Mini HOWTO Clock

Ron Bean, Dec. 1996 rbean@execpc.com (Adaptation française par Gacquer frédéric gacquer@neuronnexion.fr) Dec. 1996

Jeudi 2 avril 1998 V 1.0.

Contents

1 Introduction

Les puces d'horloge temps-réel sur les cartes mères des PC (et même des stations de travail plus onéreuses) sont de notoriété publique inexactes. Linux fourni une manière simple pour corriger cela de manière logicielle, rendant virtuellement l'horloge *trés* précise même sans horloge externe. Mais la plupart des personnes ne semblent pas être au courant, pour plusieurs raisons :

1. Ce n'est pas indiqué dans la plupart des documentations "Comment installer linux", et cela serait difficile de le mettre en place automatiquement au moment de l'installation du système (bien que pas impossible en théorie, si vous avez un modem).
2. Si vous essayez "man clock" vous obtiendrez clock(3), ce qui n'est pas ce que vous souhaitez. (Essayez "man 8 clock").
3. La plupart des gens ne semblent pas tenir compte de l'heure qu'il est de toute façon.
4. Le peu qui y font attention régulièrement veulent utiliser le logiciel xntpd de louie.udel.edu pour se synchroniser à une horloge externe, comme un serveur réseau de temps ou une horloge radio.

Ce mini-howto décrit une approche de bas-niveau. Si vous êtes vraiment intéressé par ce genre de chose, je vous recommende vivement de passer du temps à `http://www.eecis.udel.edu/~ntp/` qui inclus toutes sortes de choses intéressantes, incluant une documentation complète sur xntpd et des liens sur NIST et USNO (j'ai quelques commentaires supplémentaires sur xntpd à la fin.)

Note si vous avez plus d'un système d'exploitation sur votre machine, vous devez en laisser seulement un remettre à jour l'horloge CMOS, ainsi ils n'interfèrerons pas l'un envers l'autre. Si vous exécutez régulièrement à la fois linux et windows sur la même machine, vous pourriez éventuellement préférer quelques uns des programmes sharewares de gestion de l'horloge disponible pour windows (suivre les liens à partir de l'URL ci-dessus).

2 Utiliser le programme 'clock'

Tout ce que vous devez savoir est dans la page de manuel clock(8), mais ce mini-HOWTO va vous guider pas à pas.

Note Vous devez être root pour exécuter 'clock', ou n'importe quel autre programme qui modifie soit le temps système soit l'horloge CMOS.

2.1 Vérifier votre installation

Cherchez dans vos fichiers de démarrage une commande du style 'clock -a' ou 'clock -ua'. Selon la distribution que vous utilisez, cela peut être dans `/etc/rc.local`, `/etc/rc.d/rc.sysinit`, ou dans un endroit similaire.

Si c'est 'clock -s' ou 'clock -us', changez le 's' en 'a', puis regardez si vous avez un fichier `/etc/adjtime`, ne contenant qu'une ligne ressemblant à quelque chose comme cela :

```
0.000000 842214901 0.000000
```

Ces nombres sont le facteur de correction (en secondes par jour), le moment où l'horloge a été réajustée la dernière fois (en secondes depuis le 1er janvier 1970), et les secondes partielles qui ont été arrondies la dernière fois. Si vous n'avez pas ce fichier, connectez vous en tant que root et créez le, avec une seule ligne qui ressemble à (que des zéros) :

```
0.0 0 0.0
```

Ensuite exécutez 'clock -a' ou 'clock -ua' manuellement à partir du shell pour mettre à jour le deuxième nombre (utiliser le 'u' si votre horloge est configurée en Universel plutôt que temps local).

2.2 Mesurer le taux de dérive temporelle de votre horloge

Pour commencer, vous devez connaitre l'heure qu'il est :-). Votre temps local du jour peut ou peut ne pas être exact. Ma méthode préférée est d'appeler l'horloge parlante au 3699 (c'est un appel gratuit). Si vous avez accés à un serveur réseau de temps, vous pouvez utiliser le programme ntpdate du progiciel xntpd (utiliser le paramètre -b pour empêcher le noyau de cafouiller l'horloge CMOS). Sinon utiliser 'date -s hh:mm:ss' pour mettre l'heure système à la main, puis 'clock -w' pour mettre à jour l'horloge CMOS à partir de l'heure du système. Vous devrez vous rappeler la dernière fois que vous avez changé l'heure, donc écrivez la date quelque part où vous ne la perdrez pas. Si vous avez utilisé ntpdate, faire 'date +%s' et écrire le nombre de secondes depuis le 1er janvier 1970.

