

# ***Cron y Anacron***

*Alumno: Ramón Fernando Pérez*

*Profesor: Juan Gonzales*

*Instituto: Iessonferrer*

*Curso/Año: 1ro de FPS Desarrollo de Aplicaciones Web(DAW)*

# *Indice:*

*Que es Cron.*

*Como funciona.*

*Programación de Horas y min con Cron*

*El uso de run-parts y Cron Directorios*

*Usando Crontab*

*Que es Anacron*

*Como trabajar con Anacron*

*Como trabajar con Anacron*

# Cron

## Que es Cron:

Cron es un programa que te permite asignar tareas para ejecutar en momentos preconfigurados. Cron es una herramienta basica que se puede utilizar para automatizar casi cualquier cosa.

Es un programa experto en la gestion de tareas que deben ser realizadas por horas y dias y es capaz de hacer grandes rutinas que se deben realizar una o dos veces al año.

Cron es una de las herramientas esenciales para un administrador de tareas.

Cron sirve en cualquier distribucion moderna de Linux.

## Como funciona Cron:

Cron se inicia en el menu arranque y se ejecuta en segundo plano. Esto significa que no tiene interaccion con el usuario, cuando se esta ejecutando en segundo plano Cron comprueba su archivo de configuraciones una vez cada minuto para ver si hay algun evento programado para ese minuto, si se programa un evento, Cron ejecuta cualquier programa predeterminado que se le haya dado y luego regresa a un segundo plano, luego espera otros 60 segundos y vuelve a revisar su archivo de configuraciones.

Cron decide que comandos va a ejecutar min a min mediante la lectura de una serie de archivos como "**crontab**".

El comando para acceder es: `/etc/crontab`

Ejemplo: `less /etc/crontab`

```
horabaixa@daw1-17:~$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
t /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
t /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
t /etc/cron.monthly )
#
horabaixa@daw1-17:~$
```

Este es el crontab del sistema y no deberia ser editado en la mayoria de los casos ya que el archivo del sistema podria ser sustituido por una actualizacion y los cambios se perderian.

**Programacion de Horas y min con Cron:** La primera columna es la que tiene del min 0-59 de la primera hora que Cron debe ejecutar.

`25 6 * * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )`

La segunda columna es la hora del dia apartir de las 0-23. Un asterisco (\*) significa un

[Ramon](#)

posible valor y se utiliza como comodín. Mediante la combinación de estas dos primeras columnas, se puede obtener un valor de tiempo para el comando. Por ejemplo, la segunda línea de la tabla tiene 25 en la columna de minutos y 6 horas en la columna. Esto significa que la segunda línea se debe ejecutar a las 6:25 de la mañana.

```
25 6 *** root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
```

Del mismo modo, la primera línea significa que el comando se debe ejecutar cada hora, a los 17 minutos pasó la hora:

El cuarto registro se ejecuta en el primer día del mes a las 6:52 de la mañana

```
17 * *** root cd / && run-parts --report /etc/cron.hourly
```

Para mas informacion sobre como funciona Cron mirar:

### **How To Schedule Routine Tasks With Cron and Anacron on a VPS**

<https://www.digitalocean.com/community/tutorials/how-to-schedule-routine-tasks-with-cron-and-anacron-on-a-vps>

La siguiente columna implica la ejecucion de los comandos programados.

La sexta columna, solo esta presente en **Crontab**.

La columna final especifica el comando real que debe ser ejecutado. El comando puede contener un signo de porcentaje (%), lo que significa que todo más allá de la primera señal por ciento se pasa al comando como entrada estándar.

Cada registro tiene que terminar con un carácter de nueva línea. Hay que asegurarse de que tenga una línea en blanco para la última entrada o de lo contrario el comando no funcionará correctamente.

### **Carpets de Comandos:**

#### **El uso de run-parts y Cron Directorios.**

El comando **run-parts** se ejecuta en cada ejecutable ubicado dentro de un directorio especificado. Se utiliza con **Cron** ya que este permite le permite ejecutar varias secuencias de comandos en un periodo determinado colocandolos en una ubicacion determinada. (Te dice que ejecutes todo lo que este en la direccion que tu le pones)

Por defecto la mayoría de las distribuciones de scripts se configuran en las carpetas para cada intervalo donde se ejecutan dichos script.

