The Clock Mini-HOWTO

Ron Bean, *rbean@execpc.com* v2.1, Novembre 2000

Come regolare e mantenere preciso l'orologio del tuo computer.

Indice

1 Introduzione

1.1 Qualcuno sa veramente che ora è?

I chip Real-Time-Clock (RTC, orologio in tempo reale) usati sulle schede madri dei PC sono notoriamente inaccurati, visto che di solito guadagnano o perdono una quantità considerevole (e di solito costante) di tempo ogni giorno. Linux fornisce un sistema semplice per correggere questo via software, il che può rendere l'orologio **molto** accurato, anche senza ricorrere a una fonte d'orario esterna. La maggior parte delle persone però non sa come impostarlo, per diversi motivi:

- Non è menzionato nella maggior parte della documentazione generale su come impostare linux, e non può essere impostato automaticamente (se non si ha una fonte esterna per l'orario), per cui il default è di non usarlo.
- Se si digita `"man clock"` si ottiene la pagina man per `clock(3)`, che non è quello che si vuole. Prova `"man 8 clock"` o `"man 8 hwclock"` (alcune distribuzioni cercano le pagine man in ordine numerico se non gli specifichi un numero di sezione, altre cercano nell'ordine impostato in `/etc/man.config`).
- La maggior parte della gente non pare comunque curarsi di che ora sia.
- Quei pochi che se ne curano spesso preferiscono sincronizzare l'orologio con una fonte esterna, come un server di orario di rete o un orologio radio. Questo rende (praticamente) irrilevante l'accuratezza del RTC.

Questo mini-HOWTO descrive l'approccio low-tech (che può essere ugualmente molto accurato), e fornisce indicazioni a molte alternative più sofisticate. Nella maggioranza dei casi la documentazione è scritta molto bene, quindi non ripeterò quelle informazioni qui.

Versioni precedenti includevano le istruzioni dettagliate per il vecchio programma `clock(8)` per chi avesse ancora un sistema vecchiotto, ma ho scartato quella sezione perché la maggior parte delle distribuzioni ora usa al suo posto `hwclock(8)`, che ha una documentazione molto migliore. Se vuoi comunque una copia delle istruzioni di `clock(8)`, te le posso mandare per posta elettronica, ma prima leggi la sezione su `hwclock(8)`.

Nota

Devi essere utente **`"root"`** per avviare qualsiasi programma che influenzi l'RTC o l'orario di sistema. Tienilo a mente quando utilizzi i programmi descritti qui. Se normalmente usi un'interfaccia grafica per fare tutto, potresti dover imparare alcuni comandi di shell basilari di Unix.

Nota

Se usi più di un sistema operativo sul tuo PC, dovresti permettere ad uno solo di essi di regolare l'orologio CMOS, perché non si confondano l'un l'altro. L'eccezione è costituita dalla regolazione per l'ora legale, che avviene due volte l'anno (vedi la sezione appropriata per maggiori dettagli).

Se usi un sistema dual-boot che avvia Windows per molto tempo, forse preferisci dare un'occhiata a qualche software disponibile per quel sistema operativo. Segui i collegamenti dal sito web dell'NTP su `http://www.eecis.udel.edu/~ntp/software.html` . Molti degli orologi radio nominati qui includono software per Windows.

Nota

Per la traduzione italiana Germano Rizzo, *mano78@users.sourceforge.net*

1.2 Dove trovare roba: "i soliti posti"

Da qualche parte ho detto che il software può essere scaricato da `"i soliti posti"`, che significa quei siti dai quali puoi scaricare un sistema Linux completo se non l'hai avuto su CD-ROM. Un po' di tempo fa questo voleva dire l'archivio ftp di sunsite.unc.edu, e vari mirror nel mondo. Questo sito è stato rinominato `http://metalab.unc.edu/linux/` (poiché Sun non lo sponsorizza più). Alcune distribuzioni hanno anche un loro sito web, che può includere un bel po' di queste cose.

Darò per scontato che la maggior parte delle persone adesso abbiano Linux su CD, e quest CD spesso includono software che non è parte dell'installazione di default, così potresti già avere alcuni dei programmi nominati qui senza saperlo.

La versione più recente di questo mini-HOWTO può essere reperita alla home page del Linux Documentation Project, che è attualmente `http://www.linuxdoc.org/` (e può essere raggiunto anche dal sito di metalab nominato sopra). Penso che tutti i vecchi link conducano ora a questo. La traduzione italiana più recente può essere ritrovata ugualmente lì, o alla home page dell'Italian Linux Documentation Project, presso `http://www.pluto.linux.it/ildp/` .

