在Windwos中，系统时间的设置很简单，而且设置后，重启，关机都没关系。系统时间会自动保存在Bios的时钟里面，启动计算机的时候，系统会自动在Bios里面取硬件时间，以保证时间的不间断。

    但在Linux下，默认情况下，系统时间和硬件时间，并不会自动同步。在Linux运行过程中，系统时间和硬件时间以异步的方式运行，互不干扰。硬件时间的运行，是靠Bios电池来维持，而系统时间，是用CPU tick来维持的。

    在系统开机的时候，会自动从Bios中取得硬件时间，设置为系统时间。

# 一.Linux系统时间的设置

 在Linux中设置系统时间，可以用**date**命令：

//查看时间 , CST 是中国时区

[root@localhost ~]# date  
2008年 12月 12日星期五 14:44:12 CST

//修改时间  
[root@localhost ~]# date -set  "2013-12-24 00:01"< == （年/月/日 时:分【:秒】）  
2009年 01月 01日星期四 00:01:00 CST

[root@localhost ~]# date 012501012009.30  <== 月日时分年.秒  
2009年 01月 25日星期日 01:01:30 CST

# 二.Linux硬件时间的设置

    硬件时间的设置，可以用hwclock或者clock命令。其中，clock和hwclock用法相近，只用一个就行，只不过clock命令除了支持x86硬件体系外，还支持Alpha硬件体系。

//查看硬件时间可以是用hwclock，hwclock --show 或者hwclock -r

[root@localhost ~]# hwclock --show  
2008年12月12日星期五 06时52分07秒  -0.376932 seconds  
  
//设置硬件时间

[root@localhost ~]# hwclock --set --date="1/25/09 00:00" <== 月/日/年时:分:秒  
[root@localhost ~]# hwclock  
2009年01月25日星期日 00时00分06秒  -0.870868 seconds

[root@localhost ~]# hwclock  -w   根据系统时间设置硬件时间

# 三.系统时间和硬件时间的同步

    同步系统时间和硬件时间，可以使用hwclock命令。

//以系统时间为基准，修改硬件时间

[root@localhost ~]# hwclock --systohc<== sys（系统时间）to（写到）hc（Hard Clock）  
[root@localhost ~]# hwclock -w

//以硬件时间为基准，修改系统时间

[root@localhost ~]# hwclock --hctosys  
[root@localhost ~]# hwclock -s

# 四.不同机器之间的时间同步

    为了避免主机时间因为长期运作下所导致的时间偏差，进行时间同步(synchronize)的工作是非常必要的。Linux系统下，一般使用ntp服务器来同步不同机器的时间。一台机器，可以同时是ntp服务器和ntp客户机。在网络中，推荐使用像DNS服务器一样分层的时间服务器来同步时间。

    同步时间，可以使用ntpdate命令，也可以使用ntpd服务。

    使用ntpdate比较简单。格式如下：

[root@linux ~]# ntpdate [-nv] [NTP IP/hostname]  
[root@linux ~]# ntpdate 192.168.0.2  
[root@linux ~]# ntpdate time.ntp.org

    但这样的同步，只是强制性的将系统时间设置为ntp服务器时间。如果cpu tick有问题，只是治标不治本。所以，一般配合cron命令，来进行定期同步设置。比如，在crontab中添加：

0 12 \* \* \* \* /usr/sbin/ntpdate 192.168.0.1

     这样，会在每天的12点整，同步一次时间。ntp服务器为192.168.0.1。

    使用ntpd服务，要好于ntpdate加cron的组合。因为，ntpdate同步时间，会造成时间的跳跃，对一些依赖时间的程序和服务会造成影响。比如sleep，timer等。而且，ntpd服务可以在修正时间的同时，修正cpu tick。理想的做法为，在开机的时候，使用ntpdate强制同步时间，在其他时候使用ntpd服务来同步时间。

    要注意的是，ntpd有一个自我保护设置: 如果本机与上源时间相差太大, ntpd不运行. 所以新设置的时间服务器一定要先ntpdate从上源取得时间初值, 然后启动ntpd服务。ntpd服务运行后, 先是每64秒与上源服务器同步一次, 根据每次同步时测得的误差值经复杂计算逐步调整自己的时间, 随着误差减小, 逐步增加同步的间隔. 每次跳动, 都会重复这个调整的过程.

