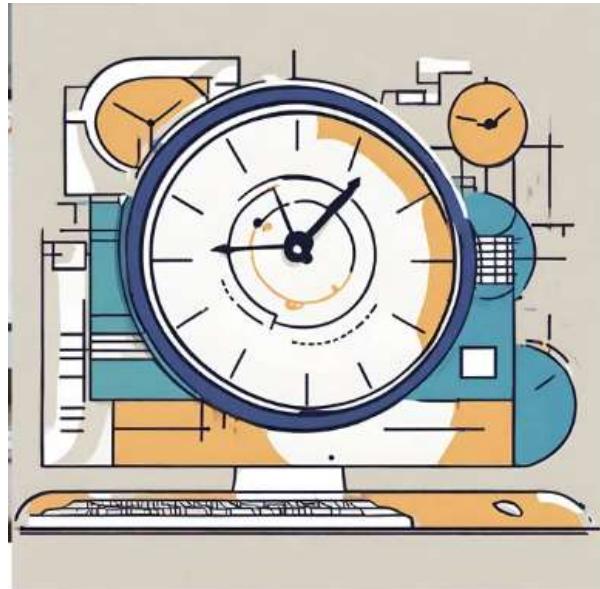


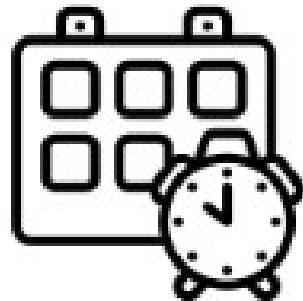
AUTOMATIZACIÓN DE TAREAS



- ❖ COMANDO AT

- ❖ SERVICIO CRONTAB

AUTOMATIZACIÓN DE TAREAS



Herramienta CRONTAB

Características de Crontab y Cron

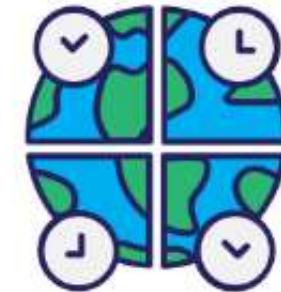
- Programación de **tareas repetitivas** en intervalos específicos: horas, días, semanas
- Cron ejecuta acciones programadas **cada minuto**, según las especificaciones en `/etc/crontab`.
- **Archivo de texto crontab diseñado para ser leído por el demonio Cron.**
- Cada usuario guarda su propia versión de este archivo
- Configuración precisa de **hora y zona horaria** es esencial para garantizar ejecuciones correctas.



Configuración de Servicio Crontab

Comando para verificar y configurar la **hora y zona horaria**:

```
timedatectl set-timezone Europe/Madrid
```



Orden para comprobar que el **demonio está activo**:

```
systemctl status cron
```

```
systemctl start cron
```

Sintaxis del archivo crontab

5 asteriscos y el comando a ejecutar

```
* * * * * command/script/to/be/executed
- - - - -
| | | | |
| | | | ----- Day of week (0 - 7) (Sunday=0 or 7)
| | | ----- Month (1 - 12)
| | ----- Day of month (1 - 31)
| ----- Hour (0 - 23)
----- Minute (0 - 59)
```

Algunas opciones de herramienta Crontab

crontab -e para editar el archivo existente

crontab -l para listar tareas en crontab del usuario

crontab -d para borrar el crontab configurado

Configuraciones predeterminadas

CADENA

@reboot:

@yearly o @anually:

@monthly:

@weekly:

@daily:

@midnight:

@hourly:

Cuándo se ejecutará

una vez que se inicia el equipo.

una vez al año.

el primer día de cada mes.

una vez por semana.

todos los días.

todas las noches a las doce.

primer minuto de cara hora del día.

