

En esta práctica vamos a trabajar la automatización de tareas con los comandos **cron**, **anacron** y **at**. Puedes ayudarte en la siguiente página:

<http://myfpschool.com/automatizacion-de-tareas-at/>

CRON

```
sudo apt install cron
sudo systemctl enable cron
```

1. ¿Qué ficheros comprueba el servicio cron constantemente para comprobar si hay tareas pendientes?

`/etc/crontab`

`/etc/cron.d/`

`/var/spool/cron/crontabs/`

2. ¿Dónde se guardan las tareas programada con el comando `crontab`?

En el directorio `/var/spool/cron/crontabs/`

3. ¿Qué dos ficheros utilizamos para controlar el acceso de los usuarios a las programación de tareas? Descríbelos.

`/etc/cron.allow`: Si existe, solo los usuarios listados en este archivo pueden usar cron.

`/etc/cron.deny`: Si existe y `cron.allow` no está presente, los usuarios listados aquí no podrán usar cron.

4. ¿Qué orden utilizarías para eliminar las tareas programadas por un usuario concreto?
- `crontab -r -u usuario`

5. Escriba las líneas del archivo `crontab` de usuario para que dado el script en

`/scripts/job.sh` se ejecute el script `job.sh` a las 5 a. M. Todos los días

`0 5 * * * /scripts/job.sh`

6. El mismo script `job.sh` se ejecute dos veces al día a las 6 a. M. Y a las 6 p. M. (una única línea)

`0 6,18 * * * /scripts/job.sh`

7. Programe un cron para que se ejecute cada minuto

`* * * * * /scripts/job.sh`

8. Programe un cron para que se ejecute en cada Monday a las 7 pm.

`0 19 * * 1 /scripts/job.sh`

9. Programe un cron para que se ejecute cada 15 minutos.

`*/15 * * * * /scripts/job.sh`

10. Programe un cron para que se ejecute en los meses seleccionados: feb, jun, oct
`* * * 2,6,10 * /scripts/job.sh`
11. Ejecute el script de shell /home/script/backup.sh el 4 de marzo a las 7:25 AM
`25 7 4 3 * /home/script/backup.sh`
12. Programe una tarea que cada cinco minutos escriba la fecha y la hora en el fichero
 /var/log/mi-cron.log.
`* /5 * * * * date >> /var/log/mi-cron.log`
13. Programe una tarea que, cada día a las 0:00 horas, busque los archivos que contengan
 la palabra "hack" y los guarde en la carpeta /root/cuarentena.
`0 0 * * * grep -rl "hack" / | xargs -l {} mv {} /root/cuarentena/`
14. Programe una tarea que elimine los ficheros almacenados en la carpeta /tmp todos los
 domingos a las 2:00 horas.
`0 2 * * 0 rm -rf /tmp/*`
15. Programe una tarea que liste todos los ficheros que puedan ser modificados por
 cualquier usuario y guarde dicha lista con la ruta exacta en el fichero
 /roo/archivo_peligrosos.txt todos los lunes a las 7:15 horas.
`15 7 * * 1 find / -perm -002 -type f > /roo/archivo_peligrosos.txt`
16. Programe una tarea que apague el equipo todos los días a las 23:59 horas.
`59 23 * * * shutdown -h now`
17. Planifica una tarea para que se eliminen los archivos y los directorios vacíos de /tmp a
 las 00:30 todos los días
`30 0 * * * find /tmp -type f -empty -delete; find /tmp -type d -empty -delete`

ANACRON

18. ¿Para qué sirve este comando?
 permite ejecutar tareas programadas como cron, pero no necesita que el equipo esté
 encendido en el momento exacto. Si es el caso, se ejecutará la tarea en el momento que
 se encienda el equipo.
19. Instala el comando si no lo está ya
`sudo apt install anacron -y`
20. ¿En qué fichero se configuran las tareas programadas con anacron?
 En /etc/anacrontab
21. ¿Qué se registra en el directorio /var/spool/anacron?
 Almacena archivos que registran la última vez que se ejecutó cada tarea definida, para así
 saber si debe ejecutar alguna tarea al encenderse el ordenador.

