# Cron (Unix)

En el sistema operativo Unix, **cron** es un administrador regular de procesos en segundo plano (*demonio*) que ejecuta procesos o guiones a intervalos regulares (por ejemplo, cada minuto, día, semana o mes). Los procesos que deben ejecutarse y la hora en la que deben hacerlo se especifican en el fichero `crontab`. El nombre *cron* viene del griego *chronos* (χρόνος) que significa "tiempo".

**Cron** se podría definir como el "equivalente" a Tareas Programadas de Windows.

## Vista general

Cron es impulsado por un *crond*, un archivo de configuración que específica comando shell para ejecutarse periódicamente a una hora específica. Los archivos **crontab** son almacenados en donde permanecen las listas de trabajos y otras instrucciones para el demonio cron. Los usuarios habilitados para crear su fichero **crontab** se especifican en el fichero **cron.allow**. De manera análoga, los que no lo tienen permitido figuran en cron.deny. Estos dos últimos ficheros se encuentran en **/etc/cron.d/**, o **/etc/**, dependiendo de la versión de Unix.

Cada línea de un archivo crontab representa un trabajo y es compuesto por una expresión CRON, seguida por un comando shell para ejecutarse. Algunas implementaciones de cron, tal como en la popular BSD 4a edición escrita por Paul Vixie, e incluido en muchas distribuciones Linux, agrega una especificación de nombre de usuario dentro del formato como un sexto campo, como quién ejecutará el trabajo especificado (sujeto a la existencia de un usuario en */etc/passwd* y permisos autorizados). Esto solo es permitido en el sistema crontab (/etc/crontab y /etc/cron.d/*), no en otros donde son asignados cada usuario es asignado a una configuración.

Para el "día de la semana" (campo 5), ambos 0 y 7, son considerados Domingo, a través de algunas versiones de Unix tal como AIX no toma como vàlido el "7" en el man. Mientras que cuando el trabajo es ejecutado normalmente cuando fueron especificados los campos tiempo/fecha, todos coinciden con la hora y fecha actual, esto es una excepción.

**Si** tanto el "día del mes" como "día de la semana" son restringidos (no son **'' * ''** ), entonces **o** el "día del mes" (campo 3) **o** el "día de la semana" (campo 5) debe coincidir con el día actual.

## Formato del fichero *crontab*

Fichero crontab de ejemplo:

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01  * * * * root nice -n 19 run-parts /etc/cron.hourly
50  0 * * * root nice -n 19 run-parts /etc/cron.daily
22 4 * * 0 root nice -n 19 run-parts /etc/cron.weekly
42 4 1 * * root nice -n 19 run-parts /etc/cron.monthly
```

Para agregar, quitar o modificar tareas, hay que editar el crontab. Esto se hace con la orden *crontab -e*, que abrirá el editor definido en la variable de entorno **EDITOR** y cargará el fichero **crontab** correspondiente al usuario que está logueado.

Cada vez que se ejecuta el crontab, se envía un mensaje al usuario que aparece en la variable de entorno **MAILTO**, si está habilitado, indicándole la tarea realizada.

A continuación se pone a cero el log de error de Apache un minuto después de medianoche (00:01 de cada día del mes, de cada día de la semana).

```
1 0 * * *  echo -n "" > /www/apache/logs/error_log
```

A continuación se ejecuta el script: /home/user/test.pl cada 5 minutos.

