

NOMBRE

gettimeofday, settimeofday - pone u obtiene la hora

SINOPSIS

```
#include <sys/time.h>
#include <unistd.h>
```

```
int gettimeofday(struct timeval *tv, struct timezone *tz);
int settimeofday(const struct timeval *tv, const struct
timezone *tz);
```

DESCRIPCIÓN

gettimeofday y settimeofday pueden poner tanto la hora como una zona horaria. tv es una estructura timeval, tal como se especifica en /usr/include/sys/time.h:

```
struct timeval {
    long    tv_sec;        /* segundos */
    long    tv_usec;      /* microsegundos */
};
```

y tz es una estructura timezone :

```
struct timezone {
    int    tz_minuteswest; /* minutos al O de Greenwich */
    int    tz_dsttime;     /* tipo de corrección horaria invierno/verano */
};
```

El uso de la estructura timezone es obsoleto; el campo tz_dsttime nunca ha sido utilizado en Linux - no lo ha sido y nunca será soportado por libc o glibc. Todas y cada una de las ocurrencias de este campo en el fuente del núcleo (distintas de la declaración) son un fallo. Por tanto, lo siguiente es puramente de interés histórico.

El campo tz_dsttime contiene una constante simbólica (los valores se dan más abajo) que indica en qué parte del año la `corrección horaria invierno/verano' (Daylight Saving Time, DST, también conocida como `horario de ahorro energético') está en vigor. (Nota: su valor es constante a lo largo del año - no indica si DST está o no en vigor, sólo selecciona un algoritmo). Los algoritmos de corrección horaria invierno/verano se definen como sigue:

```
DST_NONE      /* no hay */
DST_USA       /* estilo EE.UU. */
DST_AUST      /* estilo australiano */
DST_WET       /* como en Europa Occidental */
DST_MET       /* como en Europa Central */
DST_EET       /* como en Europa del Este */
DST_CAN       /* Canadá */
DST_GB        /* Gran Bretaña e Irlanda */
DST_RUM       /* Rumanía */
DST_TUR       /* Turquía */
DST_AUSTALT   /* est. australiano con cambio en 1986 */
```

Naturalmente, resultó ser que el periodo en el que la corrección horaria invierno/verano está en vigor no se podía dar mediante un simple algoritmo, uno por país: de hecho, este periodo se determina por decisiones políticas impredecibles. Por tanto, este método de representación de las zonas horarias se ha abandonado. Bajo Linux, en una llamada a gettimeofday el campo tz_dsttime debe ser cero.

Bajo Linux hay algunas semánticas de `Warp clock' peculiares asociadas a la llamada al sistema settimeofday si en

la primera llamada de todas (después del arranque) que tenga un argumento tz no NULL, el argumento tv es NULL y el campo tz_minuteswest es no cero. En tal caso, se asume que el reloj CMOS está en el horario local y que tiene que ser incrementado en esta cantidad para obtener el horario del sistema UTC. No cabe duda que usar esta característica es una mala idea.

Para operar sobre la estructura timeval se definen las siguientes macros:

```
#define timerisset(tvp)\
    ((tvp)->tv_sec || (tvp)->tv_usec)
#define timercmp(tvp, uvp, cmp)\
    ((tvp)->tv_sec cmp (uvp)->tv_sec ||\
    (tvp)->tv_sec == (uvp)->tv_sec &&\
    (tvp)->tv_usec cmp (uvp)->tv_usec)
#define timerclear(tvp)\
    ((tvp)->tv_sec = (tvp)->tv_usec = 0)
```

Si tv o tz es nulo, la estructura correspondiente no se ajusta ni se devuelve.

Solamente el super-usuario puede emplear settimeofday.

VALOR DEVUELTO

gettimeofday y settimeofday devuelven 0 en caso de éxito ó -1 si ocurre un fallo (en cuyo caso errno toma un valor apropiado).

ERRORES

EPERM Alguien que no es el super-usuario ha llamado a settimeofday

EINVAL La zona horaria (o algo más) es inválida.

EFAULT Uno de tv o tz apuntaba afuera de su espacio de direcciones accesible.

