

Optativa

Explicación del comando Crontab



Ejecutamos el siguiente comando y elegimos qué editor de texto queremos para escribir nuestro “script” de crontab

Crontab -e

```
usuario@usuario-VirtualBox:~$ sudo su
[sudo] contraseña para usuario:
root@usuario-VirtualBox:/home/usuario# crontab -e
```

Con el siguiente comando haremos que cada día a las 2:30 AM se ejecute nuestro Script de PHP para la eliminación de usuarios con Antigüedad de 1 año sin iniciar sesión. También podremos indicar la salida que tendrá nuestro comando (en este ejemplo podremos redirigir la salida a un log).

También podremos únicamente elegir ejecutar el script de PHP, ej (info crontab)
./limpiar_usuarios_inactivos.php).

```
“30 2 * * * /usr/bin/php
/home/usuario/Desktop/Scripts/crontab/limpiar_usuarios_inactivos.php >>
/home/usuario/logs/cron_limpieza.out 2>&1”
```

```
GNU nano 6.2 /tmp/crontab.jRaXhn/crontab *
# 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
30 2 * * * /usr/bin/php /home/usuario/Scripts/crontab/limpiar_usuarios_inactivos.php >> /home/usuario/logs/cron_limpieza.out 2>&1
```