

cron

Automatización de tareas

Laboratorio de Sistemas Operativos II



La **automatización de tareas** es conjunto de métodos que nos permiten simplificar tareas repetitivas. Simplifica y garantiza la ejecución de esas tareas.

¿Cuáles tareas se automatizan?

- Actividades de gestión periódicas y control del sistema.
- Limpieza periódica de archivos temporales (/tmp)
- Copias de seguridad.
- Generación de informes regulares del sistema.
- Programar el apagado automático del equipo.
- Programar tareas periódicas desde las necesidades del usuario (recordatorios, o la descarga programada de información en horas de poco tráfico).

cron

GNU/Linux cuenta con el programa **cron** que permite a los usuarios ejecutar automáticamente comandos o scripts (tareas) a una hora o fecha específica, o a intervalos regulares de tiempo.



cron

El nombre cron viene del griego **chronos** que significa "tiempo".



¿Cómo funciona cron?

- Cron es un demonio (**servicio**) que lleva el mismo nombre.
- Es iniciado cada vez que el sistema se inicia y se ejecuta en background.
- Revisa cada minuto el archivo que contiene la tabla de tareas (**crontab**) en búsqueda de tareas que se deban cumplir.
- La tabla de tareas se ubica en /etc/crontab

Administrar el servicio cron

Operacion	Comando
Iniciar el servicio	<code># systemctl start cron</code>
Detener el servicio	<code># systemctl stop cron</code>
Reiniciar el servicio	<code># systemctl restart cron</code>
Ver el estado del servicio	<code># systemctl status cron</code>

crontab



- Es un archivo de texto que contiene la lista de tareas planificadas para cron. Este archivo se encuentra en el directorio /etc.
- Está estructurado por líneas, cada una de las cuales contiene una tarea programada, cada línea sigue un formato determinado que debe ser entendido para poder programar cada tarea.

Formato de línea de contrab

Minuto Hora DiaDelMes Mes DiaDeLaSemana Usuario Comando

Comando, script o programa

Usuario que ejecuta el comando.

Día en la semana en que se ejecutará el comando, puede ser de 0 a 7, o por el nombre del día en inglés, solo las tres primeras letras. (0 y 7 = domingo)

Mes en que el comando se ejecutará, puede ser indicado del 1 al 12), o por el nombre del mes en inglés, solo las tres primeras letras.

Día del mes en que se quiere ejecutar el comando. Va del 1 al 31.

Hora en que el comando será ejecutado, en un formato de 24 horas, los valores deben estar entre 0 y 23, 0 es medianoche.

Corresponde al minuto en que se va a ejecutar el script, el valor va de 0 a 59

Detalle del formato

Minuto	Controla el minuto de la hora en que el comando será ejecutado, este valor debe de estar entre 0 y 59.
Hora	Controla la hora en que el comando será ejecutado, se especifica en un formato de 24 horas, los valores deben estar entre 0 y 23, 0 es medianoche.
Día del Mes	Día del mes en que se quiere ejecutar el comando. Por ejemplo se indicaría 20, para ejecutar el comando el día 20 del mes.
Mes	Mes en que el comando se ejecutará, puede ser indicado numéricamente (1-12), o por el nombre del mes en inglés, solo las tres primeras letras.
Día de la semana	Día en la semana en que se ejecutará el comando, puede ser numérico (0-7) o por el nombre del día en inglés, solo las tres primeras letras. (0 y 7 = domingo)
Usuario	Usuario que ejecuta el comando.
Comando	Comando, script o programa que se desea ejecutar. Este campo puede contener múltiples palabras y espacios.

Un asterisco * como valor en los primeros cinco campos, indicará inicio-fin del campo, es decir todo. Un * en el campo de minuto indicará todos los minutos.

Cómo editar crontab

- Como root, se editará el /etc/crontab directamente:

```
#nano /etc/crontab
```

- Como usuario normal, se utiliza el comando crontab para que de este modo cada usuario obtenga automáticamente su propio crontab en /var/spool/cron/crontabs

```
$crontab -e
```

Otras opciones del comando crontab

- Lista el crontab actual del usuario, sus tareas de cron

```
$crontab -l
```

- Elimina el crontab actual del usuario (cuidado si eres root!!)

```
$crontab -r
```

Ejemplos:

- Ejecutará el script actualizar.sh a las 10:15 a.m. todos los días

```
15 10 * * * usuario /home/usuario/scripts/actualizar.sh
```

- El día 21 de noviembre a las 6:30 el usuario correrá el script backup.sh

```
30 6 21 11 * usuario /home/usuario/scripts/backup.sh
```

- A las 12:00 del día todos los días primero, quince y 28 de cada mes

```
00 12 1,15,28 * * root /opt/generarNominas.sh
```


Más Ejemplos:

- Cada quince minutos de las 11:00a.m. a las 14:00p.m.

```
*/15 11-14 * * * root /opt/checkDNS.sh
```

- Cada 6 horas de los días 1 al 10 de cada mes y que el día sea miércoles

```
0 */6 1-10 * 3 root /opt/checkDNS.sh
```

- root ejecutará una actualización los domingos (sun) a las 10:45 a.m

```
45 10 * * sun root apt-get -y update
```

Cron por hora, diario, semanal, mensual

- **/etc/cron.daily**

Cron ejecuta todos los scripts ubicados en cron.daily una vez al día en la hora especificada en /etc/crontab.

- **/etc/cron.hourly**

Los scripts contenidos en el directorio /etc/cron.hourly se ejecutarán cada hora. (24 veces al día si el equipo está siempre encendido). La hora de ejecución se especifica en /etc/crontab.

- **/etc/cron.monthly**

Cron ejecuta todos los scripts ubicados en cron.weekly una vez a la semana en el día y hora especificados en /etc/crontab.

- **/etc/cron.weekly**

Cron ejecuta todos los scripts ubicados en cron.monthly mensualmente en el día y hora especificados en /etc/crontab

Página de referencia

CRON y ANACRON

<https://geekland.eu/planificar-tareas-con-cron-y-anacron-en-linux/>



Fin.-

¿Día y Horario de consulta?