** **

**Administración de Redes**

**Docente: Ana Claudia Zenteno Vázquez**

**Práctica 10 Uso de CRON , CRONTAB Y AT**

**Nombre:Edgar Gael Sanchez Hernandez Fecha:20/03/2024**

Hay dos formas de usar Cron:

1.- La primera es por medio de los directorios situados en el directorio "/etc"

**La primera opción -Cron por medio de directorios-**

Si nos vamos al directorio "/etc" y ejecutamos el comando "ls" nos encontraremos con que tenemos, entre otros muchos, los siguientes directorios:

"cron.daily" → (Una vez al día)

"cron.hourly" → (Cada hora)

"cron.monthly" → (Una vez al mes)

"cron.weekly" → (Una vez a la semana)

Para programar una ejecución, basta con introducir un archivo de tipo script en una de las carpetas, dependiendo de la regularidad con la que queramos ejecutar el script.

1. ¿Qué debe hacer para que un script se ejecute cada hora usando los directorios de cron?

R= **sudo nano /etc/cron.hourly/mi\_script.sh**

Escribir el script en el archivo

**#!/bin/bash**

**/ruta/a/tu/comando**

Dar permisos y ejecutar

sudo chmod +x /etc/cron.hourly/mi\_script.sh

1. ¿Y cada mes?

R= Editar el archivo crun.monthly, el cual contiene scripts que se ejecutaran cada hora. Podemos agregar el script que deseemos a este archivo.

**sudo nano /etc/cron.monthly/mi\_script.sh**

Escribir el script en el archivo

**#!/bin/bash**

**/ruta/a/tu/comando**

Dar permisos y ejecutar

sudo chmod +x /etc/cron.monthly/mi\_script.sh

1. ¿Y cada día?

**R=** Editar el archivo crun.hourlym, el cual contiene scripts que se ejecutaran cada hora. Podemos agregar el script que deseemos a este archivo.

**sudo nano /etc/cron.hourly/mi\_script.sh**

Escribir el script en el archivo

**#!/bin/bash**

**/ruta/a/tu/comando**

Dar permisos y ejecutar

sudo chmod +x /etc/cron.hourly/mi\_script.sh

2.- La segunda es por medio de la manipulación directa del archivo "/etc/crontab"

**La segunda opción -Uso de crontab-**

**Analice los ejemplos**

**Ejemplo**

|  |  |
| --- | --- |
| 30   10   \*   \*   \* | Todos los días 10:30 am |
| 0   0   1,10,20 \* \* | Todos 1, 10, 20 de cada mes a las 0:00 am |
| 0,30   \*   \*   \*   1 | Cada 30 minutos todos los lunes |
| 0   22  1,2,3,4,5,6,7   \*   5 | Los días Viernes que sean 1, 2, 3, 4, 5, 6, 7 a las 0:22 am |
| 0   3   \*   \*   1,4 | Todos los Lunes y jueves a las 00:00 am |

Usando crontab y el script de formatos de fechas, programe lo siguiente

a. Ejecute la tarea a las 11:30 de la noche

R= 30 23 \* \* \* /ruta/al/formato\_fechas.sh

b. Ejecute la tarea el 29 de junio a las 10:08 am

R=08 10 29 06 \* /ruta/al/formato\_fechas.sh

c. Ejecute la tarea en la hora actual 8 días después

R=0 0 \* \* \* /ruta/al/formato\_fechas.sh

d. Ejecute la tarea a las 8:43 dentro de 3 semanas

R=43 8 \* \* 0 /ruta/al/formato\_fechas.sh

e. Ejecute la tarea a la medianoche

R=0 0 \* \* \* /ruta/al/formato\_fechas.sh

f. Ejecute la tarea mañana a las 3:25pm

R=25 15 \* \* \* /ruta/al/formato\_fechas.sh

**Script**

#!/bin/bash

# Calcula la fecha y hora exactas dentro de 3 semanas

fecha=$(date -d "+3 weeks +43 minutes" "+%M %H %d %m \*")

# Muestra la fecha y hora exactas

echo "$fecha"

e. Ejecute la tarea a la medianoche

**0 0 \* \* \* root …./…/formatofechas.sh**

f. Ejecute la tarea mañana a las 3:25pm

**USO del comando at**

Vaya al manual de at.

¿Qué función realiza el comando?

R= at: El comando at se utiliza para programar la ejecución de comandos o scripts en un momento específico. Por ejemplo, puedes utilizar at para programar la ejecución de un script a las 3:00 PM del próximo jueves.

¿Para qué sirve atq?

R= atq: atq es un comando que muestra una lista de trabajos en cola para ser ejecutados con el comando at. Muestra el ID del trabajo, la fecha y la hora en que se programó su ejecución.

¿Para qué sirve atrm?

R= atrm: El comando atrm se utiliza para eliminar trabajos de la cola de at. Puedes especificar el ID del trabajo que deseas eliminar. Esto es útil si has programado un trabajo para ejecutarse en el futuro pero decides que ya no necesitas que se ejecute.

Programe una tarea que añada al archivo /root/uptime.txt” la salida del comando “uptime”

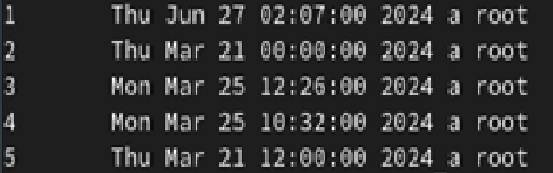
Programe este script para que se ejecute:

Sintaxis: at [tiempo] < [ruta del script]

at 20:00 **< /ruta/tarea** // ejecuta la tarea a las 8:00 PM

* 1. at 13:56  11/11/2021 **<** /ruta/tarea // el formato de fecha es MM/DD/AAAA
  2. at 02:07  JUN 27 **<** /ruta/tarea // ejecutara la tarea a esa hora el día 27 de Junio
  3. at midnight **<** /ruta/tarea // se ejecutara la media noche
  4. at now + 5 days **<** /ruta/tarea // se ejecutará dentro de 5 días a esta hora
  5. at 10:32 + 5 weeks **<** /ruta/tarea // se ejecutará dentro de 5 semanas a las 10:32
  6. at noon  **<** /ruta/tarea // se ejecuta al medio día

Compruebe la ejecución de esta tarea y muestre la captura de pantalla donde se vean todas estas tareas programadas



Proceda a eliminar las tareas de los incisos b, d y f.

Muestre la captura de pantalla donde se vean las tareas programadas actualizadas **(que se note la eliminación de dichas tareas)**