

1. 什么是GMT (格林威治) 时间与UTC 时间？

GMT时间是以太阳通过格林威治的那一刻来作为计时的标准，地球共有24 个时区，而以格林威治时间(GMT) 为标准时间，台湾本地时间为GMT + 8 小时。不准确，但是方便记忆与理解。
UTC时间是使用『原子震荡周期』所计算的物理时钟。最准确。
两个时间计时的方式不同，GMT与UTC时间有差不多16分钟的误差！

2. 现在，你将有一个月的时间要出差到美国的纽约去，时间会不一致啊！你该如何手动的调整时间参数呢？

因为时区资料档在/usr/share/zoneinfo 内，在该目录内会找到usr/share/zoneinfo/America/New_York 这个时区档。而时区设定档在/etc/sysconfig/clock，且目前的时间格式在/etc/localtime，所以你应该这样做：

```
[root@www ~]# date
Thu Jul 28 15:08:39 CST 2011    <==重点是CST这个时区喔！

[root@www ~]# vim /etc/sysconfig/clock
ZONE=" America/New_York "      <==改的是这里啦！

[root@www ~]# cp /usr/share/zoneinfo/America/New_York /etc/localtime
[root@www ~]# date
Thu Jul 28 03:09:21 EDT 2011   <==时区与时间都改变了！
```

3. NTP服务配置文件参数讲解。

- 利用restrict 来管理权限控制

```
restrict [你的IP] mask [netmask_IP] [parameter]
```

其中parameter 的参数主要有底下这些：

1. ignore：拒绝所有类型的NTP连线；
2. nomodify： 用户端不能使用ntpc与ntpq这两支程式来修改伺服器的时间参数，但用户端仍可透过这部主机来进行网路校时的；
3. noquery： 用户端不能够使用ntpq, ntpc等指令来查询时间伺服器，等于不提供NTP的网路校时啰；
4. notrap： 不提供trap这个远端事件登录(remote event logging)的功能。
5. notrust： 拒绝没有认证的用户端。
6. kod： 向不安全的访问者发送Kiss-Of-Death报文^[1]。
7. nopeer： 提供时间服务，但不作为对等体，阻止主机尝试与服务器对等，不允许欺诈性服务器控制时钟。

那如果你没有在parameter的地方加上任何参数的话，这表示『该IP或网段不受任何限制』的意思喔！一般来说，我们可以先关闭NTP的使用权限，然后再一个一个的启用允许登入的网段。

常见的配置如下：

```
# 1.对于默认的客户端拒绝所有的操作
restrict default kod nomodify notrap nopeer noquery    <==拒绝IPv4的用户
restrict -6 default kod nomodify notrap nopeer noquery <==拒绝IPv6的用户
```

```
# 2.启内部递归网络接口lo 即允许本机地址一切的操作
restrict 127.0.0.1      <==底下两个是预设值·放行本机来源
restrict -6 ::1

# 3.允许上层时间服务器主动修改本机时间
restrict 220.130.158.71 <==放行tock.stdtime.gov.tw进入本NTP伺服器
restrict 59.124.196.83  <==放行tick.stdtime.gov.tw进入本NTP伺服器
restrict 59.124.196.84  <==放行time.stdtime.gov.tw进入本NTP伺服器

# 4.放行网段来源
restrict 192.168.100.0 mask 255.255.255.0 nomodify <==放行区网来源
```

- 利用server 设定上层NTP 伺服器

```
server [IP or hostname] [ key n ] [ version n ] [ prefer ] [ mode n ] [ minpoll n ] [ maxpoll n ] [ iburst ]
```

1. host： 是上层NTP服务器的IP地址或域名。
2. key： 表示所有发往服务器的报文包含有密钥加密的认证信息，n是32位的整数，表示密钥号。
3. version： 表示发往上层服务器的报文使用的版本号，n默认是3，可以是1或者2。
4. prefer： 如果有多个server选项，具有该参数的服务器优先使用。
5. mode： 指定数据报文mode字段的值。
6. minpoll： 指定与查询该服务器的最小时间间隔为2的n次方秒，n默认为6，范围为4-14。
7. maxpoll： 指定与查询该服务器的最大时间间隔为2的n次方秒，n默认为10，范围为4-14。
8. iburst： 当初始同步请求时，采用突发方式接连发送8个报文，时间间隔为2秒。