Tutti gli HOWTO sono scritti in SGML e poi convertiti in vari altri formati da programmi di conversione standard. Molti sembrano preferire la versione HTML, che si trova a `http://www.linuxdoc.org/HOWTO/mini/Clock.html` (per la versione italiana, `http://www.pluto.linux.it/ildp/HOWTO/mini/Clock.html`). L'history delle revisioni si trova come commento nel sorgente SGML. Molte distribuzioni installano un insieme completo di HOWTO in `/usr/doc/HOWTO/` e `/usr/doc/HOWTO/mini`.

1.3 Ringraziamenti

Questo mini-HOWTO è stato migliorato di molto grazie a varie persone che mi hanno scritto fin dalla prima versione, nel 1996. In alcuni casi mi hanno scritto ponendomi domande, ma hanno finito per darmi tante informazioni quante ne davo io a loro. Sfortunatamente non ho compilato una lista completa dei nomi (forse la prossima volta). Ma so chi siete `:-)`.

2 Come Linux tiene traccia del tempo

2.1 Strategie base

Un sistema Linux ha effettivamente due orologi: uno è il "Real Time Clock", alimentato a batteria (conosciuto anche come "RTC", "orologio CMOS", o "orologio Hardware") che scandisce il tempo quando il sistema è spento ma non è utilizzato quando è funzionante. L'altro è l'"orologio di sistema" (chiamato anche "orologio del kernel" o "orologio software") che è un contatore software basato sugli interrupt del timer. Non esiste quando il sistema non è in funzione, per cui deve essere inizializzato dall'RTC (o da qualche altra fonte) al boot. I riferimenti all'"orologio" nella documentazione di `ntpd` si riferiscono a quello di sistema, non all'RTC.

I due orologi deviano dall'ora reale a ritmi differenti, perciò gradualmente i rispettivi orari differiranno sempre più tra loro, ed anche dal tempo "reale". Il modo più semplice per mantenerli precisi è di misurare i tassi di deviazione e applicare dei fattori di correzione nel software. Poichè l'RTC è usato solo quando il sistema non è attivo, il fattore di correzione è applicato quando l'orario viene letto all'avvio, usando `clock(8)` o `hwclock(8)`. L'orologio di sistema è corretto regolando la quantità di tempo di cui l'orologio di sistema è portato avanti ad ogni interrupt del timer, usando `adjtimex(8)`.

Una rude alternativa ad `adjtimex(8)` è far avviare `clock(8)` o `hwclock(8)` a `chron` periodicamente per sincronizzare l'orario di sistema all'RTC (corretto). Questa strada era raccomandata nella pagina man `clock(8)` e funziona se lo si fa abbastanza spesso da non causare grossi "salti" nell'orario di sistema, ma `adjtimex(8)` èuna soluzione più elegante. Alcune applicazioni potrebbero inoltre lamentarsi se l'orario "salta" all'indietro.

Il prossimo passo in accuratezza è utilizzare un programma come `ntpd` per leggere periodicamente l'orario da un server di orario di rete o da un'orologio radio, ed aggiustare continuamente il passo dell'orologio di sistema cosicché i tempi corrispondano sempre. Se hai sempre una connessione ad una rete attiva all'avvio del tuo sistema, puoi ignorare completamente l'RTC e usare `ntpdate` (incluso nel pacchetto `ntpd`) per inizializzare l'orologio di sistema da un server – o un server locale su LAN, o un server remoto su Internet. Se qualche volta però non hai una connessione di rete, o se ti serve che l'ora sia precisa durante la sequenza di avvio e prima della connessione alla rete, allora dovrai mantenere preciso anche l'RTC.

2.2 Potenziali Conflitti

Potrebbe sembrare ovvio che se usi un programma come `ntpd`, potresti sincronizzare l'RTC all'orologio di sistema (corretto). Questa risulta però essere una cattiva idea se il sistema starà spento per più di pochi minuti, perché questa procedura interferirà coi programmi che applicano il fattore di correzione all'RTC all'avvio.

Se il sistema è attivo 24 ore su 24 e 7 giorni su 7 ed è sempre riavviato immediatamente dopo essere stato eventualmente spento, puoi semplicemente regolare l'RTC dall'orologio di sistema giusto prima di spegnere la macchina. L'RTC non devierà significativamente nel tempo speso per il riavvio, perciò non ti servirà conoscere il suo tasso di deviazione.

