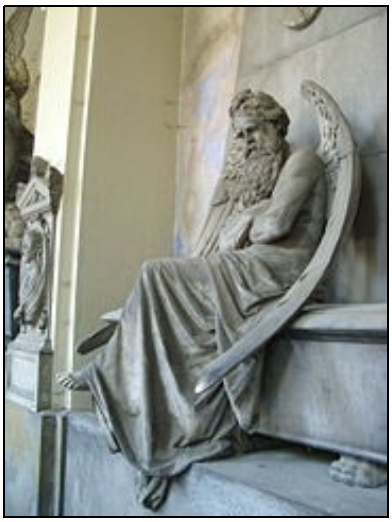
1 Planificador de tarefas: cron



Chronos esperando no Cemiterio Monumental Xenovés de Staglieno.
Para programar/planificar tarefas con certa periodicidade ou en certas datas en GNU/Linux úsase o servizo cron. O nome de **cron** vén do grego Chronos.

1.1 Sumario

1.2 Introdución

- **Cron** é un servizo que corre en segundo plano e que executa tarefas cada certo tempo (*/etc/init.d/cron*).

- **Cron** le ficheiros chamados **crontab** (CRON TABle), nos que se almacenan as tarefas a executar e cando se deben executar.

- Cada usuario pode ter o seu propio ficheiro **crontab**, onde pode especificar as accións que desexa levar a cabo e as veces que deberían ser executadas. Ollo!!! as accións que se especifiquen levaranse a acabo se o usuario ten permisos para realizalas e se ten permitido ter un ficheiro **crontab** (Por defecto todo usuario ten dereito).
    - Se o usuario non está logueado realizaranse igualmente as tarefas especificadas no seu ficheiro **crontab**.
    - Os ficheiros **crontab** de cada usuario almacénase en /**var**/**spool**/**cron**/**crontabs**/**<nome de usuario>** .
    - Existe, tamén, un ficheiro **crontab do sistema**, este almacénase en /**etc**/**crontab**. Nel, ao igual que nos outros, indícanse as tarefas a realizar, cando executalas e **usuario** que as debe executar.

- **Os usuarios que poden, ou non, executar tarefas programadas están nas seguintes listas:**

- **/etc/cron.allow** (permitir)
- **/etc/cron.deny** (denegar)
- Estes dous ficheiros non existen inicialmente nun sistema Debian, por tanto todo usuario pode executar tarefas programadas. Tan pronto como exista un deses ficheiro aplícase a permisión/denegación:
    - ◊ Existe /etc/cron.allow e non /etc/cron.deny: só poden executar cron os usuarios que estean no ficheiro cron.allow.
    - ◊ Existe /etc/cron.deny e non /etc/cron.allow: só poden executar cron os usuarios que NON estean no ficheiro cron.deny.
    - ◊ Non existe ningún: todo usuario pode executar cron.

- No directorio **/etc/cron.d/** pódese almacenar ficheiros tipo crontab co mesmo formato quen ten o ficheiro /etc/crontab. É máis aconsellable crear crontabs en /etc/cron.d/ que modificar o ficheiro /etc/crontab, porque este pode ser modificado cando se realicen actualizacións de cron.

- Se está instalada a utilidade **anacron** executaranse tódalas tarefas almacenadas en: */etc/cron.hourly*, */etc/cron.dayly*, */etc/cron.weekly* e */etc/cron.monthly*. A diferenza de **cron**, se o ordenador estivo a apagado, ao acender mira se algunha tarefa deses directorios non foi executada no seu momento e procede con ela.

- O servizo de *cron* cada minuto revisa o ficheiro /etc/crontab e o contido das carpetas /var/spool/cron/crontabs, /etc/cron.hourly, /etc/cron.dayly, /etc/cron.weekly, /etc/cron.monthly e /etc/cron.d e se hai cambios, estes son cargados en memoria. Por iso, non é preciso reiniciar o servizo de *cron* cada vez que se modifique un ficheiro *crontab*.

TAMÉN PODES VER...