常见的配置如下：

```
# 5.默认的一个内部时钟数据·用在没有外部 NTP 服务器时·使用它为局域网用户提供服务
server 127.127.1.0
fudge 127.127.1.0 stratum 10

# 6.定义使用上层ntp服务
server 220.130.158.71 iburst minpoll 5 maxpoll 10 prefer <==以这部主机为最优先
server 59.124.196.83  iburst minpoll 5 maxpoll 10
server 59.124.196.84  iburst minpoll 5 maxpoll 10
```

- 以driftfile 记录时间差异

```
driftfile [可以被ntpd 写入的目录与档案]
```

因为预设的NTP Server 本身的时间计算是依据BIOS 的晶片震荡周期频率来计算的，但是这个数值与上层 Time Server 不见得会一致。所以NTP 这个daemon (ntpd) 会自动的去计算我们自己主机的频率与上层 Time server的频率，并且将两个频率的误差记录下来，记录下来的档案就是在driftfile 后面接的完整档名当中了！关于档名你必须要知道：

- 1. driftfile 后面接的档案需要使用完整路径档名；
- 2. 该档案不能是连结档；
- 3. 该档案需要设定成ntpd 这个daemon 可以写入的权限。
- 4. 该档案所记录的数值单位为：百万分之一秒(ppm)。

driftfile 后面接的档案会被ntpd 自动更新，所以他的权限一定要能够让ntpd 写入才行。在CentOS 6.x 预设的NTP 伺服器中，使用的ntpd 的owner 是ntp，这部份可以查阅/etc/sysconfig/ntpd 就可以知道啦！

常见的配置如下：

```
driftfile /var/lib/ntp/drift
```

- keys [key_file]

除了以restrict 来限制用户端的连线之外，我们也可以透过金钥系统来给用户端认证，如此一来可以让主机端更放心了。可以参考ntp-keygen 这个指令的相关说明。

- 其它的配置

```
pidfile    /var/run/ntpd.pid  <==进程位置
logfile    /var/log/ntp.log  <==日志文件
service    192.168.75.132   <==监听地址
```

4. 防火墙设置

ntp这个daemon是以port 123为连结的端口(使用UDP封包)

```
[root@www ~]# vim /usr/local/virus/iptables/iptables.allow
iptables -A INPUT -i eth0 -p udp -s 192.168.100.0/24 --dport 123 -j ACCEPT
```

5. NTP启动与观察

设定完ntp.conf 之后就可以启动ntp 服务器了。启动与观察的方式如下：

```
# 1.启动NTP
[root@www ~]# /etc/init.d/ntpd start
[root@www ~]# chkconfig ntpd on
[root@www ~]# tail /var/log/messages    <==自行检查看看有无错误

# 2.观察启动的埠口看看：
[root@www ~]# netstat -tlunp | grep ntp
```

```
Proto Recv-Q Send-Q Local Address Foreign Address PID/Program name
udp 0 0 192.168.100.254:123 0.0.0.0:* 3492/ntpd
udp 0 0 192.168.1.100:123 0.0.0.0:* 3492/ntpd
udp 0 0 127.0.0.1:123 0.0.0.0:* 3492/ntpd
udp 0 0 0.0.0.0:123 0.0.0.0:* 3492/ntpd
udp 0 0 ::1:123 :::* 3492/ntpd
udp 0 0 :::123 :::* 3492/ntpd
# 主要是UDP 封包，且在port 123 这个端口的啦！
```

这样就表示我们的NTP伺服器已经启动了，不过要与上层NTP服务器连线则还需要一些时间，通常启动NTP后约在15分钟内才会和上层NTP伺服器顺利连接上。

请自行等待数分钟后再以下列指令查阅：

```
[root@www ~]# ntpstat
synchronised to NTP server (220.130.158.71) at stratum 3
time correct to within 538 ms
polling server every 128 s
```

这个指令可以列出我们的NTP 服务器有跟上层连线否。由上述的输出结果可以知道，时间有校正约 538×10^{-3} 秒（538ms），且每隔128 秒会主动去更新时间！

```
[root@www ~]# ntpq -p
remote refid st t when poll reach delay offset jitter
=====
*tock.stdtime.go 59.124.196.87 2 u 19 128 377 12.092 -0.953 0.942
+59-124-196-83.H 59.124.196.86 2 u 8 128 377 14.154 7.616 1.533
+59-124-196-84.H 59.124.196.86 2 u 2 128 377 14.524 4.354 1.079
```