PASOS crontab



1.- Comprobar que el demonio está activo



2.- Comprobar hora del sistema



3.- Crear el script y darle permiso de ejecución



4.- Programar fichero crontab

AUTOMATIZACIÓN DE TAREAS



Comando **at**

Características de **at**

- Permite programar **tareas únicas** en sistemas GNU/Linux.
- Útil para ejecutar comandos o scripts en momentos específicos, como apagar el sistema o enviar correos.
- Sintaxis básica:

at hora [fecha]

- También tiene su demonio: systemctl status atd

Opciones del comando **at**

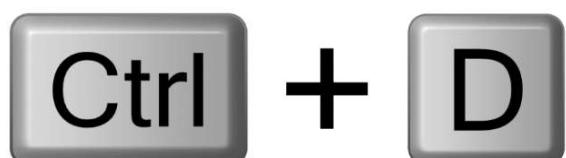
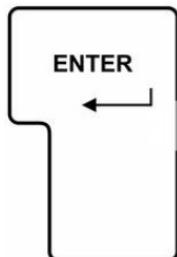
at: ejecuta comandos a la hora especificada

atq: enumera los trabajos pendientes de los usuarios (**at -l**)

atrm: borra trabajo/s por su número de trabajo (**at -r**)

at -c <num_tarea>: para ver el contenido de una tarea concreta

PASOS comando at



- 0.- Comprobar que el demonio está activo
- 1) Ejecutar el comando **at** indicando hora y día (opcional) en que se desee se ejecute la tarea.
- 2) Escribir cada comando a ejecutar en una línea, pulsar **INTRO** para finalizar la entrada de cada comando.
- 3) Finalizar la entrada de la tarea **at**, con **CTRL+D**

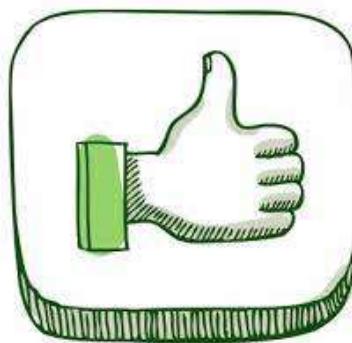
Ejecución de at

- Los comandos se ejecutan con **/bin/sh**
- La hora debe estar en el futuro para que la tarea se ejecute
- Todas las tareas se almacenan en:
/var/spool/cron/atjobs

Ejemplos de at

```
at 11:40 2024-02-13
ls -ltr /tmp > "~/prueba_comando_at.txt"
echo 'finalizado' >> "~/prueba_comando_at.txt"
CTRL + D
```

VENTAJAS Y UTILIDADES DE LA AUTOMATIZACIÓN



- Aumenta la **eficiencia** y reduce el **esfuerzo humano** en la administración del sistema.
- Permite programar **tareas repetitivas** y **únicas** de manera eficiente
- Facilita la **gestión** de actualizaciones, copias de seguridad y otras acciones sistemáticas.
- Mejora la **confiabilidad** al evitar errores humanos en ejecuciones manuales.



EJEMPLOS PRÁCTICOS de automatización de tareas



Copia de Seguridad Diaria:

Programar una tarea en Crontab para realizar copias de seguridad diarias a las 2:00 a.m.

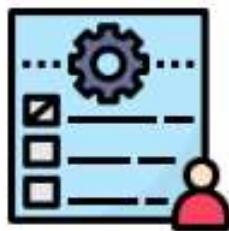
```
0 2 * * * /ruta/al/script_copia_seguridad.sh
```



Actualización Automática de Software:

Configurar una tarea en Crontab para actualizar automáticamente el sistema todos los domingos a las 3:00 a.m.

```
0 3 * * 0 /usr/bin/apt-get update && /usr/bin/apt-get upgrade -y
```



Recordatorio de Reunión Semanal:

Utilizar el comando at para enviar un recordatorio por correo electrónico el próximo viernes a las 10:00 a.m.

```
at 10:00 next fri
echo 'Recordatorio: Reunión semanal' | mail -s
'Recordatorio' usuario@dominio.com
```

Actividad

P17 – Trabajos