Pódese atopar axuda en:

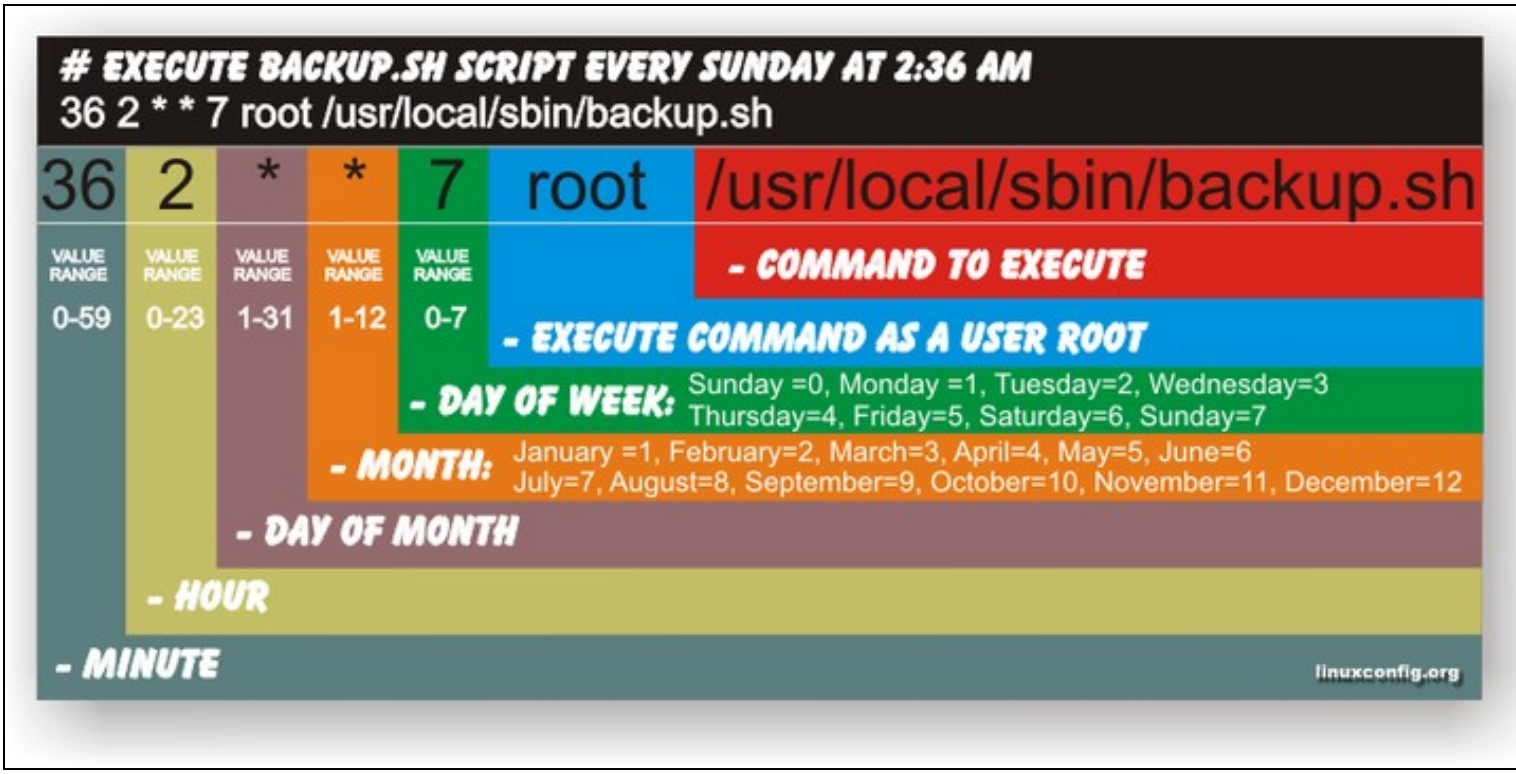
- man cron
- man crontab (axuda do comando).
- man 5 crontab (axuda do formato do ficheiro).

