

Journaling

Christian Amauri Amador Ortega

Benemérita Universidad Autónoma de Puebla
Facultad de ciencias de la computación,
Av. San claudio y 14 sur Puebla, México, christian.amadoro@alumno.buap.mx

En la introducción, damos un poco de teoría y contexto sobre el tema y explicamos de forma breve para qué sirve cada comando utilizado en la práctica. En el Desarrollo mostramos los resultados prácticos de la utilización de los comandos sobre los que hablamos en la introducción. En la conclusion hacemos una pequeña reflexión sobre la importancia del uso de estos comandos y estas practicas en un entorno real. Y finalmente tenemos una sección de referencias y una de bibliografía donde encontraremos material de apoyo donde Podemos profundizar sobre estos temas.

1 Introduction

Como ya lo hemos visto antes, la automatización de tareas es una de las funcionalidades más útiles que se pueden encontrar en cualquier sistema operativo. En Linux, una de las herramientas más populares para la automatización de tareas es el comando "cron". Cron es un demonio que se ejecuta en segundo plano en Linux y se utiliza para programar la ejecución de comandos o scripts en momentos específicos. Esto puede ser especialmente útil para realizar tareas repetitivas, como realizar copias de seguridad de archivos, actualizar bases de datos o enviar correos electrónicos. En esta practica simplemente vamos a reforzar estos conocimientos que ya antes habíamos visto con más ejercicios de cron.

2 Desarrollo

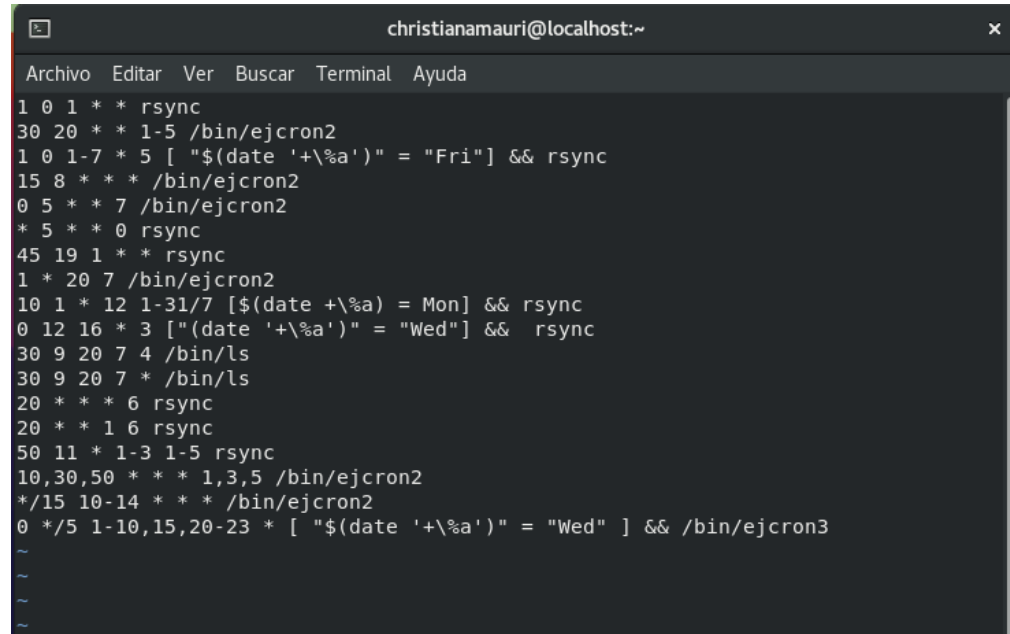
Para poder editar la hoja de cron, tenemos que abrir la terminal, e introducir el comando “crontab -e”. luego de eso, podremos empezar a ingresar los comandos de programación de tareas.