CONFORME A

SVr4, BSD 4.3

VÉASE TAMBIÉN

date(1), adjtimex(2), time(2), ctime(3), ftime(3)

NOMBRE

adjtimex - ajusta el reloj del núcleo

SINOPSIS

```
#include <sys/timex.h>

int adjtimex(struct timex *buf);
```

DESCRIPCIÓN

Linux usa el algoritmo de David L. Mills para ajustar el reloj. adjtimex lee y opcionalmente prepara parámetros de ajuste para dicho algoritmo.

adjtimex toma un indicador a una estructura timex, actualiza los parámetros del núcleo a partir de los valores de los campos, y devuelve la misma estructura con los valores actuales del núcleo. Esta estructura se declara como sigue a continuación:

```
struct timex
{
    int modes;           /* selector de modo */
    long offset;        /* ajuste de la hora (usec) */
    long freq;          /* ajuste de la frecuencia (ppm escalada) */
    long maxerror;      /* error máximo (usec) */
    long esterror;      /* error estimado (usec) */
    int status;         /* comando/estado del reloj */
    long constant;      /* pll constante de tiempo */
    long precision;     /* precisión del reloj(usec) (sólo lectura) */
    long tolerance;     /* tolerancia de la frecuencia del reloj (ppm)
                        (sólo lectura) */
    struct timeval time; /* hora actual (sólo lectura) */
    long tick;          /* usecs entre ticks del reloj */
};
```

El campo modes determina que parámetros serán asignados, si es que hay alguno. Puede contener una combinación (mediante una operación-O lógica) de cero o más elementos de la lista siguiente:

```
#define ADJ_OFFSET      0x0001 /* ajuste de la hora */
#define ADJ_FREQUENCY  0x0002 /* ajuste de la frecuencia */
#define ADJ_MAXERROR    0x0004 /* error máximo de la hora */
#define ADJ_ESTERROR    0x0008 /* error estimado de la hora */
#define ADJ_STATUS      0x0010 /* estado del reloj */
#define ADJ_TIMECONST   0x0020 /* pll constante de tiempo */
#define ADJ_TICK        0x4000 /* valor tick */
#define ADJ_OFFSET_SINGLESHOT 0x8001 /* ajuste de hora; método antiguo */
```

Los usuarios comunes sólo puede utilizar un valor cero para mode. Únicamente el superusuario puede asignar un valor a un parámetro.

VALOR DEVUELTO

En caso de éxito, adjtimex devuelve el estado del reloj:

```
#define TIME_OK      0 /* reloj sincronizado */
#define TIME_INS     1 /* insertar segundo de salto */
#define TIME_DEL     2 /* borrar segundo de salto */
#define TIME_OOP     3 /* segundo de salto en progreso */
#define TIME_WAIT    4 /* se ha producido un segundo de salto */
#define TIME_BAD     5 /* reloj no sincronizado */
```

En caso de fallo, adjtimex regresa -1 y asigna un valor apropiado a errno.

ERRORES

EFAULT buf no apunta a una zona de memoria escribible.

EPERM buf.mode no es cero y el usuario no es el superusuario.

EINVAL Se ha hecho un intento de asignar a buf.offset un valor fuera del intervalo -131071 a +131071, o de asignar a buf.status un valor no contenido en la lista anteriormente discutida, o de asignar a buf.tick un valor fuera del intervalo 900000/HZ a 1100000/HZ, donde HZ es la frecuencia de interrupción del reloj del sistema.

VÉASE TAMBIÉN

settimeofday(2).

CLOCKDIFF(8) Iputils-ss990107 CLOCKDIFF(8)

NAME

clockdiff - Measures clock difference between us and <destination> with lmsc resolution. Without -o option it uses icmp timestamps, with -o it uses icmp echo with timestamp IP option.

SYNOPSIS

clockdiff [-o|-o1] <destination>

WARNINGS

- * some nodes (f.e. Solaris<2.4) do not support icmp timestamps. clockdiff -o still works.
- * some nodes (Cisco) use non-standard timestamps, which makes clockdiff useless.
- * some nodes use wrong timestamps (Solaris>2.4), if run xntpd. Seems, it uses different clock source, which is synchronized to time-of-day clock periodically.
- * -o1 uses three-node tstamp. What option (-o or -o1) works better depends on target host.

HWCLOCK(8) HWCLOCK(8)

NOMBRE

hwclock - interroga y pone en hora el Reloj del Hardware (RTC)

SINOPSIS

```
hwclock --show
hwclock --set --date=nuevafecha
hwclock --systohc
hwclock --hctosys
hwclock --getepoch
hwclock --setepoch --epoch=año
hwclock --adjust
hwclock --version
```

otras opciones:

```
--utc --directisa --test --debug
```

Las abreviaciones mínimas de todas las opciones son aceptables, si no hay ambigüedad.

También se aceptan las opciones equivalentes -r, -w, -s, -a, -v, -u y -D, por compatibilidad con el programa "clock".

DESCRIPCIÓN

hwclock es una herramienta para acceder al Reloj del Hardware. Ud. puede mostrar la hora actual, poner el Reloj del Hardware a una hora especificada, poner el Reloj del Hardware a la Hora del Sistema, y poner el Tiempo del Sistema desde el Reloj del Hardware.