1.3 Estrutura dos ficheiros crontab

- Cada ficheiro **crontab**, ben do usuario ou ben do sistema, ten a maior parte das veces unha estrutura interna como a que segue:

```
# m h dom mon dow user    command
m:   minutos             0-59
h:   hora                0-23
dom: día do mes          1-31
mon: mes                 1-12 (ou 3 primeiras letras do nome)
dow: día da semana:      0-7  (0 ou 7 é Domingo,  ou 3 primeiras letras do nome)
user: usuario que executa o comando: non existe nos crontab de usuario (/var/spool/cron/crontabs/<nome usuario>).
Si existe no crontab do sistema (/etc/crontab).
command: comando a executar.
```

A seguinte imaxe pode clarificar un chisco máis a estrutura e os posibles valores:



Imaxe tomada de http://www.linuxconfig.org

- Neste caso, esta sería a configuración do **crontab do sistema** (/etc/crontab), pois ten o campo **usuario**.

- Notar que cando se indica un asterisco (*) o campo toma todos os valores do rango, do primeiro ao último.

- Están permitidas listas (usar comas ,), rangos (usar guións -) ou saltos (usar barra /) de números en cada un dos campos. Exemplos para o campo minutos:
  - **1,5,7,20:** a tarefa executarase neses minutos.
  - **0-5:** a tarefa executarase cada minuto, dende o minuto 0 ate o 5, ambos inclusive.
  - **0-6, 10-13,40:** combinación dos 2 anteriores.
  - ***/15:** a tarefa executarase cada 15 minutos.
  - **0,15,30,45:** fai o mesmo que o anterior.
  - **0-11/2:** a tarefa executarase cada 2 minutos dende o minuto 0 ó 10 (incluídos).
  - **0,2,4,6,8,10:** fai o mesmo que anterior liña.

- Observar que o campo día está determinado por dous campos: dom (día do mes) e dow (día da semana). Se os dous teñen valor, o comando executarase cando CALQUERA dos campos coincidan co momento correcto. Exemplo:

```
# m  h dom  mon dow command
  30 4 1,15  *   5  comando
```

O comando executárase ás 4:30 am, os días 1 e 15 de cada mes ou calquera venres á mesma hora.

- No canto dos cinco primeiros campos pode aparecer unha das 8 cadeas especiais:

```
cadea      significado                                m h dom mon down
------     -----------------------------------------------------------
```

```
@reboot    Executar unha vez ó iniciar.
@yearly    Executar unha vez ó ano, o 01/01/ano.=    ?0 0 1   1   *".
@annually  O mesmo que @yearly.
@monthly   Executar unha vez ó mes, o 01/mes/ano.=   "0 0 1   *   *".
@weekly    Executar unha vez á semana, o Domingo.=   "0 0 *   *   0".
@daily     Executar unha vez ó día, ás 12 da noite.= "0 0 *   *   *".
@midnight  O mesmo que @daily.
@hourly    Executar cada hora en punto.=             "0 * *   *   *".
```

1.3.1 Envío de correo

- **cron** envía ao usuario dono da tarefa (root@localhost, usuario@localhost) que se executa programadamente un correo coa saída da tarefa ou dos comandos da tarefa, se teñen. Para que se poida facer o envío do correo hai que ter instalada unha utilidade de mail.

- *cron* envia o correo ao dono da tarefa, coa saída desta, ou a quen se especifique no parámetro **MAILTO** do ficheiro *crontab*.

- Que facer no caso de ter unha utilidade de mail e non se desexa que cron estea enviando unha mensaxe de correo coa saída da tarefa cada vez que se executa?:
    - **comando &> /dev/null** ou **comando > /dev/null 2>&1**: envía a saída do comando/erros a /dev/null.
    - **comando >> ruta/saida.txt 2>&1**: envía a saída do comando/erros ó ficheiro saida.txt
    - **MAILTO=??**: definir o parámetro do ficheiro **crontab** a nada, e enviaráselle o correo á **ninguén**.