Naturalmente il sistema potrebbe bloccarsi imprevedibilmente, così alcune versioni del kernel sincronizzano l'RTC all'orologio di sistema ogni 11 minuti se quest'ultimo è stato riaggiustato da un altro programma. L'RTC non devierà abbastanza in 11 minuti da fare alcuna differenza, ma se il sistema è spento abbastanza a lungo da far deviare significativamente l'RTC, allora hai un problema: i programmi che applicano le correzioni all'RTC hanno bisogno di sapere **esattamente** quando l'RTC è stato resettato l'ultima volta, ed il kernel non registra questa informazione da nessuna parte.

Alcuni "tradizionalisti" di unix potrebbero chiedersi perché qualcuno potrebbe lasciare attivo un sistema linux per meno di 24 ore al giorno e sette giorni alla settimana, ma alcuni di noi utilizzano un sistema dual-boot, con un altro OS in funzione per un certo tempo, o usano Linux su portatili che devono essere spenti per preservare le batterie quando non sono utilizzati. Altre persone, semplicemente, non vogliono lasciare i PC attivi senza sorveglianza per lunghi periodi di tempo (anche se abbiamo sentito ogni sorta di argomentazioni in favore di questa linea). Perciò l'"ogni 11 minuti" diventa un bug.

Questa "caratteristica/bug" sembra comportarsi in maniera differente in diverse versioni del kernel (e forse anche in diverse versioni di `xntpd` e `ntpd`), così se stai usando sia `ntpd` che `hwclock`, potrebbe essere necessario controllare cosa il tuo sistema realmente fa. Se non puoi trattenere il kernel dal resettare l'RTC, potresti dover far funzionare il sistema senza un fattore di correzione.

La parte del kernel che regola questo aspetto si trova in `/usr/src/linux-2.0.34/arch/i386/kernel/time.c` (dove il numero di versione nel path sarà dato dalla versione del kernel che stai utilizzando). Se la variabile `time_status` è impostata a `TIME_OK` allora il kernel scriverà l'ora sull'RTC ogni 11 minuti, altrimenti lo lascerà stare. Le chiamate a `adjtimex(2)` (come quelle che vengono effettuate da `ntpd` e `timed`, ad esempio) potrebbero abilitare questo parametro. Le chiamate a `settimeofday(2)` imposteranno `time_status` a `TIME_UNSYNC`, che dice al kernel di non regolare l'RTC. Non ho trovato vera documentazione su questo aspetto.

Ho sentito che alcune versioni del kernel potrebbero aver problemi con gli "sleep modes" che spengono la CPU per risparmiare energia. La soluzione migliore è di tenere il kernel aggiornato, e riportare ogni problema alle persone che lo aggiornano.

Se ottieni risultati bizzarri dall'RTC potresti avere un problema hardware. Alcuni chip RTC includono una batteria al litio che può esaurirsi, e alcune schede madri hanno la possibilità di inserire una batteria esterna (assicurati che i jumper siano impostati correttamente). La stessa batteria mantiene la RAM CMOS, ma l'orologio assorbe più energia ed è probabile che abbia problemi prima. Se le cose strane riguardano l'orologio di sistema, potrebbe esserci un problema con gli interrupt.

2.3 L'RTC deve usare il local time o l'UTC? Cosa mi dici dell'ora legale?

l'"orologio di sistema" di Linux effettivamente conta solo il numero di secondi dopo il 1 Gennaio 1970, ed è sempre in UTC (Coordinated Universal Time - Orario Universale Coordinato; o GMT, Greenwich Mean Time - Ora Media di Greenwich, che è tecnicamente differente ma abbastanza vicino da far usare indifferentemente i due termini dagli utenti normali). L'UTC non cambia al cambiare dell'ora legale– quello che cambia è la *conversione* tra UTC e local time (orario locale). La traduzione in local time è operata da funzioni di libreria che vengono linkate nelle varie applicazioni.

Questo ha due conseguenze: primo, ogni applicazione che necessiti di conoscere il local time avrà anche bisogno di conoscere in che fuso orario ci si trovi, e se vige l'ora legale o meno (vedi la sezione successiva per maggiori dettagli sui fusi orari). Secondo, non c'è nessuna funzionalità nel kernel per cambiare l'orologio di sistema o l'RTC al variare dell'ora legale, perché l'UTC non cambia. Perciò i sistemi che funzionano solo con Linux dovrebbero avere l'RTC impostato sull'UTC, non sul local time.

