

Exercicis gestió del temps

- 1. Necessitem mesurar el temps, i per fer-ho necessitem una referència: una hora de referència. Abans es feia servir GMT. Que és?**

GMT és simplement un acrònim de Greenwich Mean Time. Per defecte hauria de significar que el nostre rellotge mostra l'hora del meridià de Greenwich, però la realitat és molt més senzilla.

- 2. Quina és l'hora de referència actual, molt semblant a la GMT, a partir de la qual mesurem les altres hores? Com es diuen aquestes "altres hores"?**

Les sigles UTC signifiquen "Temps Universal Coordinat" (en anglès, "Universal Time Coordinated"), que antigament va ser cridada "l'hora en el meridià de Greenwich" ("GMT") o el "temps Zulu" ("Z"). Aquesta és l'hora local al Meridià Primari (i.e., áquel la longitud és 0 °) donada en hores i minuts en el rellotge de 24 hores. Per exemple, considerem l'hora 1350 UTC que significa 13 hores i 50 minuts després de la mitjanit, o 01:50 pm al Meridià Primari.

- 3. Els estats com Espanya, França o Portugal sempre tenen la mateixa franja horària? O en algún moment canvien de franja horària? Com es diu aquest canvi horari?**

L'horari d'estiu: L'horari d'estiu (també denominat DST, Daylight saving time o horari d'estalvi de llum diürna) és l'horari que obeeix a la convenció per la qual s'avancen els rellotges; el motiu és per aprofitar més la llum diürna, especialment quan aquesta es redueix als matins.

- 4. Cada dia apaguem l'ordinador al marxar cap a casa i al dia següent l'engeguem. I ens dona l'hora, fins i tot sense que hi hagi connexió a internet. Com pot ser això?**

És aquí on entra en joc la pila. Aquest element, el qual va connectat directament a la placa base, permet emmagatzemar la configuració bàsica de la BIOS tot i que l'ordinador es trobi apagat i sense estar connectat a una font de subministrament energètic, ja sigui un carregador, una bateria o una font de alimentació.

5. El rellotge de hardware emmagatzema l'any, mes, dia, hora, minut i segon. Alguna dada més per saber quan arrenqui el sistema si l'ordinador es troba a Bora Bora o a Rio de Janeiro?

No, l'ordinador no té un sistema de GPS incorporat a l'equip. Només incorpora el sistema UTC per a calibrar la data exacta.

6. Hi ha una segona hora quan l'ordinador està engegat: la del sistema (o de software). Com creus que s'activa aquesta? I que penses que passa quan s'apaga el sistema?

L'hora del sistema es manté engegada tota l'estona encara que no estigui encesa, la del software només s'encén quan el sistema esta engegada.

7. Si el RTC no sap si és UTC o hora local, per defecte, quina és la referència del RTC segons un sistema operatiu Windows? I un Mac? I un GNU/Linux?

El RTC és un rellotge d'ordinador (generalment en forma de circuit integrat) que manté l'hora actual. Els RTCs són presents en gairebé tots els dispositius electrònics que necessiten del temps actual.

Linux: Dos rellotges són importants a Linux: un "rellotge de maquinari", també conegut com RTC, rellotge CMOS o BIOS. Aquest és el rellotge amb bateria que manté el temps fins i tot quan el sistema es tanca. El segon rellotge s'anomena "rellotge del sistema" o "rellotge del nucli" i el sistema operatiu manté. A l'arrencada, el rellotge del maquinari es llegeix i s'utilitza per configurar el rellotge del sistema. A partir d'aquest moment, el rellotge del sistema s'utilitza per fer el seguiment del temps.

Windows: La versió 1.2 de la interfície de programació de l'aplicació Client de Microsoft (RTC) en temps real es pot utilitzar per a aplicacions d'escriptori de l'usuari final o per als agents automatitzats per proporcionar funcionalitat a un gran nombre d'usuaris.