这个ntpq -p 可以列出目前我们的NTP 与相关的上层NTP 的状态，上头的几个栏位的意义为：

1. remote：亦即是NTP主机的IP或主机名称。注意最左边的符号^[2]：
2. refid：参考的上一层NTP主机的位址
3. st：就是stratum阶层。
4. when：几秒钟前曾经做过时间同步化更新的动作；
5. poll：下一次更新在几秒钟之后；
6. reach：已经向上层NTP伺服器要求更新的次数
7. delay：网路传输过程当中延迟的时间，单位为 10^{-3} 秒，即毫秒
8. offset：时间补偿的结果，单位与 10^{-3} 秒，即毫秒
9. jitter：Linux系统时间与BIOS硬体时间的差异时间，单位为 10^{-3} 秒，即毫秒。

差异都在0.001 秒以内，可以符合我们的一般使用了。另外，你也可以检查一下你的BIOS 时间与Linux 系统时间的差异，就是/var/lib/ntp/drift 这个档案的内容，就能了解到咱们的Linux 系统时间与BIOS 硬体时钟到底差多久。

6. 客户端时间更新方式

- 修改BIOS 记录的时间

```
[root@clientlinux ~]# hwclock [-rw]
选项与参数：
-r  ：亦即read  ，读出目前BIOS 内的时间参数；
-w  ：亦即write  ，将目前的Linux 系统时间写入BIOS 当中啊！

# 2.查阅BIOS时间，并且写入更改过的时间啰！
[root@clientlinux ~]# date; hwclock -r
Thu Jul 28 16:34:00 CST 2011
Thu 28 Jul 2011 03:34:57 PM CST -0.317679 seconds
#看一看，是否刚好差异约一个小时啊！这就是BIOS时间！

[root@clientlinux ~]# hwclock -w; hwclock -r; date
Thu 28 Jul 2011 04:35:12 PM CST -0.265656 seconds
Thu Jul 28 16:35:11 CST 2011
#这样就写入啰~所以软体时钟与硬体时钟就同步啦！很简单吧！
```

- ntpdate进行时间的同步

```
[root@clientlinux ~]# ntpdate [-dv] [NTP IP/hostname]
选项与参数：
-d  ：进入除错模式(debug) ，可以显示出更多的有效资讯。
-v  ：有较多讯息的显示。

[root@clientlinux ~]# ntpdate 192.168.100.254
28 Jul 17:19:33 ntpdate[3432]: step time server 192.168.100.254 offset -2428.396146 sec
#最后面会显示微调的时间有多少(offset)，因为这部主机时间差很多，所以秒数...

[root@clientlinux ~]# date; hwclock -r
四7月28 17:20:27 CST 2011
西元2011年07月28日(周四) 18时19分26秒 -0.752303 seconds
#知道想要表达什么吗？对啊！还得hwclock -w写入BIOS时间才行啊！

[root@clientlinux ~]# vim /etc/crontab
#加入这一行去！
10 5 * * * root (/usr/sbin/ntpdate tock.stdtime.gov.tw && /sbin/hwclock -w) &> /dev/null
```

- NTP服务更新时间

ntpdate这个方式仅适合不要启动NTP 的情况。如果你的机器数量太多了，那么用户端最好也启动一下NTP 服务。通过NTP 去主动的更新时间。


```
[root@clientlinux ~]# ntpdate 192.168.100.254
#由于ntpd的server/client之间的时间误差不允许超过1000秒，
# 因此你得先手动进行时间同步，然后再设定与启动时间伺服器呦！

[root@clientlinux ~]# vim /etc/ntp.conf
#server 0.centos.pool.ntp.org
#server 1.centos.pool.ntp.org
```

```
#server 2.centos.pool.ntp.org
restrict 192.168.100.254    <==放行伺服器来源！
server 192.168.100.254    <==这就是伺服器！
#很简单，就是将原本的server项目注解，加入我们要的伺服器即可

[root@clientlinux ~]# /etc/init.d/ntpd start
[root@clientlinux ~]# chkconfig ntpd on
```

然后取消掉crontab 的更新程序，这样你的client 电脑就会主动的到NTP 伺服器去更新。

1. KOD是NTPv4提出的一种全新的访问控制技术，主要用于服务器向客户端上提供状态报告和接入控制等信息。在服务器上使能KOD功能后，服务器会根据系统的运行状态向客户端发送DENY Kiss和RATE Kiss码。当客户端接收到DENY Kiss码，客户端将断开与服务器的所有连接，并停止向服务器发送报文。当客户端接收到RATE Kiss码，客户端将立即缩短与该服务器的轮询时间间隔，且以后每次接收到RATE Kiss码，轮询时间间隔都会进一步缩短。 

2. 如果有『 * 』代表目前正在作用当中的上层NTP;如果是『 + 』代表也有连上线，而且可作为下一个提供时间更新的候选者。