Comunque, molte persone utilizzano sistemi dual-boot con altri OS, che si aspettano che l'RTC contenga il local time; perciò, `hwclock` ha bisogno di sapere se il tuo RTC è sul local time o sull'UTC, e poi lo converte nei secondi trascorsi dal 1 Gennaio 1970 (UTC). Questo ancora non fornisce funzionalità per i cambiamenti periodici dell'RTC, perciò il cambiamento dev'essere operato dagli altri OS (questa è l'eccezione che si diceva prima alla regola di lasciar che un solo programma cambi l'orario dell'RTC).

Sfortunatamente, non ci sono flag nell'RTC o nella RAM CMOS che indichino che si sta utilizzando l'ora legale piuttosto di quella solare, perciò ogni OS registra quest'informazionein luoghi dove non è visibile agli altri OS. Questo significa che `hwclock` deve presumere che l'RTC contenga sempre il local time corretto, anche se l'altro OS non è stato mai avviato dall'ultimo cambio d'orario stagionale.

Se Linux è in funzione quando succede il cambio d'orario stagionale, l'orologio di sistema non ne è affetto, e le applicazioni effettueranno la conversione corretta. Ma se Linux deve per qualche ragione essere riavviato, l'orologio di sistema sarà impostato all'orario registrato nell'RTC, che sarà di un'ora sbagliato, finché l'altro OS (di solito Windows) avrà la possibilità di essere avviato.

Non c'è soluzione a questo problema, ma Linux non va in crash troppo spesso, perciò la ragione più probabile per un reboot su un sistema dual-boot è proprio che si debba avviare l'altro OS. Ma attenzione se sei una di quelle persone che chiudonoLinux quando non lo utilizzi per un po'– se non hai avuto la possibilità di avviare l'altro OS dall'ultima variazione di orario, l'RTC sarà spostato di un ora finché non lo farai.

Alcuni altri documenti hanno affermato che impostare l'RTC su UTC permette a Linux di gestire correttamente l'ora legale. Questo non è proprio sbagliato, ma non la dice tutta– finché non riavvii, non importa che orario c'è nell'RTC (o adirittura se la sua batteria si esaurisce). Linux manterrà l'orario corretto in ogni caso, fino al prossimo riavvio. In teoria, se riavvii una volta all'anno (il che non è irragionevole per Linux), l'ora legale potrebbe cambiare due volte e se l'RTC fosse rotto anche per molti mesi non lo noteresti neppure, perché l'orologio di sistema sarebbe rimasto corretto per tutto il tempo. Ma dal momento che non puoi prevedere quando riavvierai, è meglio avere l'RTC impostato sull'UTC se non usi un'altro OS che richieda il local time.

Il chip RTC della Dallas Semiconductors (che è un sostituto per il chip della Motorola utilizzato negli IBM AT e cloni) può effettivamente effettuare da sè il passaggio all'ora legale e viceversa, ma questa caratteristica non è utilizzata perché le date di passaggio sono codificate dentro l'hardware stesso del chip e non possono essere cambiate. Le versioni correnti effettuano il cambiamento alla prima domenica di Aprile e all'ultima di Ottobre, ma versioni precedenti usavano date differenti (e ovviamente questo non funziona in altri stati che negli USA). L'RTC è inoltre spesso integrato nel "chipset" della scheda madre (piuttosto che essere un chip separato), e di questi chip non so quali abbiano tale carateristica.

2.4 Come Linux tiene conto dei fusi orari

Probabilmente hai impostato il tuo fuso orario correttamente quando hai installato Linux. Ma se devi cambiarlo per qualche ragione, o se le leggi locali riguardanti l'ora legale sono cambiate (come fanno di frequente in alcuni stati), allora dovrai sapere come cambiarlo. Se il tuo orario di sistema è sbagliato di un numero esatto di ore rispetto all'ora reale, potresti avere un problema di fuso orario (o di impostazione dell'ora legale).

Il fuso orario e le informazioni sull'ora legale sono registrate in `/usr/share/zoneinfo` (o `/usr/lib/zoneinfo` nei sistemi più datati). Il fuso orario locale è determinato da un link simbolico da `/etc/localtime` su uno di questi file. Il modo di cambiare il fuso è cambiare il link. Se le date per l'ora legale del tuo stato sono cambiate, dovrai modificare direttamente il file.