En **Ubuntu** por ejemplo en la mayoría de las carpetas con los nombres

"Cron.daily", "Cron.hourly", "Cron.monthly", "Cron.weekly" es donde se ejecutan los comandos.

```
horabaixa@daw1-17:~$ run-parts
run-parts: missing operand
Try 'run-parts --help' for more information.
horabaixa@daw1-17:~$ cat /etc/cron
cron.d/      cron.hourly/  crontab
cron.daily/  cron.monthly/ cron.weekly/
horabaixa@daw1-17:~$ cat /etc/cron
```

```
horabaixa@daw1-17:~$ run-parts
run-parts: missing operand
Try 'run-parts --help' for more information.
horabaixa@daw1-17:~$ cat /etc/cron
cron.d/      cron.hourly/  crontab
cron.daily/  cron.monthly/ cron.weekly/
horabaixa@daw1-17:~$ cat /etc/crontab
cat: /etc/crontab: No existe el archivo o el directorio
horabaixa@daw1-17:~$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab`
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
t /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
t /etc/cron.weekly )
```

### Usando Crontab:

El comando **crontab** sirve para crear tareas con su propio usuario.

Para ver el **crontab** actual podemos escribir **crontab -l** probablemente no salga nada a menos que se haya creado antes. Si lo tienes creado hacemos una copia de seguridad antes de modificar cualquier archivo. Esto se puede hacer con el comando **crontab -l >~/cron.back** en el directorio personal.

Para editar el crontab se puede **escribir crontab -e**.

Para modificarlo seleccionamos el editor.

1. / bin / nano <---- más fácil
2. /usr/bin/vim.basic
3. /usr/bin/vim.tiny



Pagina para configurar el Crontab  
[http://crontab.guru/#\\*/15\\_\\*\\_\\*\\_\\*\\_5](http://crontab.guru/#*/15_*_*_*_5)

```
GNU nano 2.2.6      Archivo: /tmp/crontab.W9RGSx/crontab      Modificado

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
*/10 * * * 5 echo "$(date)">> /home/horabaixa/fecha

^G Ver ayuda ^O Guardar ^R Leer Fich ^Y RePág. ^K Cortar Text ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^V Pág. Sig. ^U PegarTxt ^T Ortografía
```

Ramon

Ahi esta editado para que salga la fecha cada 10 min y se muestre en el Usuario horabaixa todos los miercoles (5) y echo para que ejecute la orden completa.

# Anacron

## Que es Anacron?

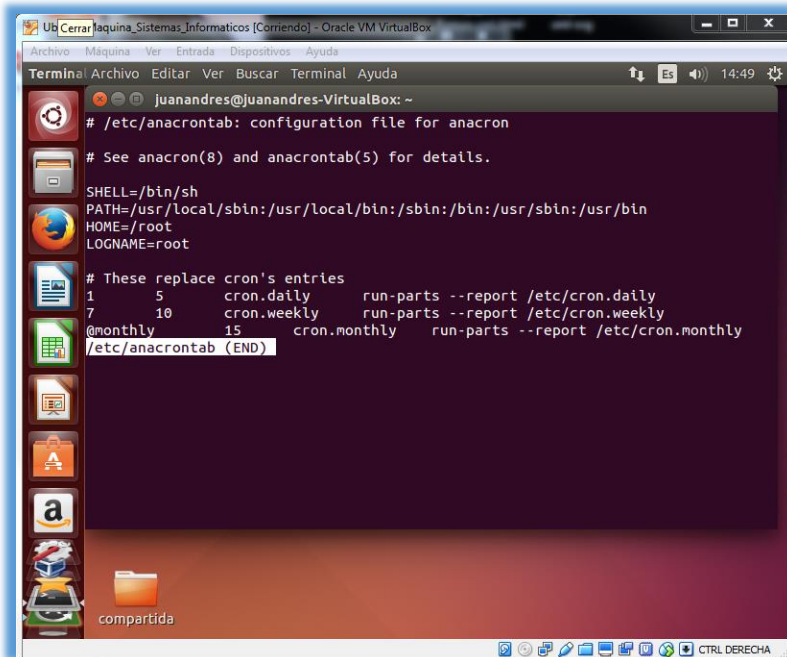
Uno de los problemas de Cron es que asume que el servidor esta siempre encendido y si el ordenador está apagado las tareas asignadas nunca se ejecutaran, por esta razón se creó Anacron, para arreglar este problema.

