque (el trabajo sucio lo hace `/etc/rcS.d/S50hwclock.sh --> /etc/init.d/hwclock.sh`, que se instala gracias al paquete `base/util-linux`). Por entonces el kernel no sabe nada de variables del sistema tipo TZ. Lo único que necesita saber es si el reloj hardware mantiene una hora local o la hora universal (GMT, UTC). Eso lo sabe mirando qué valor tiene la variable GMT en el fichero `/etc/default/rcS` (fichero que creo que se crea cuando instalas Debian por primera vez y te pregunta entre otras cosas qué tipo de hora mantiene tu reloj hardware). Entonces pone la hora del sistema como la hora del reloj hardware. Cuando durante el arranque aparece en pantalla

Local time: Mon Nov 15 19:42:10 CET 1999

es que el `hwclock.sh` ha actuado.

Desde entonces tienes el reloj del sistema con la misma hora que el reloj hardware. Pero cada uno de ellos se mueve a diferente velocidad porque tienen distintos mecanismos de funcionamiento. El hardware funciona con un oscilador de cuarzo o lo que sea, y el de sistema mediante una interrupción de tiempo (la típica `timer interrupt` que forma parte del estándar ISA). Sólo vuelven a coincidir los dos relojes cuando se apaga el sistema y entre otros se ejecuta el script `/etc/init.d/hwclock.sh` con la opción `stop`: entonces la hora del sistema se guarda en el reloj hardware, y al cabo de pocos segundos tienes el ordenador apagado. Este sistema de sincronización presupone que justo antes de apagar el ordenador, la hora del sistema es más fiable que la hora del reloj hardware.

Con el sistema en marcha puedes modificar cualesquiera de estos dos relojes, aunque no es aconsejable modificar el reloj del sistema con el comando `date` porque entonces haces que haya discontinuidades en la hora (aunque no sé por qué eso es tan malo). El reloj hardware se puede modificar con el comando `hwclock` cada vez que quieras.

Para eliminar errores sistemáticos del reloj hardware lo mejor es usar una fuente fiable de la hora (creo que es más fiable llamar a información horaria que usar tu reloj de pulsera, pero no lo he comprobado) para poner el reloj hardware a la hora que sea con `"hwclock --set ..."` (has de ser `root` para poner el reloj), y hacerlo cuando enciendas el ordenador y bastantes horas después estando enchufado (eso es para que no se ejecute `/etc/init.d/hwclock.sh` con la opción `stop` y te joda el invento). Así, tendrás en el fichero `/etc/adjtime` la información necesaria para poner siempre el reloj hardware en hora con un simple `"hwclock --adjust"`.

Sin embargo, esto no funciona si en algún momento apagas el ordenador, se ejecuta `/etc/init.d/hwclock.sh` con la opción `stop` (que te hace un `hwclock --systohc`), y por una de esas la hora del sistema no es exacta.

Entonces, en `/etc/adjtime` tendrías la información para ajustar el reloj hardware al reloj del sistema, lo cual es un problema si el reloj del sistema no es suficientemente fiable.

Por eso, creo que la solución es tener siempre el reloj del sistema funcionando perfectamente. Pero no es aconsejable poner el reloj del sistema con el comando `date` porque eso provocaría una discontinuidad en la hora, y además habría que repetirlo continuamente cada vez que se observara un atraso o adelanto considerable. Por ello es mejor usar el



comando adjtimex (paquete admin/adjtimex) que cambia la hora del sistema de forma progresiva (suave) y sistemática (siempre). Para ello modifica ciertas variables del kernel (no confundir con la variable de entorno TZ) que se usan para determinar la hora: tick y frequency para que el reloj del sistema se ajuste a una fuente externa. Dependiendo de la fuente externa se usa con unas opciones u otras.

Lista de enlaces de este artículo:

1. <http://bulma.net/body.phtml?nIdNoticia=154&nIdComentario=-1>
-

E-mail del autor: wilfred_ARROBA_teleline.es

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=153>