Puoi usare anche la variabile d'ambiente `TZ` per cambiare il fuso corrente, il che è pratico se lavori accedendo in remoto a una macchina che si trova in un'altra zona. Vedi anche la pagina man di `tzset` e `tzfile`.

Tutto ciò è ottimamente riassunto a `http://www.linuxsa.org.au/tips/time.html`

2.5 In definitiva

Se non ti serve un'accuratezza a meno di un secondo, `hwclock(8)` e `adjtimex(8)` potrebbero essere tutto ciò che ti serve. È facile essere presi dagli orologi radio e simili, ma io ho usato il programma `clock(8)` per anni con eccellenti risultati. D'altro canto, se hai molte macchine su una LAN potrebbe risultare comodo far loro sincronizzare gli orologi automaticamente l'una con l'altra. E potrebbe essere divertente giocherellare con il resto della roba anche se non ne hai propriamente bisogno.

Su macchine che utilizzano solo Linux, imposta l'RTC su UTC (o GMT). Su sistemi dual-boot che richiedono che nell'RTC ci sia il local time, sappi che se devi riavviare Linux dopo il cambio di orario stagionale, l'orologio potrebbe essere sfasato di un'ora, finché non avrai la possibilità di avviare l'altro OS. Se hai più di due OS, assicurati che uno solo di essi operi l'impostazione dell'ora legale.

3 Software

3.1 Clock(8) e Hwclock(8)

Tutte le distribuzioni linux installano o il vecchio `clock(8)` o il più recente `hwclock(8)`, ma senza un fattore di correzione. Alcune potrebbero anche installare `adjtimex(8)`, o includerlo nel CD come optional (puoi comunque scaricarlo dai soliti archivi). Alcune distribuzioni includono anche un programma grafico di regolazione dell'orologio che funziona in X-window, ma questi programmi sono concepiti per un uso interattivo, ed il sistema installerà comunque `clock(8)` o `hwclock(8)` per l'uso negli script di avvio.

`Clock(8)` ti richiede di calcolare il fattore di correzione a mano, mentre `hwclock(8)` lo calcola automaticamente quandunque lo userai per resettare l'orologio (usare un altro programma per impostare l'orario interferirà con la correzione della deriva, perciò usa sempre lo stesso programma se stai utilizzando il fattore di correzione). Se hai un sistema più datato che usa ancora `clock(8)` e vuoi aggiornarlo, puoi trovare `hwclock(8)` nel pacchetto `"util-linux"`, versione 2.7 o successiva. Consulta la pagina man per maggiori informazioni.

Nota

> La pagina man di `hwclock(8)` può esser chiamata `"clock"` per compatibilità all'indietro, perciò prova entrambi i nomi. `Hwclock(8)` risponderà ai comandi scritti per `clock(8)`, ma il risultato potrebbe non essere lo stesso– in particolare, `"hwclock -a"` non è proprio identico a `"clock -a"`, perciò se stai

> aggiornando a `hwclock` ti suggerirei di sostituire tutti i riferimenti a `"clock"` nei tuoi script di avvio in modo che usino i comandi propri di `hwclock`.

Gli script di avvio variano da una distribuzione all'altra, perciò potrai dover cercare un po' per trovare dove impostano l'orologio. Le posizioni tipiche sono `/etc/rc.local`, `/etc/rc.d/rc.sysinit`, `/etc/rc.d/boot`, o simili.

Il fattore di correzione per l'RTC è archiviato in `/etc/adjtime`. RedHat ha uno script in `/etc/sysconfig/clock` che controlla le opzioni di `hwclock`.

Quando stai regolando l'orologio per determinare il tasso di deviazione, tieni a mente che l'annuncio telefonico dell'orario locale può essere accurato come no. Se non hai una radio ad onde corte o un ricevitore GPS, puoi ascoltare il segnale audio della WWV chiamando lo (01)(303)499-7111 (è una chiamata a pagamento a Boulder, Colorado). Il servizio ti sconnetterà dopo tre minuti, ma dovrebbe essere abbastanza per regolare l'orologio. Anche l'USNO e il CHU del Canada hanno servizi telefonici, ma io preferisco quello del WWV perché passa più tempo tra l'annuncio ed il "beep". Puoi anche ottenere l'orario da un server d'orario di rete usando il programma `ntpdate` fornito con `ntpd`, e c'è un orologio Java a `www.time.gov` .