1.4 Variables de entorno

- Algunhas variables de entorno son iniciadas polo servizo *cron*. Cando se lanza unha tarefa programada estes son os seus valores por defecto.
    - **HOME**= colle o valor do ficheiro */etc/passwd* do dono do crontab.
    - **LOGNAME**= colle o valor do ficheiro */etc/passwd* do dono do crontab.
    - **PATH**=/usr/bin:/bin
    - **SHELL**=/bin/sh

Estes valores poden ser modificados ao inicio dun ficheiro crontab. E poden ser definidas outras novas variables, como: *LANG*, que por defecto é inglés. Se se desexase que a *locale* estivese en galego, indicar ao inicio do crontab:

```
LANG=gl_ES.UTF-8
```

1.4.1 Exemplos de ficheiros crontab

Antes de entrar en materia e realizar tarefas programadas véxanse algúns exemplos destas. Ao final destes exemplos creranse varias tarefas programadas.

- **Exemplo de Ficheiro crontab dun usuario: /var/spool/cron/crontabs/<usuario>**:

```
# Un ficheiro de usuario debe ser editado co comando crontab, logo verase.

 #Usar /bin/bash para executar os comandos, no canto do shell por defecto: /bin/sh
 SHELL=/bin/bash
 #
 # Enviar un correo, se estivera configurado, coa saída dos comandos a tres
 # usuarios, o último é un usuario local. Non importa que sexa o dono do crontab
 MAILTO=usuario@edu.xunta.es, user@gmail.com, usuario_local


 # Executa o script '"diario.job"', tódolos días, cada 5 minutos, despois da media
 # noite. A súa saída non é enviada ó correo senón que se envía ó ficheiro
 # saida.txt
```

```
  5 0 * * *       $HOME/bin/diario.job >> $HOME/saida.txt 2>&1


# Execútase a tarefa monthly ás 2:15pm do primeiro de cada mes, e envíase un
# correo coa súa saída ós tres destinatarios de MAILTO.
  15 14 1 * *     $HOME/bin/monthly


#
# Enviar un corre a xose as 10 pm ós días de semana:
  0 22 * * 1-5    mail -s "Son as 10pm" xose%Xose,%%Onde están os nenos?%


# Envía a saída do comando echo ós tres de MAILTO, tódolos días, cada 2 horas,
# 23 minutos despois da hora.
 23 0-23/2 * * * echo "run 23 minutes after midn, 2am, 4am ..., everyday"


# Envía a saída do comando echo ós tres de MAILTO, cada domingo ás 4:05 am.
# 23 minutos despois da hora.
  5 4 * * sun     echo "run at 5 after 4 every sunday"


# Envía o erro da saída do comando mkdir /u  ós tres de MAILTO, cada minuto.
# Un usuario calquera non ten, por defecto, permisos para crear na raíz.
# A mensaxe que enviaría sería algo como isto:
#         mkdir: non se pode crear o directorio "/u": Permission denied
   * * * * *  mkdir /u
```

- **Exemplo de Ficheiro crontab de sistema: /etc/crontab**:

```
# Este ficheiro, ao igual que os crontabs que se creen dentro de /etc/cron.d/,
 # ten un campo de usuario, que indica que usuario executa ese comando.

 # Este ficheiro ou os crontabs de /etc/cron.d/ edítanse á man sen usuar o
 # comando crontab, como se verá máis adiante.

 # Non se recomenda que se modifique o ficheiro /etc/crontab, pois se se
 # actualiza o cron, pode ser que se actualice este ficheiro. Para tarefas
 # programadas do sistema créense ficheiros crontabs en /etc/cron.d/

 # Este é o contido real de /etc/crontab.
 #Definición de variables de entorno.
 SHELL=/bin/sh
 PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

 # run-parts: executa programas ou scripts dun directorio.
 # m h dom mon dow user command

   17 * * * *   root         cd / && run-parts --report /etc/cron.hourly
 # Comproba se existe o comando '''anacron''' run
   25 6 * * *   root   test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
   47 6 * *7   root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
   52 61 * *   root     test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )


 # A modo de exemplo vaise engadir unha entrada neste ficheiro, cousa non
 # aconsellable, é mellor crear un ficheiro tipo crontab en /etc/cron.d/

 MAILTO=usuario@edu.xunta.es, user@gmail.com, usuario_local

 # Envía a saída do comando (o espazo que consumen as carpetas persoais dos
 # usuarios LDAP) ós tres de MAILTO. Ás 2am de cada día.
   * 2 * * * root du -h --max-depth=1 /home/iescalquera
```

1.5 Crear tarefas programadas: liña comandos

Vistos os exemplos anteriores vanse realizar varias tarefas programadas en *dserver00*.

