

Servicio de Tareas Programas

Servicio: crond

Permite ejecutar aplicaciones según un patrón de tiempo (con una resolución mínima a 1 minuto).

Ya viene instalado y corriendo por defecto en la mayoría de los Linux.

service crond [start|stop|restart]

systemctl [start|stop|restart] crond.service

Configuración:

/etc/crontab

Configuración básica del cron

```
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
```

/etc/cron.d

/etc/cron.daily/

/etc/cron.hourly/

/etc/cron.monthly/

/etc/cron.weekly/

/etc/cron.allow

Listado de usuarios que pueden utilizar el crond

/etc/cron.deny

Listado de usuarios que no pueden utilizar el crond, se puede utilizar la opción “ALL” para especificar a todos los usuarios.

/var/spool/cron/

Carpeta donde se almacenan los cron de cada usuario.

Comandos:

crontab -u XXX -l

crontab -u XXX -e

Macros:

Entrada	Descripción	Equivale A
@reboot	Cada vez que se reinicia el equipo.	
@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 * * * *

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

Comodines:

- 5 Se va a ejecutar el valor especificado (ej: 5).
- 1,2 Se va a ejecutar cuando el valor sea 1 o 2.
- 1-4 Rango, se va a ejecutar cuando el valor sea 1,2,3 y 4.
- * Se va a ejecutar para cualquier valor.
- */2 Se va a ejecutar cada dos valores.
- 3/2 Se va a ejecutar cada dos valores comenzando por el tercero.

Ejemplos:

```
* * * * * comando
    Se va a ejecutar todos los días cada minuto
30 10 * * 1 comando
    Los lunes a las 10:30 Hs
0,30 * * * 1 comando
    Los lunes cada media hora (30 minutos)
0,15,30,45 * * * * comando
    Todos los días cada 15 minutos
*/15 * * * * comando
    Todos los días cada 15 minutos
* * * * 1-2 comando
    Se va a ejecutar los lunes y martes cada minuto
```

```
@reboot                sleep 60 && comando.sh
    Ejecuta comando.sh 60 segundos luego de arrancar el equipo
```

Resoluciones mínimas:

```
* * * * *                comando && sleep 30 && comando
    Se va a ejecutar el comando cada 30 segundos.
```

IMPORTANTE:

Verificar que la demora del comando a ejecutar no sea mayor a la frecuencia especificada.

Si no, los procesos se apilarán y se agotará la CPU y memoria.

Controlar el tiempo de ejecución "timeout"

Controlar si el proceso ya no se está ejecutando.

Herramientas Online para crear cron:

<https://crontab-generator.org/>

<https://crontab.guru/>

<https://www.site24x7.com/es/tools/crontab/cron-generator.html>

Servicio: anacron

Anacron es parecido al “cron” (ejecuta tareas en segundo plano) pero para equipo de escritorio y notebooks, no espera que el equipo esté encendido 24 x 7 como un servidor.

Por ejemplo, si tiene una secuencia de comandos de respaldo programada todos los días a las 2300 Hs. Como un trabajo cron regular, y si su computadora portátil no funciona a las 2300 Hs., Su trabajo de respaldo no se ejecutará.

Sin embargo, si tiene el mismo trabajo programado en anacron, puede estar seguro de que se ejecutará una vez que la computadora portátil vuelva a funcionar.

Anacrontab Format

Al igual que el “cron” tiene el archivo “/etc/crontab”, anacron tiene “/etc/anacrontab”. El archivo /etc/anacrontab tiene los trabajos de anacron mencionados en el siguiente formato:

```
period delay job-identifier command
```

Field 1 is Recurrence period: This is a numeric value that specifies the number of days.

1 – daily

7 – weekly

30 – monthly

N – This can be any numeric value. N indicates number of days

Note: You can also use ‘@monthly’ for a job that needs to be executed monthly.

Field 2 is Delay: This indicates the delay in minutes. i.e X number of minutes anacron should wait before executing the job after the the machine starts.

Field 3 is Job identifier: It is the name for the job’s timestamp file. It should be unique for each job. This will be available as a file under the /var/spool/anacron directory. This file will contain a single line that indicates the last time when this job was executed.

```
# ls -l /var/spool/anacron/  
test.daily  
cron.daily  
cron.monthly  
cron.weekly
```

```
# cat /var/spool/anacron/test.daily  
20110507
```

Field 4 is command: Command or shell script that needs to be executed.

Just like shell scripts, comments inside anacrontab file starts with #

Note: For /etc/crontab file format, refer to our Linux Crontab: 15 Awesome Cron Job Examples article.

Ejemplo:

El siguiente ejemplo ejecuta el script /home/sathiya/backup.sh una vez cada 7 días.

El día en que se supone que se ejecutará el trabajo backup.sh, si el sistema no funciona por alguna razón, anacron ejecutará el script backup.sh 15 minutos después de que el sistema vuelva a funcionar (sin tener que esperar otros 7 días).

```
# cat /etc/anacrontab
7      15      test.daily      /bin/sh /home/sathiya/backup.sh
START_HOURS_RANGE and RANDOM_DELAY
```

The above example indicates that the backup.sh script should be executed every day, with a delay of 15 mins. i.e When the laptop was started, executed it only after 15 minutes.

What happens when the laptop or desktop was not shutdown? When does the job gets executed? This is specified by the START_HOURS_RANGE environment variable in the /etc/anacrontab file.

By default this is set to 3-22 in the file. This indicates the time range from 3 a.m to 10 p.m.

```
# grep START /etc/anacrontab
START_HOURS_RANGE=3-22
```

On top of the user defined delay specified in the 2nd field of the /etc/anacrontab file, anacron also randomly adds x number of minutes. The x is defined by the RANDOM_DELAY variable in the /etc/anacrontab file.

By default this is set to 45 in the file. This means that anacron will add x minutes (randomly picked from 0 and 45), and add this to the user defined delay.

```
# grep RANDOM /etc/anacrontab
RANDOM_DELAY=45
```

cron vs. anacron

cron y anacron tienen sus propias ventajas y desventajas. Dependiendo de sus necesidades, utilice uno de ellos.

cron	anacron
La granularidad mínima es de minutos (es decir, se pueden programar trabajos para que se ejecuten cada minuto).	La granularidad mínima es solo en días.
El trabajo cron puede ser programado por cualquier usuario normal (si no está restringido por un super usuario).	anacron solo puede ser utilizado por superusuarios (pero existen soluciones para que los usuarios normales puedan utilizarlo).
Cron espera que el sistema funcione las 24 horas del día, los 7 días de la semana. Si un trabajo está programado y el sistema no funciona durante ese tiempo, el trabajo no se ejecuta.	anacron no espera que el sistema funcione las 24 horas del día, los 7 días de la semana. Si un trabajo está programado y el sistema no funciona durante ese tiempo, iniciará los trabajos cuando el sistema vuelva a funcionar.
Ideal para servidores.	Ideal para equipos de escritorio y equipos móviles.
Use cron cuando un trabajo deba ejecutarse a una hora y minuto en particular.	Utilice anacron cuando deba ejecutar un trabajo independientemente de la hora y los minutos.