五.ntpd服务的设置

    ntpd服务的相关设置文件如下：

1./etc/ntp.conf：这个是NTP daemon的主要设文件，也是 NTP 唯一的设定文件。

2./usr /share/zoneinfo/:在这个目录下的文件其实是规定了各主要时区的时间设定文件，例如北京地区的时区设定文件在 /usr/share/zoneinfo/Asia/Beijing 就是了。这个目录里面的文件与底下要谈的两个文件(clock 与localtime)是有关系的。

3./etc/sysconfig/clock：这个文件其实也不包含在NTP 的 daemon 当中，因为这个是linux的主要时区设定文件。每次开机后，Linux 会自动的读取这个文件来设定自己系统所默认要显示的时间。

4./etc /localtime：这个文件就是“本地端的时间配置文件”。刚刚那个clock 文件里面规定了使用的时间设置文件(ZONE) 为 /usr/share/zoneinfo/Asia/Beijing ，所以说，这就是本地端的时间了，此时， Linux系统就会将Beijing那个文件另存为一份 /etc/localtime文件，所以未来我们的时间显示就会以Beijing那个时间设定文件为准。

5. /etc/timezone：系统时区文件

    下面重点说说 /etc/ntp.conf文件的设置。在 NTP Server 的设定上面，其实最好不要对 Internet 无限制的开放，尽量仅提供您自己内部的 Client 端联机进行网络校时就好。此外， NTP Server 总也是需要网络上面较为准确的主机来自行更新自己的时间啊，所以在我们的 NTP Server 上面也要找一部最靠近自己的 Time Server 来进行自我校正。事实上， NTP 这个服务也是 Server/Client 的一种模式。

[root@linux ~]# vi /etc/ntp.conf  
# 1. 关于权限设定部分  
#　　权限的设定主要以 restrict 这个参数来设定，主要的语法为：  
# 　　restrict IP mask netmask\_IP parameter  
# 　　其中 IP 可以是软件地址，也可以是 default ，default 就类似 0.0.0.0  
#　　至于paramter则有：  
#　　　ignore　：关闭所有的 NTP 联机服务  
#　　　nomodify：表示 Client 端不能更改 Server 端的时间参数，不过，

#　　　Client 端仍然可以透过 Server 端来进行网络校时。  
#　　　notrust：该 Client 除非通过认证，否则该 Client 来源将被视为不信任网域  
#　　　noquery：不提供 Client 端的时间查询

#　　　notrap：不提供trap这个远程事件登入

#　　如果paramter完全没有设定，那就表示该 IP (或网域)“没有任何限制”

restrict default nomodifynotrapnoquery　# 关闭所有的 NTP 要求封包  
restrict 127.0.0.1　　　 #这是允许本级查询  
restrict 192.168.0.1 mask 255.255.255.0 nomodify  
#在192.168.0.1/24网段内的服务器就可以通过这台NTP Server进行时间同步了  
# 2. 上层主机的设定  
#　　要设定上层主机主要以 server 这个参数来设定，语法为：  
#　　server [IP|HOST Name] [prefer]  
#　　Server 后面接的就是我们上层 Time Server 啰！而如果 Server 参数  
#　　后面加上perfer的话，那表示我们的 NTP 主机主要以该部主机来作为  
#　　时间校正的对应。另外，为了解决更新时间封包的传送延迟动作，  
#　　所以可以使用driftfile来规定我们的主机  
#　　在与 Time Server 沟通时所花费的时间，可以记录在driftfile   
#　　后面接的文件内，例如下面的范例中，我们的 NTP server 与   
#　　cn.pool.ntp.org联机时所花费的时间会记录在 /etc/ntp/drift文件内  
server 0.pool.ntp.org

