

## Fecha, hora y tiempo en Linux

Alberto Ruiz – IES Virgen de la Paloma

## Órdenes relacionadas con fecha, hora y tiempo

- date
- sleep
- time
- at
- cron

### date

- Sirve para mostrar la fecha y la hora actual en distintos formatos
- También sirve para modificarla (sólo puede hacerlo el administrador)

```
[root@localhost ~]# date
sáb abr  5 15:58:07 CEST 2008
[root@localhost ~]# █
```

- Podemos poner la fecha en un archivo:

```
[root@localhost ~]# date > hola.txt
[root@localhost ~]# more hola.txt
sáb abr  5 16:11:03 CEST 2008
```

### date

- Personalizando el formato de salida:  
date +FORMATO
- donde formato puede ser una combinación de opciones:
  - %a: nombre corto del día de la semana
  - %m: número del mes
  - %B: mes completo
  - %y: últimos dígitos del año

```
[root@localhost ~]# date +%a%m%B%y
mar05mayo08
```

### modificadores en date

- Modificadores:
  - ^ para usar mayúsculas
  - \_\_ (guión bajo) para sustituir por espacios los ceros por la izquierda
  - (guión) para suprimir los ceros por la izquierda

```
[root@localhost ~]# date +%a%m%B%y
mar05mayo08
[root@localhost ~]# date +%a%_m%^B%-y
mar 5MAY08
```

### añadir texto en date

- También se puede poner texto o símbolos entre valores
- Lo escribiremos entre comillas (simples o dobles) para poder poner espacios
- %n, %t: salto de línea y tabulador

```
[root@localhost ~]# date +%a,%m' de '%B/%y
mar,05 de mayo/08
```

## formato con date

- Día de la semana:
  - %a: nombre corto (lun)
  - %A: nombre completo (lunes)
- Día del mes:
  - %d: día del mes
- Mes:
  - %b: nombre corto (ene)
  - %B: nombre completo (enero)
  - %m: número (01)
- Año:
  - %y: corto (08)
  - %Y: completo (2008)

## formato con date

- Horas, minutos y segundos:
  - %H: hora 24h (00-23)
  - %l: hora 12h (01-12)
  - %M: minutos (00-59)
  - %S: segundos (00-59)
- Atajos:
  - %D: equivalente a %m/%d/%y (primero el mes)
  - %T: equivalente a %H:%M:%S

## ejercicios date

Indica las órdenes necesarias para mostrar la fecha y la hora actual en los siguientes formatos:

- 20 de mayo de 2008, 19:24:05
- MARTES 20 DE MAYO DEL 08 A LAS 7 Y 24
- 19:24:5-20/5/8
- mar,20,may,08

## modificar la fecha y la hora

- Se hace con la opción **-s** indicando los nuevos datos en un String con este formato:  
"mm/dd/aaaa hh:mm:ss"
- Podemos poner sólo la fecha o sólo la hora
- Antes de arriesgarnos, podemos comprobar si interpreta bien el String con la opción **-d**

```
[root@localhost ~]# date -d "05/06/2007 14:25:00"  
dom may 6 14:25:00 CEST 2007
```

## modificar la fecha y la hora

- Aunque cambiemos la hora, en el próximo reinicio volverá a coger la hora de la BIOS
- Para ver la hora de la BIOS: **hwclock**
- Para modificarla:

```
hwclock --set --date "05/06/2008 14:25:00"
```

- Al ser máquina virtual, no afectará realmente a nuestro equipo

## date

- También sirve para averiguar la última modificación de un archivo:

```
[root@localhost ~]# date -r carta.txt  
sáb abr 5 16:10:40 CEST 2008
```

## sleep

- Lo hemos utilizado al estudiar las órdenes relativas a procesos
- Simplemente se duerme durante un número de segundos
- Si detrás del número se añade “m”, “h” o “d”, dormirá durante minutos, horas o días
- Ejemplos:

```
sleep 200
sleep 3h
sleep 1d
```

## time

- Monitoriza el tiempo de ejecución de una orden

```
[root@localhost ~]# time tar -xf gftp-2.0.18.tar
```

```
real    0m0.126s
user    0m0.005s
sys     0m0.001s
```

- real: tiempo que ha pasado desde que lanzaste la orden hasta que terminó
- user: tiempo que la CPU pasa ejecutando tu orden
- sys: tiempo que la CPU pasa ejecutando código de sistema (kernel) para poder ejecutar tu orden