1.5.1 Tarefas programadas de usuarios

Usar o comando **crontab**:

```
crontab [-u usuario] { -e | -l | -r }

# Parámetros
# -u: para editar o crontab dun usuario distinto ó actual.
# -e: edita o ficheiro crontab de usuario
# -l: lista o ficheiro crontab de usuario
# -r: borra o ficheiro crontab de usuarios
```

- **A miña primeira tarefa programada:**

Crear unha tarefa programada que cada minuto escriba nun ficheiro o nome do usuario, o seu *home*, e a data completa.

- En **dserver00**, entramos co usuario administrador ou nos pasamos a el dende root

```
su - administrador
```

Editamos o ficheiro crontab do usuario *administrador*. Non hai en */var/spool/cron/crontabs* ningún ficheiro crontab para o administrador así que o comando vai crear un ficheiro novo automaticamente usando o comando *nano*:

```
administrador@dserver00:~$ crontab -e
```

Na tarefa programada introducir:

```
# m h  dom mon dow   command
  * *   *   *   *    echo "$LOGNAME, $HOME " `date`>>ficheiro.txt
```

- Notar que ao ficheiro non se lle puxo a ruta. A ruta que tomará por defecto é o home do usuario, pois esa tarefa execútase no contorno do usuario administrador. Sería equivalente a escribir *$HOME/ficheiro.txt*.

- En **/var/spool/cron/crontabs/administrador** foi onde se gardou o contido da tarefa programada do usuario administrador. Podemos vela usando o usuario *root*:

```
root@dserver00:/home/administrador# ls /var/spool/cron/crontabs/ -l
total 4
-rw------- 1 administrador crontab 1153 Mai  9 00:59 administrador
```

- Pasados uns minutos:

```
administrador@dserver00:~$ cat ficheiro.txt
administrador, /home/administrador  Ven Mai 9 01:00:01 CEST 2014
administrador, /home/administrador  Ven Mai 9 01:01:01 CEST 2014
administrador, /home/administrador  Ven Mai 9 01:02:01 CEST 2014
administrador, /home/administrador  Ven Mai 9 01:03:01 CEST 2014

administrador@dserver00:~$ date
Ven Mai  9 01:04:09 CEST 2014
```

- Notar como o comando date executado na tarefa programada pon o día en galego. Imos configurar un idioma diferente.

- Configurar a variable de entorno na tarefa programada.

```
administrador@dserver00:~$ crontab -e
```

Engadir: LANG=gl_ES.UTF-8 (revisar o comando *locale*).

```
LANG=es_ES.UTF-8

# m h  dom mon dow   command
  * *   *   *   *    echo "$LOGNAME, $HOME " `date`>>ficheiro.txt
```

- Pasados uns minutos, observar a última liña.

```
administrador@dserver00:~$ cat ficheiro.txt
administrador, /home/administrador  Ven Mai 9 01:00:01 CEST 2014
administrador, /home/administrador  Ven Mai 9 01:01:01 CEST 2014
administrador, /home/administrador  Ven Mai 9 01:02:01 CEST 2014
administrador, /home/administrador  Ven Mai 9 01:03:01 CEST 2014
administrador, /home/administrador  Ven Mai 9 01:04:01 CEST 2014
administrador, /home/administrador  vie may 9 01:05:01 CEST 2014
```

- **A segunda tarefa programada: script**

Trátase de facer o mesmo que no caso anterior, pero que o comando a executar sexa un script:

```
nano diahora.sh
```

- Editar o seu contido:

```
#!/bin/bash
echo "$LOGNAME, $HOME " `date`>>ficheiro2.txt
```

- Editar o crontab de administrador e modificar o seu contido:

```
crontab -e
```

- Editar contido:

```
LANG=es_ES.UTF-8

# m h  dom mon dow   command
 * *   *   *   *    sh diahora.sh # Segunda tarefa
```

- Pasados uns minutos:

```
administrador@dserver00:~$ cat ficheiro2.txt
administrador, /home/administrador  vie may 9 01:16:01 CEST 2014
administrador, /home/administrador  vie may 9 01:17:01 CEST 2014
```

- **Terceira tarefa programada**

Quérese recibir un correo cada minuto coa mesma información anterior e ademais co contido de /root.