server 1.pool.ntp.org

server 2.pool.ntp.org

server cn.pool.ntp.org prefer

#其他设置值，以系统默认值即可

server  127.127.1.0     # local clock

fudge   127.127.1.0 stratum 10

driftfile /var/lib/ntp/drift  
broadcastdelay  0.008  
keys /etc/ntp/keys

总结一下，restrict用来设置访问权限，server用来设置上层时间服务器，driftfile用来设置保存漂移时间的文件。

六.ntp服务的启动与观察

在启动NTP服务前，先对提供服务的这台主机手动的校正一次时间咯。（因为启动服务器，端口会被服务端占用，就不能手动同步时间了）

[root@linux ~] # ntpdate cn.pool.ntp.org

25 Apr 14:33:51 ntpdate[8310]: step time server 80.85.129.2 offset 6.655976 sec

然后，启动ntpd服务：

[root@linux ~] # servicentpd start

或 [root@linux ~] # /etc/init.d/ntpd start

查看端口：

[root@linux ~] # netstat -ln|grep 123

udp        0      0 192.168.228.153:123        0.0.0.0:\*

udp        0      0 127.0.0.1:123               0.0.0.0:\*

udp        0      0 0.0.0.0:123                  0.0.0.0:\*

udp        0      0 :::123                       :::\*

如何确认我们的NTP服务器已经更新了自己的时间呢？

[root@linux ~] # ntpstat

synchronized to NTP server(127.127.1.0) at stratum 11

time correct to within 950ms

polling server every 64 s

#改指令可列出NTP服务器是否与上层联机。由上述输出结果可知，时间校正约

#为950\*10(-6)秒。且每隔64秒会主动更新时间。

常见的错误：

25 Apr 15:30:17 ntpdate[11520]: no server suitable for synchronization found

其实，这不是一个错误。而是由于每次重启NTP服务器之后大约要3－5分钟客户端才能与server建立正常的通讯连接。当此时用客户端连接服务端就会报这样的信息。一般等待几分钟就可以了。

[root@linux ~] # ntptrace –n 127.0.0.1

127.0.0.1:stratum 11, offset 0.000000，synch distance 0.950951

222.73.214.125：stratum 2，offset –0.000787，synch distance 0.108575

209.81.9.7:stratum 1，offset 0.000028，synch distance 0.00436，refid ‘GPS’

#这个指令可以列出目前NTP服务器（第一层）与上层NTP服务器（第二层）彼此之间的

#关系

[root@linux ~] # ntpq –p

指令“ntpq -p”可以列出目前我们的NTP与相关的上层NTP的状态，以上的几个字段的意义如下：

remote：即NTP主机的IP或主机名称。注意最左边的符号，如果由“+”则代表目前正在作用钟的上层NTP，如果是“\*”则表示也有连上线，不过是作为次要联机的NTP主机。

refid：参考的上一层NTP主机的地址

st：即stratum阶层

when：几秒前曾做过时间同步更新的操作

poll：下次更新在几秒之后

reach：已经向上层NTP服务器要求更新的次数

delay：网络传输过程钟延迟的时间

offset：时间补偿的结果

jitter：Linux系统时间与BIOS硬件时间的差异时间

    最后提及一点，ntp服务，默认只会同步系统时间。如果想要让ntp同时同步硬件时间，可以设置/etc/sysconfig/ntpd文件。

在/etc/sysconfig/ntpd文件中，添加 SYNC\_HWCLOCK=yes 这样，就可以让硬件时间与系统时间一起同步。

---------------------------------------------------------------------------------------------------------------------------------------------------------------------

linux配置时间服务器（ntp）

2007-10-27 16:34

在linux下，我们可以通过自带的NTP(Network Time Protocol)协议通过网络使自己的系统保持精确的时间。可用的公共时间服务器列表可以从下面的地址获取：  
  
　　http://ntp.isc.org/bin/view/Servers/NTPPoolServers  
  
　　NTP是用来使系统和一个精确的时间源保持时间同步的协议。建议大家在自己管理的网络中建立至少一台时间服务器来同步本地时间，这样可以使得在不同的系统上处理和收集日志和管理更加容易。  
  
　　介绍一下环境：179为本地时间服务器，其他服务器和179同步。179和网上时间服务器同步。  
  
　　一、配置179时间服务器  
  
　　1、首先查询NTP软件版本  
  
　　rpm -qa|grepntp  
  
　　ntp-4.1.2-4.EL3.1  
  
　　如果没有可以从linux安装盘上查找，安装此ntp包  
  
　　2、编辑配置文件  
  
　　vi /etc/ntp.conf  
  
　　首先定义服务器  
  
　　server pool.ntp.org  
  
　　restrict default nomodifynotrapnoquery  
  
　　restrict 192.168.0.0 mask 255.255.255.0 notrustnomodifynotrap #从192.168.0.1-192.168.0.254的服务器都可以使用我们的NTP服务器来同步时间。  
  