Mac: Només a Mac OS s'admet el component del client (inclòs el client Eclipse i el client web). Si també voleu instal·lar el component del servidor, no és compatible (es pot executar, però sota el vostre propi risc).

8. Juguem amb l'ordre timedatectl i el significat de les diferents línies:

```
[billy_portes@pc666 ~]$ timedatectl
    Local time: Mon 2016-11-28 13:50:49 CET
    Universal time: Mon 2016-11-28 12:50:49 UTC
            RTC time: Mon 2016-11-28 12:50:49
    Time zone: Europe/Madrid (CET, +0100)
Network time on: no
NTP synchronized: yes
RTC in local TZ: no
```

1. El nom de l'usuari, el nom de l'ordinador i la data emprada → `timedatectl`.
2. La data local.
3. La data universal
4. L'hora real del rellotge.
5. UTC.
6. L'hora via xarxa està desactivada.
7. NTP està sincronitzada.
8. RTC no està en local TZ.

Demana aquest exercici al professor.

9. Mostrem la **data/hora** del sistema actual:

```
[isx36579183@j20 ~]$ date
Thu Nov 30 13:40:03 CET 2017
```

10. Mostra la data de demà (ha de ser la **mateixa ordre** sempre) en format dd/mm/YYYY

```
[isx36579183@j20 ~]$ (date --date='tomorrow' +"%d/%m/%Y")
01/12/2017
```

11. Abans d'executar la següent ordre, intenta esbrinar que fa?

```
[ $(date +%d%m%Y) = "20051992" ] && echo "Goooooooo de Koeman"
```

Mostra l'hora en format dd/mm/YY de la data 20/05/1992 y concatena un string de «Goooooooo de Koeman»

12. Estableix la data del sistema una setmana abans de la de l'exercici anterior.

```
[isx36579183@j20 ~]$ date -d "1 weeks ago"
Thu Nov 23 13:53:08 CET 2017
```

13. *offtopic* Mostra un calendari del proper any amb el dilluns com a primer dia de la setmana:

14. Estableix l'hora del sistema a les 15:15:00

```
[isx36579183@j20 ~]$ date --set "2017-12-05 15:15:00"
```

15. Que passa si activem ara el servidor de temps? És a dir si executem:
`systemctl start chronyd`

La ubicació predeterminada del dimoni de crony és /usr/sbin/chronyd. La utilitat de línia d'ordres s'instal·larà a /usr/bin/chronyc.
S'inicia el servei chronyd.

16. Mostra totes les zones horàries. Mostra les zones horàries europees.

```
ls -la /etc/localtime
lrwxrwxrwx 1 root root 1725 Jan 12 2011 /etc/localtime -> /usr/share/zoneinfo/Europe/Andorra
```

Mitjançant la variable d'entorn TZ podem seleccionar una zona horària diferent i executar-la amb

```
date:
# TZ="Europe/Dublin" date
Fri Nov 4 05:58:36 GMT 2011
# TZ="America/Los_Angeles" date
Thu Nov 3 22:58:42 PDT 2011
# TZ="America/Toronto" date
Fri Nov 4 01:58:47 EDT 2011
```

17. Estableix l'hora del sistema com la d'Atenes

```
[isx36579183@j20 ~]$ date --set "2017-12-05 20:23:00"
```

20:23

lunes, 4 de diciembre de 2017 (EET)
Hora en Atenas, Grecia

18. Sembla que `timedatectl` mostrava el darrer i següent DTS segons [aquest enllaç] (https://docs.fedoraproject.org/en-US/Fedora/24/html/System_Administrators_Guide/ch-Configuring_the_Date_and_Time.html) però ja no ho fa.

Tot i així podem trobar aquesta informació. Primer de tot, necessitem saber quina és la nostra zona horària (al nostre cas `*Europe/Madrid*`).