In ogni caso, quello che stai regolando è l'orologio di sistema, non l'RTC (vedi la pagina man del comando `date` per i formati da utilizzare. Usa poi `hwclock` per impostare l'RTC e calcolare il tasso di deviazione. Se stai facendo tutto a mano, dovresti essere in grado di impostarlo secondo una precisione di un secondo o due, e ottenere una approssimazione ragionevole del tasso di deviazione dopo un po' di settimane. Poi puoi avviare `adjtimex(8)` per impostare con precisione l'orologio di sistema.

3.2 Adjtimex(8)

`Adjtimex(8)` permette all'utente di regolare le variabili dell'orario del kernel, e perciò cambiare la velocità dell'orologio di sistema (devi avere l'accesso come **"root"** al sistema per farlo). È progettato in modo intelligente, affinché compari l'orologio di sistema all'RTC usando lo stesso fattore di correzione utilizzato da `clock(8)` o `hwclock(8)`, come registrato in `/etc/adjtime`. Perciò, una volta che hai stabilito il tasso di deriva dell'RTC, è abbastanza semplice correggere anche l'orologio di sistema. Quando sarai riuscito a farlo funzionare alla giusta velocità, potrai aggiungere una linea agli script d'avvio per regolare le variabili corrette del kernel al momento dell'avvio. Poiché `adjtimex(8)` è stato progettato per lavorare con `clock(8)` o `hwclock(8)`, include una soluzione per il bug dell'"ogni 11 minuti".

Dopo che hai installato `adjtimex(8)` puoi aver maggiori informazioni su come regolarlo digitando `"man 8 adjtimex"` (c'è anche una pagina man `adjtimex(2)`, ma non è ciò che serve) e leggendo il file `README` reperibile a `/usr/doc/adjtimex-1.3/README` (dove il numero di versione nel percorso sarà la versione corrente di `adjtimex(8)`).

3.3 Xntpd and ntpd: il Network Time Protocol

`Xntpd` (NTPv3) è stato sostituito da `ntpd` (NTPv4); la versione più vecchia non viene più aggiornata.

`Ntpd` è il programma standard per sicronizzare orologi in una rete, ed è corredato di una lista di server di orario pubblici a cui puoi connetterti. Può essere un po' più complicato da configurare degli altri programmi descritti qui, ma se sei interessato a questo tipo di cose, ti raccomando vivamente di dargli un'occhiata comunque.

La "casa base" per le informazioni su `ntpd` è il sito web dell'NTP a `http://www.eecis.udel.edu/~ntp/` che include pure collegamenti a tutti gli altri generi di cose riguardanti il tempo (compreso software per altri OS). Alcune distribuzioni Linux includono `ntpd` nei CD. C'è una lista di server d'orario pubblici su `http://www.eecis.udel.edu/~mills/ntp/clock2.html` .

Una caratteristica relativamente nuova in `ntpd` è un "burst mode", progettato per macchine che hanno solo un accesso intermittente ad Internet.

`Ntpd` include driver per abbastanza pochi orologi radio (e pare che alcuni siano meglio supportati di altri). La maggior parte degli apparecchi di questo tipo sono progettati per l'utilizzo commerciale e costano migliaia di dollari, ma ci sono alcune alternative più economiche (esaminate in successive sezioni). Nel passato la maggior parte erano ricevitori WWV o WWVB, ma ora il tipo più comune sembrano essere i GPS. NIST mantiene un file PDF sul proprio server che elenca i fabbricanti di orologi radio, a `http://www.boulder.nist.gov/timefreq/links.htm` (vicino alla fine della pagina). Anche il sito web di NTP include molti collegamenti ai produttori di orologi radio, a `http://www.eecis.udel.edu/~ntp/hardware.htm` e `http://www.eecis.udel.edu/~mills/ntp/refclock.htm` . Queste liste possono essere aggiornate come non esserlo `:-)`. La lista dei driver per `ntpd` è a `http://www.eecis.udel.edu/~ntp/ntp_spool/html/refclock.htm` .

`Ntpd` include anche driver per molti servizi di orario telefonici. Sono tutte chiamate a lunga distanza, perciò assicurati di calcolarne l'effetto sulla tua bolletta prima di usarli.