　　注释掉以下一行  
  
　　#restrict default ignore  
  
　　3、启动NTP服务器  
  
　　#chkconfigntpd on  
  
　　#/etc/init.d/ntpd start  
  
　　#/etc/init.d/ntpd stop  
  
　　#/etc/init.d/ntpd restart  
  
　　默认情况下，我们配置的NTP服务器不会去时间源那里同步时间，所以必须修改/etc/ntp/step-tickers文件，加入我们的时间源，这样每次通过/etc/init.d/ntpd来启动服务的时候就会自动更新时间了  
  
　　检查服务器同步状态：  
  
　　ntpq -p  
  
　　ntptrace 192.168.0.179  
  
　　如果输出正确，则说明时间服务器成功。每次启动服务器，会自动同步时间。  
  
　　配置LINUX客户端  
  
　　在linux客户端上执行ntpdatentp\_server\_ip就可以根据时间服务器统一局域网的时间了，将上面的命令放到cron里每天早上3点定期执行，crontab –e 然后输入  
  
　　0 3 \* \* \* /usr/sbin/ntpdate 192.168.0.179

为什么电脑要对时？因为电脑自己的钟不准。破电脑一天慢五分钟也不奇怪。

对时需要对到什么精度？一般家用的电脑时钟误差一分钟完全可以接受。集群服务器配合工作一般需要所有的时钟同步在一秒之内。

和谁同步？一般是和世界各地的 NTP （Network Time Protocol）服务器同步的。米国标准时间由 NIST 发布，NIST 提供了一些[服务器](http://tf.nist.gov/service/time-servers.html)。现代 Windows 操作系统自动和 time.windows.com 对时。Linux 下面一般用[NTPPool](http://support.ntp.org/bin/view/Servers/NTPPoolServers)来自动选择服务器。中国国家授时中心 NTP 210.72.145.44 是中国权威时间。中国教育网有自己的[NTP 服务器网](http://time.edu.cn/mem.htm)。如果是集群服务器，一般会在内网配置几个本地 NTP 服务器。

上面这么多 NTP 服务器有没有优劣呢？有的，但是对于一般的应用来说，看不出什么区别。理论上说，NTP 服务器是分等级（Stratum）的，Stratum = 1 的 NTP 服务器是直接和世界标准时钟同步的，包括 GPS 时间、铯原子钟、某些手机网络等。NIST、中国国家授时中心和中国教育网的第一级时间服务器都是这个级别的。Stratum = 2 的 NTP 服务器是和 Stratum = 1 的服务器同步的，性能稍差，但精确度也在毫秒的量级，所以用起来没什么区别。再往下每同步一级，Stratum 就加一。

Windows 2000 开始包含了自动时间同步的服务。Windows XP 上，打开时间设置就有网络对时的设置。默认是每星期和 time.windows.com 对时一次。这个一星期同步一次太长了，可以通过注册表调整，或者用[Wits](http://www.anobyte.com/wits/index.htm)修改。

Linux 下面的ntpd就要灵活多啦。默认配置一般足够好用。下面在 Fedora 下举个例子。配置：

# grep "^server" /etc/ntp.conf  
server 0.pool.ntp.org  
server 1.pool.ntp.org  
server 2.pool.ntp.org  
server pool.ntp.org  
server   127.127.1.0 # local clock

启动ntpd之后查看状态

# ntpq -pn  
      remote            refid       st t when poll reach    delay    offset   jitter  
==============================================================================  
 64.25.87.54      128.118.25.5      2 u    10    64    17    79.194   -542.89    1.942  
 64.72.116.51     129.7.1.66        2 u     9    64    17    51.569   -532.23    1.803  
 64.72.116.50     129.7.1.66        2 u    11    64    17    51.417   -516.70    1.417  
 64.72.116.45     129.7.1.66        2 u     7    64    17    51.586   -532.36    1.135  
\*127.127.1.0      LOCAL(0)         10 l     3    64    17     0.000     0.000    0.001

st这列显示自动选的四个 NTP 服务器都是 Stratum = 2。最后一个是本地时钟。前四行的 offset 显示本地时钟和四个 NTP 服务器都有大概 500 毫秒的差距。ntpstat显示目前只和本地时钟同步：