Puis revenez quelques jours ou semaines après et regardez de combien l'horloge a dérivé. Si vous mettez l'horloge à jour à la main, je vous recommanderais d'attendre au moins deux semaines, et de seulement calculer le taux de dérive le plus proche d'un dixième de secondes par jour. Après plusieurs mois vous pourrez obtenir le plus proche d'un centième de secondes par jour (quelques personnes affirment être encore plus précis mais je resterais prudent dans ce cas). Si vous utilisez ntpdate, vous n'aurez pas à attendre si longtemps, mais dans tous les cas vous pourrez toujours affiner plus tard.

Vous pouvez avoir cron qui exécute 'clock -a' à des moments réguliers pour garder le temps système en accord avec le temps (corrigé) CMOS. Cette commande sera aussi exécutée à partir de vos fichiers de démarrage à chaque fois que vous relancerez le système, ainsi si vous le faites souvent (comme quelques uns d'entre nous le font), cela peut suffire à vos besoins.

Remarquez que certains programmes peuvent se plaindrent si le système saute plus d'une seconde à la fois, ou s'il retranche du temps. Si vous avez ce problème, vous pouvez utiliser xntpd ou ntpdate pour corriger le temps plus graduellement.

2.3 Exemple

2.3.1 Mettre à jour le temps

Se connecter root. Appeler le 3699 (vocal), écouter l'annonce de l'heure. Puis taper:

```
date -s hh:mm:ss
```

Mais ne tapez Entrée que lorsque vous entendez le bip. (vous pouvez utiliser 'ntpdate' ici plutôt que 'date', et éviter l'appel téléphonique). Cela met à jour le 'temps noyau'. Puis taper :

```
clock -w
```

Cela met à jour l'horloge CMOS pour correspondre au temps noyau. Puis taper:

```
date +%j
```

(ou 'date +%s' si vous utilisez 'ntpdate', plutôt que 'date' ci-dessus) et écrire le nombre qu'il vous donne pour la prochaine fois.

2.3.2 Pour remettre à jour le temps et vérifier le taux de dérive

Trouver la date que vous avez écrit la dernière fois. Se connecter root puis taper:

```
clock -a
```

Cela met à jour le temps noyau qui correspond ainsi au temps CMOS. Appeler le 3699 (vocal), écouter l'annonce. Puis taper :

```
date
```

et appuyer sur Entrée quand vous entendez le signal sonore, mais alors que vous attendez, notez le temps annoncé, et ne raccrochez pas de suite. Cela vous dit à quel temps votre machine pensait être, quand cela aurait du être exact à la minute. Maintenant entrez :

```
date hh:mm:00
```

utilisant la minute *après* celle qui vient juste d'être annoncée, et appuyez sur entrée quand vous entendez le signal sonore à nouveau (maintenant vous pouvez raccrocher). Pour hh utiliser le temps local. Cela met à jour le 'temps noyau'.

Puis taper :

```
clock -w
```

qui écrit le nouveau (correct) temps dans l'horloge CMOS. Maintenant tapez :

```
date +%j
```

(ou 'date +%s' si c'est ce que vous avez utilisé avant).

Vous avez maintenant trois nombres (deux dates et une heure) qui vont vous permettre de calculer la dérive de temps.

2.3.3 Calculer le facteur de correction

quand vous exécutez 'date' à l'instant, est-ce que votre machine était en avance ou en retard ? Si elle avançait, vous aurez à retrancher quelques secondes, alors écrivez le comme un nombre négatif. Si elle retardait, vous aurez à ajouter quelques secondes, alors écrivez le comme positif.

Maintenant soustrayez les deux dates. Si vous avez utilisé 'date +%j', les nombres représentent le jour de l'année (1-365, ou 1-366 pour les années bissextiles). Si vous avez passé le 1er janvier depuis votre dernière modification horaire vous aurez à ajouter 365 (ou 366) au deuxième nombre. Si vous avez utilisé 'date +%s' alors votre nombre est en secondes, et vous aurez à le diviser par 86400 pour obtenir des jours.

