
Administración de sistemas



Actividad clase 1

18/10/2021

Marcos Eladio Somoza

ÍNDICE

Enunciado	4
Resoluciones.....	4
1. Ejecutar un script todos los días 22 de noviembre a las 7:30	5
2. Ejecutar un script todos los lunes y viernes a las 18:30h.....	5
3. Ejecutar un script a todas las horas en el minuto 30	5
4. Ejecutar un script todos los días 15 de marzo a las 22:30 y que sea domingo	5
5. Ejecutar un script todos los días a las 5:30 de la tarde todos los días de lunes a viernes ...	5
6. Ejecutar un script de lunes a viernes cada 8 minutos desde las 2:00 horas durante una hora.....	5
7. Ejecutar un script todos los viernes a las 22:00 y guardar la salida en /var/log/script.log	6
8. Ejecutar un script todos los días entre el 10 y 20 de todos los meses y guardar la salida en un archivo llamado registro_[fecha].log en la carpeta /var/log	6

Enunciado

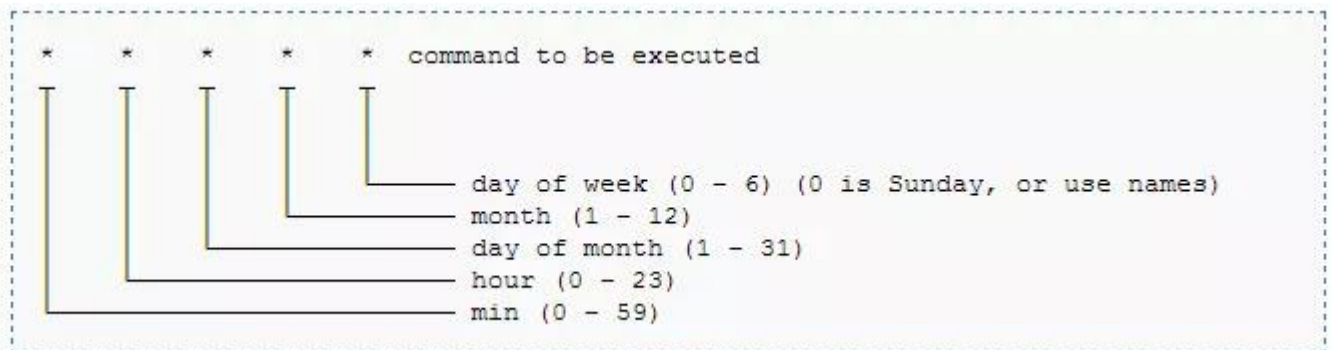
Indica el proceso general para agregar una tarea programada en el crontab y de forma individual indica las líneas que deberías agregar para las siguientes programaciones:

- 1.-Ejecutar un script todos los días 22 de noviembre a las 7:30.
- 2.-Ejecutar un script todos los lunes y viernes a las 18:30h.
- 3.-Ejecutar un script a todas las horas en el minuto 30.
- 4.-Ejecutar un script todos los días 15 de marzo a las 22:30 y que sea domingo.
- 5.-Ejecutar un script todos los días a las 5:30 de la tarde todos los días de lunes a viernes.
- 6.-Ejecutar un script de lunes a viernes cada 8 minutos desde las 2:00 horas durante una hora.
- 7.-Ejecutar un script todos los viernes a las 22:00 y guardar la salida en /var/log/script.log.
- 8.-Ejecutar un script todos los días entre el 10 y 20 de todos los meses y guardar la salida en un archivo llamado registro_[fecha].log en la carpeta /var/log

Resoluciones

Para agregar una tarea al crontab, dado que son tareas asociadas a usuarios, hay que asegurarse de estar trabajando con root, para ello, se usará **sudo crontab -e** y al elegir nano como editor, se podrán añadir las tareas a cada línea.

La estructura para una tarea en crontab debe seguir el siguiente diagrama:



Donde hay 5 asteriscos, representando de izquierda a derecha minuto, hora, día del mes, mes, día de la semana seguidos del comando a ejecutar.

1. Ejecutar un script todos los días 22 de noviembre a las 7:30

La clave de esta tarea es tener en cuenta que para que se ejecute todos los días 22 de noviembre, no debe importar el día de la semana, por lo que se dejará el * haciendo referencia a que da igual qué día de la semana sea .

30 7 22 11 * command

2. Ejecutar un script todos los lunes y viernes a las 18:30h

En este caso, dado que se quiere ejecutar los lunes y viernes (1 y 5 respectivamente) se deberá poner 1,5 en el lugar del asterisco.

30 18 * * 1,5 command

3. Ejecutar un script a todas las horas en el minuto 30

Dado que en realidad lo que se quiere es ejecutar el script cada media hora, bastaría con decir que a los 30 minutos da igual la hora, el día, el mes o cualquier otro parámetro.

30 * * * * command

4. Ejecutar un script todos los días 15 de marzo a las 22:30 y que sea domingo

En este caso, no hay nada especial más que recordar que la numeración de día de la semana empieza con el domingo como el 0.

30 22 15 3 0 command

5. Ejecutar un script todos los días a las 5:30 de la tarde todos los días de lunes a viernes

Para este comando hay que darse cuenta de que las 5 de la tarde son las 17 a la hora de codificarlo.

30 17 * * 1,2,3,4,5 command

6. Ejecutar un script de lunes a viernes cada 8 minutos desde las 2:00 horas durante una hora

Para decir que se ejecute cada X minutos, se tendrá que poner en su asterisco correspondiente */x . Sería lo mismo que escribir tantas veces X como se necesitaran para llegar a cumplir todos los minutos que forman la hora (de 0 a 60).

****/8 2 * * 1,2,3,4,5 command***

7. Ejecutar un script todos los viernes a las 22:00 y guardar la salida en /var/log/script.log

Este comando tiene la peculiaridad de generar un log. Para guardarlo, se deberá usar >> después del comando seguido del path . Cabe destacar que con un solo > también lo guardaría, pero sobrescribiría el log anterior, con dos >> añade el nuevo log en el mismo archivo.

0 22 * * 5 command >> /var/log/script.log

8. Ejecutar un script todos los días entre el 10 y 20 de todos los meses y guardar la salida en un archivo llamado registro_[fecha].log en la carpeta /var/log

Para decir en el comando que ejecute todos los días entre el 10 y el 20, se podría haber puesto como se ha hecho hasta ahora con los días de la semana (10,11,12,13,14...) pero a fin de ahorrar tiempo y hacer todo más visible, se puede asignar un rango desde el 10 al 20 poniendo un guion tal que 10-20. Además, dado que se pide guardar la fecha actual en el nombre del fichero log, se ha recurrido al comando date y al llamado **command substitution**, para llamar a dicho comando desde la misma línea que se crea el fichero con ``date +%F``. El `%F` devuelve la fecha completa y no se debe olvidar poner el `\` para que se reconozca el `%`.

0 0 10-20 * * command >> /var/log/registro_`date +%F`.log