3.4 Chrony

`Xntpd` è stato scritto originariamente per sistemi provvisti di una connessione a tempo pieno ad un server d'orario di rete o a un orologio radio. In teoria può essere anche usato con macchine che sono connesse solo a tratti, ma Richard Curnow non è mai riuscito a farlo funzionare come voleva... perciò ha scritto `"chrony"`, perché fosse un'alternativa per quelli di noi che hanno accesso in rete solo quando sono collegati ad un provider (il "burst mode" di `ntpd` è stato progettato appunto per risolvere questo stesso problema). La versione corrente di `chrony` comprende la correzione della deriva per l'RTC, per macchine che rimangono spente per un lungo periodo di tempo.

Puoi avere maggiori informazioni dal sito web di Richard Curnow a `http://www.rrbcurnow.freeuk.com/chrony` o `http://go.to/chrony` . Ci sono anche due mailing list di `chrony`, una per annunci e una per le discussioni degli utenti. Per informazioni manda un'e-mail a *chrony-users-subscribe@egroups.com* o a *chrony-announce-subscribe@egroups.com* .

Il programma è distribuito solo come sorgente, ma la Debian ha incluso una distribuzione binaria nel suo archivio "non stabile". Il file sorgente è disponibile anche negli archivi linux usuali.

3.5 Clockspeed

Un'altra possibilità è il programma `clockspeed` di DJ Bernstein. Prende l'orario da un server e semplicemente reimposta l'orario di sistema ogni 3 secondi. Può anche essere usato per sincronizzare molte macchine su una LAN.

Qualche volta ho avuto problemi a raggiungere il suo sito web su `http://Cr.yp.to/clockspeed.html` , perciò se ti da un errore DNS prova di nuovo un'altro giorno. Proverò ad aggiornare questa sezione se avrò informazioni migliori.

4 Radio Clocks

4.1 CHU ed il "Gadget Box"

CHU, l'emittente d'orario ad onde corte situato presso Ottawa, è simile al WWV negli USA, ma con un'importante differenza: in aggiunta ad annunciare l'ora sia in inglese che in francese, trasmette pure l'orario corrente una volta al minuto usando i vecchi toni da modem "Bell 103" (300 baud). Questi toni sono molto facili da decodificare, e Bill Rossi si è reso conto che non ti serve neppure un modem: tutto ciò che serve è una radio ad onde corte ed una scheda audio. Se puoi ricevere il segnale dal CHU, questo potrebbe risultare il più economico orologio radio disponibile. La ricezione a onde corte varia nell'arco della giornata, ma Bill afferma che cambiando le frequenze due volte al giorno (mattino e sera) ottiene una copertura per quasi tutte le 24 ore. CHU trasmette su 3.33, 7.335 e 14.670 MHz.CHU, the Canadian shortwave time station near Ottawa, is similar

Per maggiori informazioni vedi il sito web di Bill Rossi a `http://www.rossi.com/chu/` . Il file sorgente è disponibile anche presso i soliti archivi Linux. Per informazioni sui servizi d'orario della CHU vedi `http://www.nrc.ca/inms/time/ctse.html` .

Il sito web di NTP ha un progetto di un "gadget box" che decodifica il segnale orario CHU usando un economico chip di un modem da 300 baud e una qualsiasi radio ad onde corte, a `http://www.eecis.udel.edu/~ntp/ntp_spool/html/gadget.htm` . I piani includono un'immagine Postscript di un circuito stampato, a due facce, ma devi realizzartela da solo (o trovare qualcuno che lo faccia per te).

`Ntpd` include un driver (type 7) per i ricevitori CHU, che funziona sia con modem come il `"gadget box"`, sia immettendo il sonoro direttamente nella presa del microfono di una SPARCstation Sun (o qualsiasi altra macchina con "driver audio compatibili").

4.2 WWV e l'"orologio più accurato"

Puoi aver sentito parlare del "Most Accurate Clock" (l'"orologio più accurato", appunto), della Heatkit, che riceveva e decodificava il segnale orario della WWV e aveva una porta seriale opzionale per essere connesso ad un computer. La Heatkit cessò di vendere questi kit molto tempo fa, ma continuarono a vendere la versione costruita in fabbrica fino al 1995, quando anche questa linea fu interrotta. Per i nostalgici dell'Heatkit (ma senza l'orologio), si veda `http://www.heathkit-museum.com` . La compagnia esiste ancora, e vende materiale educativo. Vedi `http://www.heathkit.com` .

Secondo Dave Mills, il brevetto della Heatkit sull'"orologio più accurato" dovrebbe scadere presto, perciò forse qualcuno lì fuori potrebbe voler replicarlo sotto forma di un IC a chip unico.