Si ara fem:

```
zdump -v Europe/Madrid
```

Podem mostrar només les línies referides a aquest any (suposem 2017) i el següent (suposem 2018):

```
zdump -v Europe/Madrid -c 2017,2018
```

O de forma més genèrica perquè ens serveixi sempre:

```
zdump -v Europe/Madrid -c $(date '+%Y'),$(date -d '+1 year' '+%Y')
```

Suposem que a més volem retallar la info que ens interessa, mes, dia i any dels dies que hi ha DTS,

això es pot fer amb una ordre molt potent, `awk`:

```
zdump -v Europe/Madrid -c $(date '+%Y'),$(date -d '+2 year' '+%Y') | awk -v y=$((date '+%Y')+1)) '$6==y && $15~1 {print $4, $3, $6}'
```

que si estiguéssim al 2017 ens mostraria:

```
26 Mar 2017
29 Oct 2017
```
```

Podeu fer el mateix pel següent any.

```
```
```

```
zdump -v Europe/Madrid -c $(date '+%Y'),$(date -d '+1 year' '+%Y') | awk
-v y=$(date '+%Y') '$6==y && $15~1 {print $4, $3, $6}'
```
```

Feu el mateix sense utilitzar awk, amb les ordres grep i cut per exemple.  
No cal mateix ordre.

És a dir és suficient que es mostri per a l'any 2017:

```
```
```

```
Sun Mar 26
Sun Oct 29
```
```

### **19. Com es diu el protocol que ajuda a sincronitzar un ordinador amb un servidor de temps?**

I pots posar dos exemples de serveis/dimonis que fan servir aquest protocol? Quin fa servir el nostre Fedora?

Network Time Protocol: NTP.

El protocol de temps (Network Time Protocol o NTP) és un protocol de xarxa per sincronitzar rellotges entre ordinadors. Aquest servei està des de l'any 1985 i és un dels protocols més antics.

NTP aquesta fet per sincronitzar amb una diferència d'uns mili segons respecte als seus servidors de temps. Els servidors consulten l'hora per aquest protocol i sincronitzen els seus rellotges amb aquests servidors de temps. D'aquesta manera l'hora és molt propera a l'exacta.

### **20. Que fa la següent ordre?**

```
```
```

```
chronyd -q 'server 2.fedora.pool.ntp.org iburst'
```

```
```
```

**Chronyd ajustarà l'hora del sistema i sortirà. S'ajustarà amb el servidor 2.fedora.pool.ntp.org.**

Es necessita que estigui parat el dimoni del temps.

El server es pot treure de:

```
```
```

```
/etc/chrony.conf
```

```
```
```

també es pot utilitzar com a servidor de temps

```
```
```

```
pool.ntp.org
```

```
```
```

## 21. Exercici per jugar amb els servidors de temps:

Juguem amb terminator per exemple i dues seccions (o amb dues sessions de gnome-terminal).

En una examinarem l'ordre `timedatectl` com a usuari normal i a l'altre executarem ordres com a root.

Ens interessa veure com canvia l'ordre `timedatectl`, podem utilitzar per a això `timedatectl` amb l'ordre `watch`

```
watch timedatectl
```

```
Every 2.0s: timedatectl Mon Dec 4 19:35:20 2017

 Local time: Mon 2017-12-04 19:35:20 CET
 Universal time: Mon 2017-12-04 18:35:20 UTC
 RTC time: Mon 2017-12-04 18:35:21
 Time zone: Europe/Madrid (CET, +0100)
 Network time on: yes
 NTP synchronized: yes
 RTC in local TZ: no
```

Millor encara volem que es ressaltin les diferències entre un instant de temps i el següent

```
watch -d timedatectl
```

```
Every 2.0s: timedatectl Mon Dec 4 19:35:55 2017

 Local time: Mon 2017-12-04 19:35:55 CET
 Universal time: Mon 2017-12-04 18:35:55 UTC
 RTC time: Mon 2017-12-04 18:35:55
 Time zone: Europe/Madrid (CET, +0100)
 Network time on: yes
 NTP synchronized: yes
 RTC in local TZ: no
```

Millor encara només em fixo en les 3 darreres línies de l'ordre, necessito llegir `*man*` per les cometes