Si vous avez déjà un facteur de correction dans `/etc/adjtime`, vous aurez à tenir compte du nombre de secondes que vous avez déjà corrigé. Si vous avez trop corrigé, ce nombre aura le signe opposé à celui que vous venez juste de mesurer; si vous n'avez pas assez corrigé il aura le même signe. Multipliez l'ancien facteur de correction par le nombre de jour, et ensuite ajouter le nouveau nombre de secondes (addition signée – si les deux nombres ont le même signe, vous obtiendrez un nombre plus grand, s'ils ont des signes opposés vous aurez un nombre plus petit).

Puis divisez le nombre total de secondes par le nombre de jours pour obtenir le nouveau facteur de correction, et le mettre dans `/etc/adjtime` à la place de l'ancien. Conservez la nouvelle date (en secondes par jour) pour la prochaine fois.

Voici à quoi ressemble mon `/etc/adjtime` :

```
-9.600000 845082716 -0.250655
```

Note (on remarque que 9.6 secondes par jour c'est presque 5 minutes par mois !)

2.4 Quelques mots à propos de xntpd

Votre système a en fait deux horloges – l'horloge temps réel sur batterie qui garde trace du temps quand le système est éteint (aussi connue comme "l'horloge CMOS", "horloge matérielle" ou "RTC") et le 'temps noyau' (parfois appellée "horloge logicielle" ou "horloge système") qui est basé sur l'interruption timer et qui est initialisée à partir de l'horloge CMOS au démarrage du système. Les deux vont dériver à des rythmes différents, ainsi elles vont graduellement s'écarter l'une de l'autre, tout en s'écartant du temps 'réel'.

Toutes les références à "l'horloge" dans la documentation de xntpd se réfèrent à "l'horloge noyau". Quand vous exécutez xntpd ou timed (ou n'importe quel autre programme qui utilise l'appel système adjtimex), le noyau linux suppose que l'horloge du noyau est plus précise que l'horloge CMOS, et remets à jour le temps CMOS toutes les 11 minutes à partir de ce moment (jusqu'à ce que l'on relance l'ordinateur). Cela signifie que 'clock' ne sais plus quand l'horloge CMOS a été modifié la dernière fois, ainsi vous ne pouvez plus utiliser le facteur de correction dans `/etc/adjtime`. Vous pouvez utiliser ntpdate dans votre fichier de démarrage pour mettre à jour initialement l'horloge à partir d'un serveur de temps avant de lancer xntpd. Si vous n'avez pas toujours accés à une source fiable de temps lors de l'allumage de l'ordinateur, cela peut être un peu gênant – xntpd n'est pas vraiment conçu pour être utilisé dans des situations comme celles là.

Xntpd inclue des drivers pour plusieurs horloges radio, et peut être configuré pour appeler le service de temps téléphonique de NIST à des temps réguliers (soyez sûr de calculer l'incidence sur votre facture téléphonique lors du paramétrage de l'intervalle entre deux appels). Il peut aussi appliquer un facteur de correction à l'horloge noyau s'il perd contact avec les autres sources pour une période de temps assez longue.

La plupart des horloges radio coûtent 3-4000$, mais vous pouvez obtenir des plans pour une 'boite gadget' peu chère (en fait un modem 300 baud) qui se met entre votre ordinateur et n'importe quelle radio onde courte réglée sur la station de temps canadienne CHU (voir `ftp://ftp.udel.edu/pub/ntp/gadget.tar.Z`). Le récepteur Heathkit WWV ("l'Horloge la Plus Précise") est aussi encore disponible (bien que n'étant pas en kit), et coûte aux environs de 4-500 $. Les signaux GPS contiennent aussi des informations de temps, et quelques recepteurs GPS peuvent se connecter sur le port série. Cela peut être la solution la moins onéreuse dans un futur proche.

En théorie, on peut écrire un programme pour utiliser le service de temps téléphonique NIST pour calculer automatiquement le taux de dérive de l'horloge CMOS et de l'horloge noyau. Je ne suis pas au courant d'un quelconque programme démon qui ferait cela, mais la plupart du code peut être emprunté à xntpd.