```
*/5 * * * *  /home/user/test.pl
```

```
.---------------- minuto (0 - 59)
|  .------------- hora (0 - 23)
|  |  .---------- día del mes (1 - 31)
|  |  |  .------- mes (1 - 12) O jan,feb,mar,apr ... (los meses en inglés)
|  |  |  |  .---- día de la semana (0 - 6) (Domingo=0 ó 7) O sun,mon,tue,wed,thu,fri,sat (los días en inglés)
|  |  |  |  |
*  *  *  *  *  comando para ser ejecutado
```

Nota: no soporta variables de entorno

## Sintaxis

El formato de configuración de cron es muy sencillo.

- El símbolo almohadilla «#» es un comentario, todo lo que se encuentre después de ese carácter no será ejecutado por *cron*.
- El momento de ejecución se especifica de acuerdo con la siguiente tabla:

1. Minutos: (0-59)
2. Horas: (0-23)
3. Días: (1-31)
4. Mes: (1-12)
5. Día de la semana: (0-6), siendo 1=lunes, 2=martes,... 6=sábado y 0=domingo (a veces también 7=domingo)

```
#############################################################
#minuto (0-59),                                             #
#|  hora (0-23),                                           #
#|  |  día del mes (1-31),                                 #
#|  |  |  mes (1-12),                                       #
#|  |  |  |  día de la semana (0-6 donde 0=Domingo)         #
#|  |  |  |  |       comandos                               #
#############################################################
15 02  *  *  *
```

Para especificar todos los valores posibles de una variable se utiliza un asterisco (*).

- La última columna corresponde a la ruta absoluta del binario o *script* que se quiere ejecutar.

## Ejemplos

Por ejemplo:

```
30 10 * * 1 /usr/bin/who >> /home/quien.tex
```

Ejecuta la orden **who** todos los lunes a las **10:30** y guarda la salida en el fichero *quien.tex*

Para especificar dos o más valores en cada variable, estas deben estar separadas por comas, siguiendo con el ejemplo anterior:

```
0,30 * * * 1 /usr/bin/who >> /home/quien.tex
```

Ejecuta la orden who todos los lunes cada media hora y guarda la salida en el fichero *quien.tex*

Si queremos que se ejecute cada 15 minutos sería

```
0,15,30,45 * * * * /usr/bin/who >> /home/quien.tex
```

o

```
*/15 * * * * /usr/bin/who >> /home/quien.tex
```

En este ejemplo veremos como pasarle más de un comando al cron y de paso como puede programarse una descarga:

```
30 21 * * * cd /media/sda7/dexter/distributions/isos;wget
```
http://example.com/fichero_a_descargar.loquesea"

Este otro es para programar el apagado del PC. En este caso todos los sábados a las 21.30