- Crear un script:

```
nano diahora_mail.sh
```

- Contido:

```
#!/bin/bash
echo "$LOGNAME, $HOME " `date`
```

```
ls /root
```

- Observar que as saídas dos comandos non se redireccionan a ningures. Ademais o usuario administrador non ten permisos para ver o contido de /root.

- Editar o crontab de administrador e modificar o seu contido:

```
crontab -e
```

- Editar contido:

```
LANG=gl_ES.UTF-8

# m h  dom mon dow   command
 * *   *   *   *    sh diahora_mail.sh # Terceira tarefa
```

- Pasado un minuto, consultamos o *mail* do usuario administrador co comando **mail**:

```
administrador@dserver00:~$ mail
Mail version 8.1.2 01/15/2001.  Type ? for help.
"/var/mail/administrador": 1 message 1 new
>N  1 root@derver00.ies  Fri May 09 01:21   23/1059  Cron <administrador@dserver
& 1
Message 1:
From administrador@derver00.iescalquera.local Fri May 09 01:21:02 2014
Envelope-to: administrador@derver00.iescalquera.local
Delivery-date: Fri, 09 May 2014 01:21:02 +0200
From: root@derver00.iescalquera.local (Cron Daemon)
To: administrador@derver00.iescalquera.local
Subject: Cron <administrador@dserver00> sh diahora_mail.sh
Content-Type: text/plain; charset=UTF-8
X-Cron-Env: <LANG=es_ES.UTF-8>
X-Cron-Env: <SHELL=/bin/sh>
X-Cron-Env: <HOME=/home/administrador>
X-Cron-Env: <PATH=/usr/bin:/bin>
X-Cron-Env: <LOGNAME=administrador>
Date: Fri, 09 May 2014 01:21:02 +0200

administrador, /home/administrador  vie may 9 01:21:01 CEST 2014
ls: no se puede abrir el directorio /root: Permiso denegado

&q
```

- Observar como amosa a saída dos dous comandos:
    - ♦ echo, saída correcta
    - ♦ ls /root: saída con erros, porque o usuario non pode acceder a /root.

1.5.2 Editar tarefas programadas doutros usuarios

Só o pode facer o administrador do sistema (root ou pertencentes a sudoers).

```
root@dserver00:~# crontab -u sol -e
no crontab for sol - using an empty one
crontab: installing new crontab
root@dserver00:~# ls /var/spool/cron/crontabs/ -l
total 8
-rw------- 1 administrador crontab 1145 Mai  9 01:26 administrador
-rw------- 1 sol           crontab 1153 Mai  9 01:27 sol
```

1.6 Borrar tarefas programadas de usuarios

```
root@dserver00:~# crontab -u sol -r
root@dserver00:~#
root@dserver00:~# ls /var/spool/cron/crontabs/ -l
total 4
```

```
-rw------- 1 administrador crontab 1145 Mai  9 01:26 administrador
```

1.7 Tarefas programadas de sistema

Estas poden ser creadas cun editor calquera, modificando o ficheiro /**etc**/**crontab** ou creando un ficheiro crontab no directorio /**etc**/**cron.d**/. Isto último é o aconsellable. Lembrar que hai que especificar o usuario.

Se se desexa que se execute un script, este debe residir nun sitio accesible para o usuario que o executa.

Exemplo de entradas en /etc/crontab ou /etc/cron.d/tarefa

```
nano /etc/cron.d/tarefa
```

- O contido

```
# m  h  dom mon dow user            command
  *  *   *   *   *   administrador  echo ola>>ficheiro4.txt
  30 23  *   *   *   root           /usr/local/sbin/backup.sh
  @reboot            root           ntpdate es.pool.ntp.org
```

-- Antonio de Andrés Lema e Carlos Carrión Álvarez