Ud. también puede ejecutar hwclock periódicamente para insertar o quitar tiempo del Reloj del Hardware para compensar desviaciones sistemáticas (en las que el reloj gana o pierde tiempo consistentemente a una cierta velocidad si se deja solo).

OPCIONES

Ud. necesita exactamente una de las siguientes opciones para decirle a hwclock qué función realizar:

--show Lee el Reloj del Hardware y muestra la hora en la salida estándar.

--set Pone el Reloj del Hardware a la hora dada por la opción --date.

--hctosys Pone el Tiempo del Sistema a partir del Reloj del Hardware. Ésta es una buena opción para poner en uno de los guiones de arranque del sistema.

--systohc Pone el Reloj del Hardware a la hora del sistema actual.

--adjust Añade o sustrae tiempo del Reloj del Hardware para tener en cuenta el desvío sistemático desde la última vez que el reloj se puso o se ajustó. Vea la discusión al respecto más adelante.

--getepoch Muestra en la salida estándar el valor de la época del Reloj del Hardware del núcleo. Esto es el

número de años en la Era al que se refiere el año cero del Reloj del Hardware. Por ejemplo, si Ud. emplea el convenio de que el contador de años en su Reloj del Hardware contiene el número de años completos desde 1952, entonces el valor de la época del contador de años del Reloj del Hardware debe ser 1952.

Este valor de la época se emplea cada vez que hwclock lee o pone el Reloj del Hardware.

--setepoch

Pone el valor de la época del Reloj del Hardware del núcleo al valor especificado por la opción --epoch. Vea la opción --getepoch para más detalles.

--version

Muestra la versión de hwclock en la salida estándar.

Si Ud. especifica la opción --set, necesita la opción siguiente. Si no, no es tenida en consideración.

--date=nuevafecha

Especifica la hora a la cual poner el Reloj del Hardware. El valor de esta opción es un argumento válido para el programa date(1). Por ejemplo,

```
hwclock --set --date="9/22/96 16:45:05"
```

--epoch=año

Especifica el año que es el principio de la época del Reloj del Hardware. Esto es el número de años en la Era a la cual se refiere un valor cero en el contador de años del Reloj del Hardware.

Por ejemplo,

```
hwclock --setepoch --epoch=1952
```

Las opciones siguientes se aplican a la mayoría de las funciones.

--utc Indica que el Reloj del Hardware se mantiene en el Tiempo Universal Coordinado (UTC). Es cosa suya si mantiene su reloj en hora local o UTC, pero nada en el reloj le dice qué es lo que ha escogido. Así que con esta opción es como le da esa información a hwclock.

Si no especifica --utc cuando debiera, o viceversa, el poner y preguntar la hora del reloj del hardware acabarán funcionando mal.

--directisa

tiene sentido solamente en una máquina ISA. Para todas las otras máquinas no tiene efecto. Esta opción le dice a hwclock que use instrucciones de E/S explícitas para acceder al reloj del hardware. Sin esta opción, hwclock intentará emplear el dispositivo /dev/rtc (que supone que está controlado por el controlador del dispositivo rtc). Si es incapaz de abrir el dispositivo (para lectura), empleará las instrucciones explícitas de E/S, de todas formas.

El controlador de dispositivo rtc fue introducido en Linux versión 2.

--test Hace todo excepto actualizar realmente el Reloj del Hardware o cualquier otra cosa. Esto es útil, especialmente en conjunción con --debug, para aprender sobre cómo funciona hwclock.

--debug

Muestra un montón de información sobre lo que hwclock está haciendo internamente. Algo de su función es compleja y esta salida puede ayudarle a comprender cómo funciona el programa.

NOTAS

Relojes en un sistema Linux

Hay dos relojes principales en un sistema Linux:

El Reloj del Hardware: Esto es un reloj que corre independientemente de cualquier programa de control ejecutándose en la CPU e incluso cuando la máquina está apagada.

En un sistema ISA, este reloj está especificado como parte del estándar ISA. El programa de control puede leer o poner este reloj a un segundo entero, pero el programa de control también puede detectar los límites de los pulsos del reloj de 1 segundo, así que el reloj realmente tiene una precisión virtualmente infinita.

Este reloj se llama comúnmente el Reloj del Hardware, el reloj de tiempo real, el RTC, el reloj de la BIOS, y el reloj CMOS. El Reloj del Hardware, en esta forma con las iniciales en mayúsculas, se acuñó para el uso en hwclock porque todos los otros nombres son inapropiados hasta el punto de llevar a confusión.