```
30 21 * * 6 /sbin/shutdown –h now
```

**Editar crontab de un usuario en particular**

crontab [ -u usuario ] fichero

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

La opción -u se utiliza para indicar el crontab de usuario que queremos administrar. Sólo root podrá usar la orden crontab con esta opción.

La opción -e se utiliza para editarlo

## Definición de horarios predefinidos

Hay varios valores predefinidos que se pueden utilizar para sustituir la expresión CRON.

| Entrada | Descripción | Equivale A |
|---|---|---|
| `@yearly` | Se ejecuta una vez al año | `0 0 1 1 *` |
| `@annually` | (igual que `@yearly`) | `0 0 1 1 *` |
| `@monthly` | Se ejecuta una vez al mes | `0 0 1 * *` |
| `@weekly` | Se ejecuta una vez a la semana | `0 0 * * 0` |
| `@daily` | Se ejecuta una vez al día | `0 0 * * *` |
| `@midnight` | (igual que `@daily`) | `0 0 * * *` |
| `@hourly` | Se ejecuta una vez cada hora | `0 * * * *` |

También esta disponible `@reboot`, que permite a un trabajo ejecutarse una vez cada vez que el demonio cron se inicie, que eso tipìcamente coincidirá con el arranque del servidor. Puede ser útil si es necesario levantar un servidor o demonio bajo un usuario en particular o si el usuario no tiene permisos al archivo *rc.d/init.d*.

### Manipulación de zonas horarias

Muchas implementaciones de cron simplemente interpretan entradas crontab en la configuración del sistema de zona horaria en virtud de lo que establece el demonio al ejecutarse. Esto puede ser el origen de conflictos si grandes equipos multiusuarios tienen usuarios en varias zonas horarias, especialmente si el sistema de zona horaria predefinida incluye el confusor potencial DST. Por lo tanto una implementación cron puede tener cualquier caso especial "TZ=<timezone>" variable de entorno ajustando líneas en el usuario crontab, interpretando entradas relativas crontab posteriores a la zona horaria.[1]

## Historia

### Primeras versiones

Cron en Unix versión7, escrita por Brian Kernighan, fue un sistema de servicio (después llamado demonio) invocado de `/etc/inittab` cuando el S.O. entró en modo multiusuario. Este algoritmo fue sencillo:

1. Leer `/usr/etc/crontab`
2. Determinar si algún comando se estaba ejecutando en la fecha y hora actual, y si se ejecutaba como Root.
3. Suspender por un minuto
4. Repetir paso 1.

Esta versión de cron fue básica y robusta, pero también consumió recursos si encontraba algún trabajo que hacer o no; al escuchar esta descripción, Douglas Comer, un profesor de la Universidad Purdue, remarcó, "Ah, un algorítmo de poca memoria.". En un experimento en la Universidad Purdue pasados los 1970s para extender los servicios de cron a todos los 100 usuarios sobre un tiempo compartido VAX que fue situado en un lugar de mucha carga del sistema.

### Capacidad multi-usuario

La siguiente versión de cron, con la liberación System V, fue creada para extender las capacidades de cron para todos los usuarios de un sistema Unix, no solo root (o superusuario). Aunque esto hoy puede ser trivial con la mayoría de Unix y sistemas tipo Unix con procesadores potentes y un número pequeño de usuarios, al momento que requiere un nuevo enfoque sobre un sistema 1MIPS teniendo alrededor de 100 cuentas de usuario.

### Versiones modernas

Con la ventaja del proyecto GNU y Linux, apareció un nuevo cron. Lo más relevante de este es el cron Vixie, originalmente codificado por Paul Vixie en 1987. La versión 3 de **cron Vixie** fue liberada después de 1993. La versión 4.1 fue renombrada como **Cron ISC (Consorcio de Sistema de Internet)** y fue liberada en enero del 2004. La versión 3, con unas mínimas correcciones de error, es usada en la mayoría de las distribuciones de GNU/Linux y BSD.

En 2007, RedHat bifurcó cron-vixie 4.1 al proyecto cronie e incluyó anacron 2.3 en 2009.

Otra implementación popular incluye anacron y fcron. De cualquier forma, anacron no es un programa cron independiente; se basa en otro programa cron para llamarlo en orden para ejecutarse.

## Referencias

[1] Sun.com (http://blogs.sun.com/chrisg/entry/timezone_aware_cron_finally_pushed)

## Enlaces externos

- Guía Gentoo Linux de Cron (http://www.gentoo.org/doc/es/cron-guide.xml) Documentación oficial de Gentoo.
- `cron(8)` (http://www.freebsd.org/cgi/man.cgi?query=cron&sektion=8): demonio para ejecutar comandos regularmente – Administración del sistema en el manual de FreeBSD (en inglés)

# Fuentes y contribuyentes del artículo

**Cron (Unix)** *Fuente*: http://es.wikipedia.org/w/index.php?oldid=67051252 *Contribuyentes*: Abece, Agguizar, Aleivag, Alexav8, Alvk4r, Anarres, Angietigger, Angro, Barcex, Bedwyr, Belianisto, DRoBeR, Digigalos, Echani, Ecoronel, Elwikipedista, Especiales, GermanX, Hari Seldon, Humberto, Kizar, Kurnosem, Lucien leGrey, Muro de Aguas, Ossany, Pino, Ramonserra, Rosarinagazo, Rutrus, Serrador, Shooke, SuperBraulio13, 77 ediciones anónimas

# Licencia