Anacron utiliza parámetros que no son tan detallados como las opciones de Cron, el sistema de anacron se ejecuta por días, esto significda que Anacron se creó para complementar a Cron, njo para sustituirlo.

Ventajas:

Anacron utiliza marcas de tiempo para averiguar cuando fue la última vez que sus comandos fueron ejecutados. Esto significa que cuando se programa una tarea y el servidor está apagado cuando se ejecuta Anacron deja una marca de tiempo y esto le permite luego ver cuando fue la última vez que se ejecutó dicha orden pasado 24 horas.

Anacron también tiene su directorio, se puede acceder a el con el comando  
Less /etc/anacrontab

A screenshot of a terminal window titled 'Ubuntu\_12.04 LTS [Corriendo] - Oracle VM VirtualBox'. The terminal shows the contents of the /etc/anacrontab file. The output includes comments about the configuration file, shell settings (SHELL=/bin/sh, PATH, HOME, LOGNAME), and a table of cron-like entries. The entries are: 1 5 cron.daily run-parts --report /etc/cron.daily, 7 10 cron.weekly run-parts --report /etc/cron.weekly, and @monthly 15 cron.monthly run-parts --report /etc/cron.monthly. The terminal ends with '/etc/anacrontab (END)'. The desktop background is Ubuntu 12.04 LTS, and the terminal window is on top of it.

```

# /etc/anacrontab: configuration file for anacron
# See anacron(8) and anacrontab(5) for details.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
HOME=/root
LOGNAME=root

# These replace cron's entries
1 5 cron.daily run-parts --report /etc/cron.daily
7 10 cron.weekly run-parts --report /etc/cron.weekly
@monthly 15 cron.monthly run-parts --report /etc/cron.monthly

/etc/anacrontab (END)
```

### Como trabajar con Anacron:

Podemos ver que el formato es bastante similar a Cron aunque con menos líneas y algunas diferencias, la primera columna especifica con que frecuencia se debe ejecutar el comando, en este caso tenemos un 1 (se ejecutara 1 vez por día). Si tuviéramos un 3 o algún otro valor 2 o el número que sea indica que se ejecutara el comando en el caso de 2(cada 2 días), 3 (cada 3 días) etc.

La segunda columna es el delay(tiempo que tarda) en utilizar o ejecutar los comandos. Anacron escalona las ejecuciones de los comandos, de esta manera se pueden ejecutar todas las ordenes programadas a la vez.

Por ejemplo: La primera columna se ejecuta cada día a los 5 min de, la siguiente línea se ejecuta cada 7 días a los 10 min, y la tercera línea contiene la instrucción.

Ejemplo: 1 – 5 – cron.daily run-parts --report /etc/cron.daily

Día 1 a los 5 min carpeta que le indica la frecuencia con la que se ejecutara cron.daily, en este caso diario y la instrucción que debe de ejecutar.

7 10 cron.weekly nice run-parts --report /etc/cron.weekly

Si se le quiere cambiar el tiempo solo tiene que cambiar la última parte de la instrucción (cron.weekly) esta es semanal.

Como vemos anacron está configurado para ejecutar las mismas secuencias de comandos que ejecuta cron.

### Conclusiones:

Tanto cron como anacron son herramientas útiles para cuando se necesita automatizar el proceso de arranque del sistema. Aunque al principio pueda ser algo difícil de entender estas herramientas ahorrarán tiempo y por lo general no tienen que ajustarse a menudo por lo que facilitará mucho el trabajo y pueden llegar a ser de mucha utilidad si usted necesita que el servidor u ordenador ejecute tareas aunque no esté en funcionamiento.