

# at - Programación de tareas

**at** es una herramienta de programación de tareas que permite al mismo tiempo realizar tareas en su sistema. El **batch** es como **at** pero realiza las tareas de una sola vez cuando la carga del sistema lo permite, todos los comandos y funciones de **at** pueden ser replicados por **batch** con los mismos argumentos.

## Comandos relacionados

Se incluyen los siguientes comandos para monitorear el comando **at** :

- **at** : el programa solía enviar una tarea única.
- **atq** : el programa utilizado para revisar la cola de trabajos enviados con **at**.
- **atrm** : el programa utilizado para eliminar trabajos de la cola.

## El uso del comando at

El comando **at** puede emitirse para permitirle escribir un script personalizado o señalarlo a un script preconstruido. Además, las amplias especificaciones de tiempo permiten aumentar la complejidad.

El argumento `-m` se puede agregar a cualquier comando que le indique al daemon que envíe un correo al usuario en el sistema local para avisarle de la finalización de la tarea.

## Programación de un script existente

Este ejemplo le dirá a la aplicación **at** que ejecute el script `/home/user/testscript.sh` a las 4PM de hoy y envíe un correo electrónico al usuario al finalizar.

```
user@darkstar$ at 16:00 -m -f /home/user/testscript.sh
```

Como puede ver, el argumento `-f` significa **archivo** ya que apunta al archivo que se ejecutará.

## Programando una nueva acción

Como se ve a continuación, cuando no se especifica ningún archivo, la aplicación **at** abre una solicitud para que el usuario ingrese el nuevo script.

```
user@darkstar$ at 19:40 -m
warning: commands will be executed using /bin/sh
at> ping -c 4 www.google.com
```

```
at> echo $?  
at> <EOT>  
job 4 at Sun Dec 30 19:40:00 2012
```

El usuario deberá escribir la secuencia de comandos que se emitirán dentro de la solicitud; al finalizar, cierre la solicitud con `Ctrl` + `d`

El comando anterior le indicó a **at** que haga ping a Google 4 veces a las 7:40 PM, que devuelva el estado de la aplicación y envíe los resultados por correo electrónico al usuario.



Asegúrese de anotar el número de trabajo, es el único identificador visible para el trabajo y debe ser referenciado si desea eliminar el trabajo más adelante.

## Revisión de tareas en cola

El comando **atq** enumera los trabajos pendientes del usuario. La salida es Número de trabajo, fecha, hora, cola y nombre de usuario.

```
user@darkstar$ atq  
4 Sun Dec 30 19:40:00 2012 a user
```

Una vez que se haya anotado el número de tarea, puede revisar los comandos que se incluyen en el trabajo emitiendo el comando `en -c {número de trabajo}`, y el ejemplo se muestra a continuación.

```
user@darkstar$ at -c 4  
#!/bin/sh  
# atrun uid=1000 gid=1000  
# mail user 1  
umask 22  
....Environment Details....  
cd /home/user || {  
    echo 'Execution directory inaccessible' >&2  
    exit 1  
}  
ping -c 4 www.google.com  
echo $?
```

## Eliminar tareas en cola

El comando **atrm** elimina los trabajos de la cola pendiente, no presenta ninguna confirmación de que el trabajo se haya eliminado, por lo que puede ser recomendable ejecutar **atq** después de **atrm** para confirmar que el trabajo se ha eliminado.

```
user@darkstar$ atrm 4
```

# Further Resources

- [at manual page](#)
- [BrunoLinux](#)
- <http://sathyaphoenix.wordpress.com/2009/01/18/using-at-command-to-schedule-jobs-in-linux/>

# Fuentes

- Originalmente escrito por [mfillpot](#)
- Traducido por [Victor](#) 2019/08/27 14:49 (UTC)

[howtos](#), [task scheduling](#), [author mfillpot](#)

From:  
<https://docs.slackware.com/> - **SlackDocs**

Permanent link:  
<https://docs.slackware.com/es:howtos:software:at>

Last update: **2019/08/27 14:52 (UTC)**