```
watch -d 'timedatectl | tail -3'
```

```
Every 2.0s: timedatectl | tail -3 Mon Dec 4 19:36:56 2017

 Network time on: yes
 NTP synchronized: yes
 RTC in local TZ: no
```

A l'altra secció/terminal, com a root, fem el següent:

```
systemctl status chronyd # comprovem l'estat del dimoni chrony
```

```
[root@localhost ~]# systemctl status chronyd
• chronyd.service - NTP client/server
```

```

Loaded: loaded (/usr/lib/systemd/system/chronyd.service; enabled; vendor pres
Active: active (running) since Mon 2017-12-04 19:28:34 CET; 8min ago
Process: 771 ExecStartPost=/usr/libexec/chrony-helper update-daemon (code=exit
Process: 720 ExecStart=/usr/sbin/chronyd $OPTIONS (code=exited, status=0/SUCCE
Main PID: 742 (chronyd)
Tasks: 1 (limit: 512)
CGroup: /system.slice/chronyd.service
 └─742 /usr/sbin/chronyd

```

```

Dec 04 19:28:32 localhost.localdomain systemd[1]: Starting NTP client/server...
Dec 04 19:28:33 localhost.localdomain chronyd[742]: chronyd version 2.3 starting
Dec 04 19:28:33 localhost.localdomain chronyd[742]: Frequency -256.393 +/- 1.930
Dec 04 19:28:34 localhost.localdomain systemd[1]: Started NTP client/server.
Dec 04 19:29:06 localhost.localdomain chronyd[742]: Selected source 150.214.94.1
Dec 04 19:29:06 localhost.localdomain chronyd[742]: System clock wrong by 8.6453
Dec 04 19:29:15 localhost.localdomain chronyd[742]: System clock was stepped by
Dec 04 19:29:17 localhost.localdomain chronyd[742]: Selected source 147.156.7.18

```

Comprovem que està arrencat el servei i després el parem i observem els canvis a l'altra terminal

```
systemctl stop chronyd
```

Arranquem ara el servei i mirem que passa a l'altre terminal (2 canvis):

```
systemctl start chronyd
```

```

[root@localhost ~]# systemctl stop chronyd
[root@localhost ~]# systemctl start chronyd
[root@localhost ~]# █

```

Fem un watch de l'ordre 'chronyc sources', no és necessari ser root, que ens dona informació dels servidors de temps a on es connecta chronyd

```
watch chronyc sources
```

```

Every 2.0s: chronyc sources Mon Dec 4 19:38:34 2017

210 Number of sources = 6
MS Name/IP address Stratum Poll Reach LastRx Last sample
=====
^* shackleton.ci.uv.es 2 6 17 21 +1490us[+1049us] +/- 40ms
^+ ntp.redimadrid.es 2 6 17 22 -3018us[-3459us] +/- 33ms
^? scott.red.uv.es 0 6 0 - +0ns[+0ns] +/- 0ns
^? bjaaland.red.uv.es 0 6 0 - +0ns[+0ns] +/- 0ns
^? ntp.redimadrid.es 0 6 0 - +0ns[+0ns] +/- 0ns
^? shackleton.red.uv.es 0 6 0 - +0ns[+0ns] +/- 0ns

```

Esperem una estona (2 minuts com a màxim) i comprovem com es sincronitza i es dona un altre canvi.

Quan la columna LastRx, que representa els segons que fa que no es sincronitza amb un dels servidors es posi a 0, es donarà aquest canvi.

També podem forçar-lo, es a dir que es sincronitzi:

```
chronyc -a 'burst 4/4'
```

```
[root@localhost ~]# chronyc -a 'burst 4/4'
200 OK
```

**22. \*Offtopic\*** Aquesta funcionalitat no la suporten tots els rellotges de hardware.

Heu de trobat al directori un fitxer que representa el "l'alarma de despertar-se" del rellotge de hardware RTC.

Un cop trobat envieu-li uns quants segons, per exemple 20:

```
...
echo +20 > ...
...
```

**23.** Amb la següent ordre heu de trobar la línia on diu que es copia l'hora del hardware al sistema.

Fa referència a l'arrencada del sistema

```
...
journalctl -b | grep -i rtc
o també aquesta altra ordre:
dmesg | grep -i rtc
...
```

```
[root@localhost ~]# journalctl -b | grep -i rtc
Dec 04 19:28:20 localhost.localdomain kernel: RTC time: 18:28:19,
date: 12/04/17
Dec 04 19:28:20 localhost.localdomain kernel: platform rtc_cmos:
registered platform RTC device (no PNP device found)
Dec 04 19:28:20 localhost.localdomain kernel: rtc_cmos rtc_cmos: rtc
core: registered rtc_cmos as rtc0
Dec 04 19:28:20 localhost.localdomain kernel: rtc_cmos rtc_cmos:
alarms up to one day, 114 bytes nvram
Dec 04 19:28:20 localhost.localdomain kernel: rtc_cmos rtc_cmos:
setting system clock to 2017-12-04 18:28:20 UTC (1512412100)
Dec 04 19:28:33 localhost.localdomain chronyd[742]: chronyd version
2.3 starting (+CMDMON +NTP +REFCLOCK +RTC +PRIVDROP +SCFILTER
+SECHASH +ASYNCDNS +IPV6 +DEBUG)
Dec 04 19:29:40 localhost.localdomain org.gnome.Shell.desktop[1252]:
(gnome-shell:1252): Gjs-WARNING **: JS ERROR: could not get remote
objects for service org.gnome.SettingsDaemon.Smartcard path
/org/gnome/SettingsDaemon/Smartcard: Gio.DBusError:
GDBus.Error:org.freedesktop.DBus.Error.ServiceUnknown: The name
org.gnome.SettingsDaemon.Smartcard was not provided by any .service
files
Dec 04 19:30:54 localhost.localdomain org.gnome.Shell.desktop[1252]:
(gnome-shell:1252): Cog1-WARNING **: Failed to set crtc mode
```



944x960: Permission denied

Dec 04 19:31:02 localhost.localdomain org.gnome.Shell.desktop[1622]:  
(gnome-shell:1622): Gjs-WARNING \*\*: JS ERROR: could not get remote  
objects for service org.gnome.SettingsDaemon.Smartcard path  
/org/gnome/SettingsDaemon/Smartcard: Gio.DBusError:  
GDBus.Error:org.freedesktop.DBus.Error.ServiceUnknown: The name  
org.gnome.SettingsDaemon.Smartcard was not provided by any .service  
files

Dec 04 19:35:16 localhost.localdomain systemd[1]: Starting System  
clock and RTC settings service...

Dec 04 19:35:16 localhost.localdomain systemd[1]: Started System  
clock and RTC settings service.

Dec 04 19:38:05 localhost.localdomain chronyd[2411]: chronyd version  
2.3 starting (+CMDMON +NTP +REFCLOCK +RTC +PRIVDROP +SCFILTER  
+SECHASH +ASYNCDNS +IPV6 +DEBUG)

Dec 04 19:38:24 localhost.localdomain chronyd[2432]: chronyd version  
2.3 starting (+CMDMON +NTP +REFCLOCK +RTC +PRIVDROP +SCFILTER  
+SECHASH +ASYNCDNS +IPV6 +DEBUG)

Dec 04 19:38:26 localhost.localdomain chronyd[2436]: chronyd version  
2.3 starting (+CMDMON +NTP +REFCLOCK +RTC +PRIVDROP +SCFILTER  
+SECHASH +ASYNCDNS +IPV6 +DEBUG)