Il sito web dell'NTP ha un programma DSP (e un file PDF che lo descrive) disponibile a `http://www.eecis.udel.edu/~mills/resource.htm` che decodifica il segnale orario del WWV utilizzando una radio ad onde corte e il TAPR/AMSAT DSP-93, un kit DSP che non è più disponibile. Era basato sul chip TMS320C25 DSP chip della Texas Instruments. Il sito web del TAPR a `http://www.tapr.org` contiene molte informazioni sulla programmazione DSP casalinga.

`Ntpd` include un driver (type 6) per i codici d'orario IRIG-B e IRIG-E, che utilizza il `/dev/audio` di Sun SPARCstation, ma una nota dice che è "probabilmente portabile su altri sistemi". Il WWV usa il codice IRIG-H.

Il WWV è gestito dal NIST, che ha un sito web a `http://www.boulder.nist.gov/timefreq/index.html` . Questo sito comprende il testo della "Special Publication 432", che descrive i loro servizi di orario e frequenze, a `http://www.boulder.nist.gov/timefreq/pubs/sp432/sp432.htm` . Il WWV trasmette sui 2.5, 5, 10, 15, e 20 Mhz.

4.3 GPS e l'"orologio totalmente accurato"

Il segnale GPS include l'orario preciso, e alcuni ricevitori GPS hanno porte seriali. `Ntpd` comprende driver per molti ricevitori GPS. La caratteristica 1PPS ("One Pulse Per Second", una pulsazione per secondo, richiesta per un'elevata accuratezza) di solito richiede un'interfaccia separata per connetterli al computer.

TAPR ("Tuscon Amateur Packet Radio") fabbrica un kit per un'interfaccia chiamata "TAC-2" (sta per "Totally Accurate Clock", orologio totalmente accurato) che si collega a una porta seriale e funziona con qualsiasi ricevitore GPS che possa fornire un uscita 1PPS– compresi alcuni modelli "bare board" ("solo scheda"), che possono esser montati direttamente alla scheda dei circuiti. Per maggiori informazioni, vedi il loro sito a `http://www.tapr.org` . Il prezzo (a Giugno 1999) è attorno ai 140$, escluso il ricevitore GPS. Il kit non comprende alcun dispositivo di montaggio.

Il "gadget box" del CHU (descritto in un'altra sezione) può essere anche usato come interfaccia per il segnale 1PPS. Il sito dell'NTP contiene una discussione di questo argomento a `http://www.eecis.udel.edu/~ntp/ntp_spool/html/pps.htm` .

4.4 Segnali orari a bassa frequenza: DCF77, MSF(Rugby), WWVB

Queste stazioni a bassa frequenza trasmettono un codice orario semplicemente accendendo o spegnendo la linea portante. Ogni stazione usa il proprio schema di codifica, le cui specifiche sono disponibili sul sito dell'NTP a `http://www.eecis.udel.edu/~mills/ntp/index.htm` (vicino alla fine della pagina). DCF77 in Germania trasmette sui 77.5kHz. MSF in Inghilterra (chiamata anche "Rugby", che si riferisce apparentemente alla sua locazione) e WWVB in Colorado trasmettono entrambe sui 60 kHz.

La ricezione di WWVB varia, ma ci sono piani per aumentare il suo potere di trasmissione, in diversi passaggi. Puoi seguire il suo stato sul sito web di NIST a `http://www.boulder.nist.gov/timefreq/wwvstatus.html` .

Sembra che in Europa siano disponibili ricevitori economici che si possono collegare a una porta seriale. `Ntpd` include driver per un paio di ricevitori MSF.

Parecchie compagnie negli USA vendono orologi relativamente economici (compresi molti orologi analogici da parete) che hanno dei ricevitori WWVB incorporati, ma sono al corrente solamente di due che possano esser collegati ad un computer:

L'Ultralink Model 320, venduto per circa 120$ (a Giugno 1999), ha un'interfaccia seriale e un semplice set di comandi ASCII, perciò non dovrebbe essere troppo difficile da programmare. Trae 1mA dalla porta seriale per l'alimentazione. L'antenna può essere distante fino a 30 metri dal computer, e l'unità contiene un suo orologio per tenere il tempo se perde il segnale. Vendono anche una versione "bare board" per circa 80$ che è progettata per funzionare coi microcontroller delle serie "BASIC Stamp". Vedi `http://www.ulio.com/timepr.html` .

L'Arcron Technology vende un orologio da tavolo per circa 130$, comprendente il software per Windows. Vedi `http://www.arctime.com`