Resolveremos los siguientes planteamientos:

1. Programar un trabajo para ejecutarse cada 15 minutos
2. Programar un trabajo para ejecutarse cada día a las 00:00h
3. Programar un trabajo para ejecutarse cada primer día de mes a las 00:00h
4. Programar un trabajo para ejecutarse durante los días laborables a las 20:30h
5. Ejecutar un trabajo el primer viernes de cada mes a las 00:01h
6. A las 8:15 a.m. de cada día
7. A las 5 a.m. todos los domingos
8. Cada minuto de 5:00a.m. a 5:59a.m. todos los domingos
9. A las 7:45 p.m. del primero de cada mes
10. Al minuto 1 de cada hora del 20 de julio
11. A la 1:10 a.m. todos los lunes de diciembre
12. Al mediodía de los días 16 de cada mes y que sea Miércoles
13. A las 9:30 a.m. del día 20 de julio y que sea jueves
14. A las 9:30 a.m. del día 20 de julio sin importar el día de la semana
15. Al minuto 20 de cada hora de los sábados
16. Al minuto 20 de cada hora de los sábados de enero
17. A las 11:59 a.m. de lunes a viernes, de enero a marzo
18. En el minuto 10, 30 y 50 de todas las horas de los días lunes, miércoles y viernes
19. Cada quince minutos de las 10:00a.m. a las 2:00p.m.
20. Cada 5 horas de los días 1 al 10, el día 15 y del día 20 al 23 de cada mes y que el día sea miércoles.

Resultados:

A continuación se muestra la hoja de crone n la cual se guardaron los comandos correspondientes a los respectivos ejercicios previamente planteados:



```
christianamauri@localhost:~  
Archivo Editar Ver Buscar Terminal Ayuda  
1 0 1 * * rsync  
30 20 * * 1-5 /bin/ejcron2  
1 0 1-7 * 5 [ "$(date '+\%a')" = "Fri" ] && rsync  
15 8 * * * /bin/ejcron2  
0 5 * * 7 /bin/ejcron2  
* 5 * * 0 rsync  
45 19 1 * * rsync  
1 * 20 7 /bin/ejcron2  
10 1 * 12 1-31/7 [$(date +\%a) = Mon] && rsync  
0 12 16 * 3 [ "$(date '+\%a')" = "Wed" ] && rsync  
30 9 20 7 4 /bin/ls  
30 9 20 7 * /bin/ls  
20 * * * 6 rsync  
20 * * 1 6 rsync  
50 11 * 1-3 1-5 rsync  
10,30,50 * * * 1,3,5 /bin/ejcron2  
*/15 10-14 * * * /bin/ejcron2  
0 */5 1-10,15,20-23 * [ "$(date '+\%a')" = "Wed" ] && /bin/ejcron3  
~  
~  
~  
~
```

3 Conclusión

En conclusión, los comandos cron son una herramienta muy útil para automatizar tareas repetitivas en sistemas operativos Unix y Linux. Con la capacidad de ejecutar comandos en momentos específicos o en intervalos regulares, cron es una solución práctica para ahorrar tiempo y reducir errores. La sintaxis simple y flexible de los comandos cron permite una gran variedad de opciones de programación para adaptarse a diferentes necesidades y requisitos. Es importante recordar que la correcta configuración y uso de cron requiere un conocimiento adecuado del sistema operativo y de los comandos que se ejecutarán.

4 Referencias

- <https://www.digitalocean.com/community/tutorials/how-to-use-journalctl-to-view-and-manipulate-systemd-logs-es>
- <https://www.digitalocean.com/community/tutorials/how-to-use-cron-to-automate-tasks-centos-8>
- <https://serverspace.io/support/help/automate-regular-tasks-cron-centos-8/>

5 Bibliografía

- Wilson, R. (2015). Linux: The ultimate beginner's guide! CreateSpace Independent Publishing Platform.
- Petersen, S. (2018). Linux Command Line and Shell Scripting Bible. John Wiley & Sons.