# ntpstat  
synchronised to local net at stratum 11  
    time correct to within 949 ms  
    polling server every 64 s

过一会再看：

# ntpq -pn  
      remote            refid       st t when poll reach    delay    offset   jitter  
==============================================================================  
+64.25.87.54      128.118.25.5      2 u    56    64   377    78.548   250.871   37.180  
+64.72.116.51     129.7.1.66        2 u    58    64   377    51.551   268.538   36.817  
\*64.72.116.50     129.7.1.66        2 u    58    64   377    51.539   274.497   36.629  
+64.72.116.45     129.7.1.66        2 u    49    64   377    51.485   271.750   37.841  
 127.127.1.0      LOCAL(0)         10 l    44    64   377     0.000     0.000    0.001  
# ntpstat  
synchronised to NTP server (64.72.116.50) at stratum 3  
    time correct to within 263 ms  
    polling server every 64 s

本地时钟已经成功和外面的 NTP 服务器同步。ntpq报告中第一列 \* 表示目前选择的主同步服务器，标 + 的表示有可能被用来进一步提高同步精度的次要服务器。因为是和 Stratum = 2 的服务器同步，所以本地的ntpd Stratum 就是 3 了。一个细节是ntpq对时钟是慢慢调整的，而不是直接跳好多秒，这样平滑的调整时间可以保证很多程序的流程平稳。不过，如果时钟误差过大，ntpd可能会拒绝调整时间；或者有人也可能希望立刻调正时间，这样的话可以直接执行命令：ntpdate -b pool.ntp.org（需要停掉ntpd服务执行）。

在我另一个服务器上，配置了好多 NTP 服务器：

# grep "^server" /etc/ntp.conf  
server time-a.nist.gov  
server time-b.nist.gov  
server time.nist.gov  
server time.windows.com  
server 0.pool.ntp.org  
server 1.pool.ntp.org  
server 2.pool.ntp.org  
server pool.ntp.org  
server   127.127.1.0 # local clock  
# ntpq -p  
      remote            refid       st t when poll reach    delay    offset   jitter  
==============================================================================  
+time-a.nist.gov .ACTS.            1 u    61   128   377    76.113    -1.046    3.424  
\*time-b.nist.gov .ACTS.            1 u    65   128   377    81.063     0.398    1.892  
-time.nist.gov    .ACTS.            1 u   251   128   356    38.911     1.353   30.226  
-time.windows.co 18.26.4.105       2 u    45   128   267    31.218    13.180    6.039  
-194.109.64.200   192.87.106.2      2 u   122   128   377   155.132     0.596   38.674  
-a.mirror.fizzel 43.75.42.44       3 u    56   128   377   163.391   -11.756   13.006  
-enfield.ikk.szt 195.111.99.186    2 u   118   128   377   188.326    -2.520   32.359  
+ntp1.esat.net    .GPS.             1 u    59   128   377   161.103    -1.321    0.460  
LOCAL(0)         .LOCL.           10 l    48    64   377     0.000     0.000    0.001

标注 - 的是那些相对来说不太准的钟（offset 或 jitter 偏大），自动被剔除了。从 delay 这里可以看到，我这里和 NIST 几个钟网络延迟比较小（100 毫秒之内），一般这样的钟误差小一些。倒数第二个 pool.ntp.org 选择的是一个和 GPS 同步的钟，Stratum = 1，是一个比较准的钟了，只是网络延迟稍大。同时可以看到 time.windows.com 的 Stratum = 2，在一年以前，它曾经是 6。堂堂国际大公司微软也不花几百块钱买个 GPS 接收器建一个Stratrum = 1 的时钟服务器造福千千万万的 windows 用户，真是不可理解。

最后，linux下防火墙规则如果极严格的话可能会影响ntpd对时，打开 sport 123 即可（假设 OUTPUT 链全 ACCEPT）：

iptables -I INPUT -p udp -m udp --sport 123 -j ACCEPT