El Tiempo del Sistema: Esto es el tiempo mantenido por un reloj de dentro del núcleo Linux y controlado por un interruptor de temporización. (En una máquina ISA, el interruptor de temporización es parte del estándar ISA.) Sólo tiene sentido mientras Linux está ejecutándose en la máquina. El Tiempo del Sistema es el número de segundos desde las 00:00:00 horas del 1 de Enero de 1970, UTC (o más sucintamente, el número de segundos desde 1969). El Tiempo del Sistema no es un entero, sin embargo. Tiene precisión virtualmente infinita.

El Tiempo del Sistema es el que tiene importancia. El propósito básico del Reloj del Hardware en un sistema Linux es mantener la hora mientras que Linux no está ejecutándose. Ud. inicia el Tiempo del Sistema al tiempo del Reloj del Hardware cuando Linux arranca, y ya no utiliza nunca más el Reloj del Hardware. Observe que en DOS, para quien se diseñó ISA, el Reloj del Hardware es el único reloj de tiempo real.

Es importante que el Tiempo del Sistema no tenga discontinuidades como las que podrían ocurrir si Ud. empleara el programa date(1) para ponerlo en hora mientras el sistema esté en ejecución. Sin embargo, Ud. puede hacer lo que quiera con el Reloj del Hardware mientras el sistema esté ejecutándose, y la siguiente vez que Linux arrancara, lo haría con la nueva hora ajustada del Reloj del Hardware. También puede emplear el programa adjtimex(8) para ajustar suavemente el Tiempo del Sistema mientras que el sistema está ejecutándose.

Cómo hwclock accede al Reloj del Hardware

hwclock utiliza muchas formas diferentes de obtener y poner valores en el Reloj del Hardware. La forma más normal es hacer E/S en el fichero especial de dispositivo /dev/rtc, que se supone controlado por el controlador del dispositivo rtc. Sin embargo, este método no está siempre disponible. Por una cosa: el controlador rtc es una adición relativamente reciente a Linux. Sistemas más antiguos no lo tienen.

En sistemas más viejos, el método de acceder al Reloj del Hardware depende del hardware del sistema.

En un sistema ISA, hwclock puede acceder directamente a los registros de la "memoria CMOS" que constituye el reloj, haciendo E/S a los puertos 0x70 y 0x71. Esto sólo lo puede hacer si se ejecuta con el UID efectivo 0 (esto es, con privilegios de súper-usuario).

Esto es realmente un pobre método de acceder al reloj, por la razón de que los programas en el espacio de usuario se supone generalmente que no deben hacer E/S directa y deshabilitar interrupciones. Hwclock lo proporciona porque es el único método disponible con núcleos Linux más antiguos para máquinas ISA.

En un sistema m68k (68000 de Motorola), hwclock puede acceder al reloj a través del controlador de consola, mediante el fichero especial de dispositivo /dev/ttyl.

En un Alpha, de DEC, /dev/rtc es la única posibilidad.

Hay o había algunos sistemas Linux en Alpha que no tienen /dev/rtc y hay o había programas que accedían al reloj a través de E/S casi directa usando /dev/port. Sin embargo, esto no es un método tan bueno como el de /dev/rtc y tales programas no fueron tan ampliamente empleados como para que hwclock tuviera ninguna necesidad de ser compatible hacia atrás con ellos. Así que hwclock no proporciona el método de /dev/port y consecuentemente no funcionará en un Alpha que no tenga /dev/rtc.

hwclock intenta emplear /dev/rtc. Si se ha compilado para un núcleo que no tiene esa función o es incapaz de abrir /dev/rtc, hwclock intentará otro método, si hay alguno disponible. En una máquina ISA, Ud. puede forzar a hwclock a usar la manipulación directa de los registros de la CMOS sin ni siquiera intentar /dev/rtc especificando la opción --directisa.

La Función de Ajuste

El Reloj del Hardware no es usualmente muy preciso. Sin embargo, mucha de su imprecisión es completamente predecible; gana o pierde la misma cantidad de tiempo cada día. Esto se llama desviación sistemática. La función de "ajuste" de hwclock le permite hacer correcciones sistemáticas para corregir la desviación sistemática.

Funciona así: Hwclock mantiene un fichero, /etc/adjtime, que mantiene alguna información histórica. Este fichero se llama el fichero adjtime.