## ejercicios time

- Copia la orden y los tiempos obtenidos al ejecutar...
  - la descompresión de un archivo
  - la búsqueda de un archivo en un directorio
  - un proceso que duerma durante 10 segundos
  - una aplicación (navegador, editor de textos)

```
[root@localhost /]# time find /etc/ -name httpd.conf
/etc/httpd/conf/httpd.conf
```

```
real    0m0.083s
user    0m0.004s
sys     0m0.022s
```

## at

- Sirve para ejecutar una o varias órdenes orden en un momento determinado
- ¿Cómo se le dice cuándo ejecutar?
  - now (ahora), today (hoy), tomorrow (mañana), midnight (medianoche), noon (mediodía), ...
  - horas y fechas completas
  - combinaciones: + 1 minutes | hours | weeks...

```
at 19:25 today
at now + 1 minute
at today + 3 days
at 12/31/08
```

## at

- Nos sale un cursor y podemos introducir una serie de órdenes
- Al final, pulsamos Ctrl + D
- Nos dice la fecha en que se ejecutará, y un número de trabajo

```
[root@localhost /]# at now + 1 minute
at> cp carta.txt carta.txt.seg
at> cp doc.txt doc.txt.seg
at> <EOT>
job 10 at Tue May 20 15:49:00 2008
```

## at

- Importante: “at” ejecuta las tareas en segundo plano
- No tendremos confirmación por consola de lo que está ocurriendo
- Si planeamos un sleep 60, y una vez comenzado ponemos “ps -e”, podremos ver el proceso

## atq y atrm

- atq: muestra los trabajos programados por el usuario
  - Si lo ejecuta el administrador, se muestran todos los trabajos programados
- atrm: permite eliminar un trabajo

```
[root@localhost ~]# atq
12 Tue May 20 20:39:00 2008 a root
11 Tue May 20 19:00:00 2008 a root
13 Sat May 24 16:40:00 2008 a root
[root@localhost ~]# atrm 11
[root@localhost ~]# atq
12 Tue May 20 20:39:00 2008 a root
13 Sat May 24 16:40:00 2008 a root
```

## ejercicios at

- Copia las órdenes necesarias para programar un evento que ocurra ...
  - dentro de un minuto
  - dentro de tres días
  - mañana a las 14:30
  - el 30 de noviembre a las 17:05

## cron

- Es un demonio que se dedica a gestionar trabajos programados
- Para programar un trabajo hay que escribirlo en la “crontab” del usuario
- Antes de empezar, fijaremos la variable de entorno EDITOR si no lo hemos hecho:

```
[root@localhost ~]# echo $EDITOR
[root@localhost ~]# export EDITOR=/usr/bin/gedit
[root@localhost ~]# echo $EDITOR
/usr/bin/gedit
```

## crontab

Las opciones de crontab:

- -l : mostrar nuestro crontab
- -e : editar nuestro crontab
- -r : eliminar nuestro crontab
- -u juan -x : actuar sobre el crontab de juan (x puede ser l, e, r)
- Prueba a listar tu crontab

## crontab

- Editando el crontab: crontab -e
- Se abre el editor. Cuando lo cerremos, el crontab que hemos escrito se instalará
- El formato de las líneas es:  
min hor numdía mes díasemana acción
- Ejemplo:  
00 14 1 \* \* cp a.log a.log.bak  
“Haz la copia a las 14:00 del día 1 de cada mes”

## crontab

- Modificadores:
  - \* : cualquier valor
  - /x : cada x [días, horas, minutos...]
  - x-y : entre x e y (rango)
  - x,y : x e y (enumeración)
- Ejemplo:  
*min hor numdía mes díasemana acción*  
1,15 \*/2 \* \* 1-5 accion  
“Ejecuta ‘acción’ todos los meses de lunes a viernes, cada dos horas, en los minutos 1 y 15”

## crontab

- Uso de crontab:
- Normalmente se escriben 'scripts' en un archivo que hagan algo (copias de seguridad, limpieza de logs, generar estadísticas del servidor web, ...)
- En la línea de crontab no se pone una orden, sino la ruta del script

- Ejemplo:

```
* /10 * * * * /usr/bin/fetchmail
```

*"Consulta si hay correo nuevo cada 10 minutos"*

## ejercicios cron

- Copia las líneas que debes poner en tu crontab para que se ejecute el script `/usr/bin/miscript ...`
  - Todos los días a las 13:30 y a las 23:30
  - De lunes a viernes a las 12:05
  - Cada dos horas los días 15 de enero y junio