Suponga que Ud. empieza sin fichero adjtime. Ud. da una orden hwclock --set para poner el Reloj del Hardware a la hora verdadera actual. Hwclock crea el fichero adjtime y graba en él la hora actual como la última vez que el reloj se calibró. 5 días más tarde, el reloj ha ganado 10 segundos, así que Ud. da otra orden hwclock --set para ponerlo con 10 segundos menos. Hwclock ajusta el fichero

adjtime para mostrar la hora actual como la última vez que el reloj se calibró, y graba dos segundos por día como la velocidad de desviación sistemática. Pasan 24 horas, y entonces Ud. da una orden hwclock --adjust. Hwclock consulta el fichero adjtime y ve que el reloj gana 2 segundos por día cuando se le deja solo, y que se le ha dejado solo exactamente un día. Así que sustrae 2 segundos del Reloj del Hardware. Entonces graba la hora actual como la última vez que se ajustó el reloj. Otras 24 horas pasan y Ud. da otra orden hwclock --adjust. Hwclock hace lo mismo: sustrae 2 segundos y actualiza el fichero adjtime con la hora actual como la última vez que el reloj se ajustó.

Cada vez que Ud. calibra (pone en hora) el reloj, hwclock recalcula la velocidad de desviación sistemática basado en cuánto tiempo ha pasado desde la última calibración, cuánto tiempo ha pasado desde el último ajuste, qué velocidad de desviación se suponía en cualesquiera ajustes intervinientes, y la cantidad que el reloj está desviado actualmente de la hora verdadera.

Se arrastra una pequeña cantidad de error en cualquier momento en que hwclock pone el reloj en hora, de forma que rehúsa hacer un ajuste que vaya a ser de menos de 1 segundo. Más tarde, cuando Ud. pida un ajuste de nuevo, la desviación acumulada será de más de un segundo y hwclock hará el ajuste entonces.

Es bueno hacer un hwclock --adjust justo antes del hwclock --hctosys en la hora del arranque del sistema, y quizás periódicamente mientras que el sistema está ejecutándose, mediante cron.

El formato del fichero adjtime es:

Línea 1: 3 números: 1) velocidad de desviación sistemática en segundos por día, número en base diez en coma flotante; 2) número resultante de segundos desde 1969 UTC del más reciente ajuste o calibración, entero en base diez; 3) cero (por compatibilidad con clock).

Línea 2: 1 número: El número resultante de segundos desde 1969 UTC de la calibración más reciente.

Ud. puede emplear con hwclock un fichero adjtime que fuera usado previamente con el programa clock.

FICHEROS

/etc/adjtime

VÉASE TAMBIÉN

adjtimex(8), date(1), gettimeofday(2), settimeofday(2), crontab(1)

AUTORES

Escrito por Bryan Henderson, en Septiembre de 1996 (bryanh@giraffe-data.com), basándose en el trabajo hecho en el programa clock de Charles Hedrick, Rob Hooft y Harald König. Mire el código fuente para la historia completa y los créditos.

SETCLOCK(8) SETCLOCK(8)

NAME
setclock - sets the hardware clock from the system clock

SYNOPSIS
setclock

DESCRIPTION
setclock sets the hardware clock on the current system to the current time stored in the system clock. /etc/sysconfig/clock is read to discover the format the time should be converted to before storage. This is the same method used by the standard Red Hat startup scripts and timeconfig(8).

FILES
/etc/sysconfig/clock
Configures the format used by the hardware clock.

SEE ALSO
timeconfig(8)

AUTHOR
Erik Troan <ewt@redhat.com>

TIMECONFIG(8) TIMECONFIG(8)

NAME
timeconfig - simple interface for configuring system time parameters

SYNOPSIS
timeconfig [--utc] [--arc] [--back] [--test] [timezone]

DESCRIPTION
timeconfig provides a simple method of configuring both /etc/sysconfig/clock and /etc/localtime which is compatible with both the standard Red Hat startup scripts and timeconfig(8).

If a timezone is specified, the system timezone is change to the timezone specified. Otherwise, a menu of available timezones is displayed. If --utc is given, the system clock is assumed to be in universal (Greenwich mean) time. If --arc is used, the system clock is assumed ot be stored in the format used by the ARC platform on Digital Alpha platforms (the --arc option is only available on Alpha hardware).

The timeconfig window contains a Cancel button by default. If --back is specified at run time, a Back button is presented instead.

If --test is specified, timeconfig may be run as a non-root user, and no changes are written to the filesystem.

RETURN CODES
timeconfig returns 0 on success, 2 on error, and 1 if the user cancelled the program (by using either the Cancel or Back button).

FILES
/etc/sysconfig/clock
Configures the format used by the hardware clock.

/etc/localtime
File containing timezone data, copied from /usr/share/zoneinfo.

SEE ALSO
setclock(8)

AUTHOR
Erik Troan <ewt@redhat.com>