22. Deja preparada una tarea diaria para que una vez pasados 15 minutos tras encender la máquina clasifique todos los archivos de descargas por carpetas (los pdf a un directorio que se llame PDF y los doc a un directorio DOC).

1 15 clasificar_descargas /bin/bash /home/usuario/scripts/clasificar_descargas.sh

clasificar_descargas.sh

#!/bin/bash

mkdir -p ~/Descargas/PDF ~/Descargas/DOC

mv ~/Descargas/*.pdf ~/Descargas/PDF/ 2>/dev/null

mv ~/Descargas/*.doc ~/Descargas/DOC/ 2>/dev/null

mv ~/Descargas/*.docx ~/Descargas/DOC/ 2>/dev/null

23. Semanalmente borra todos los archivos que no estén clasificados (los que no sean ni pdf ni doc).

7 20 limpiar_descargas /bin/bash /home/usuario/scripts/limpiar_descargas.sh

limpiar_descargas.sh

#!/bin/bash

find ~/Descargas -type f ! -name "*.pdf" ! -name "*.doc" ! -name "*.docx" -delete

AT

24. ¿Para que sirve este comando?

echo "rm -rf /tmp/*.tmp" | at now + 10 minutes

25. ¿Qué archivos están relacionados con at?

/var/spool/at/ Tareas pendientes programadas con at.

/etc/at.allow Lista de usuarios que pueden usar at.

/etc/at.deny Lista de usuarios que no pueden usar at.

26. ¿Qué ocurre si no existe ningún fichero relacionado con at?

Si no existe /etc/at.allow ni /etc/at.deny, solo root podrá usar at.

Si existe solo /etc/at.allow, solo los usuarios dentro de ese archivo podrán usar at.

Si existe solo /etc/at.deny, todos los usuarios pueden usar at, excepto los que estén en ese archivo.

27. Instala el comando si no lo está ya

sudo apt install at -y

28. ¿Cuál es el fichero que está usando **at** en tu sistema? Modifícalo y añade un usuario para permitir o denegar usar **at**. Adjunta captura de pantalla.

```
ubuntu@ip-172-31-91-158:~$ sudo cat /etc/at.deny
alias
backup
bin
daemon
ftp
games
gnats
guest
irc
lp
mail
man
nobody
operator
proxy
qmaild
qmail1
qmailp
qmailq
qmailr
qmails
sync
sys
www-data
```

```
ubuntu@ip-172-31-91-158:~$ echo "usuariol" | sudo tee -a /etc/at.allow
usuariol
ubuntu@ip-172-31-91-158:~$ echo "ubuntu" | sudo tee -a /etc/at.allow
ubuntu
ubuntu@ip-172-31-91-158:~$ sudo cat /etc/at.allow
usuariol
ubuntu
```

29. Deja programada una tarea para dentro de 5 minutos que elimine de un directorio todos los ficheros con extensión tmp.

```
ubuntu@ip-172-31-91-158:~$ echo "rm -rf /home/ficheros/*.tmp" | at now + 5 minutes
warning: commands will be executed using /bin/sh
job 1 at Fri Jan 31 08:30:00 2025
ubuntu@ip-172-31-91-158:~$
```

30. Deja programada una tarea con at a las 14:20 de hoy para que elimine los ficheros del directorio descargas de tu sistema.

```
ubuntu@ip-172-31-91-158:~$ echo "rm -rf ~/Descargas/*" | at 14:20
warning: commands will be executed using /bin/sh
job 2 at Fri Jan 31 14:20:00 2025
ubuntu@ip-172-31-91-158:~$
```

31. Impide que tu usuario pueda ejecutar tareas programadas con at utilizando el fichero deny y prueba a programar una tarea cualquiera. Verifica que el sistema le impide dejar programada una tarea.
- echo "\$(whoami)" | sudo tee -a /etc/at.deny

32. ¿Qué hace el siguiente ejemplo?

```
$ at -m 1930
at> rm /home/jones/*.backup
at> Press Control-D
job 897355800.a at Thu Sep 29 19:30:00 2022
```

Este comando programa la eliminación de todos los archivos .backup en /home/jones/ a las 19:30 del día elegido, y enviará un correo al usuario una vez ejecutado.

33. Programar con at para que se ejecute a las 12 a.m. del día siguiente un script preparado por el usuario "script.sh":

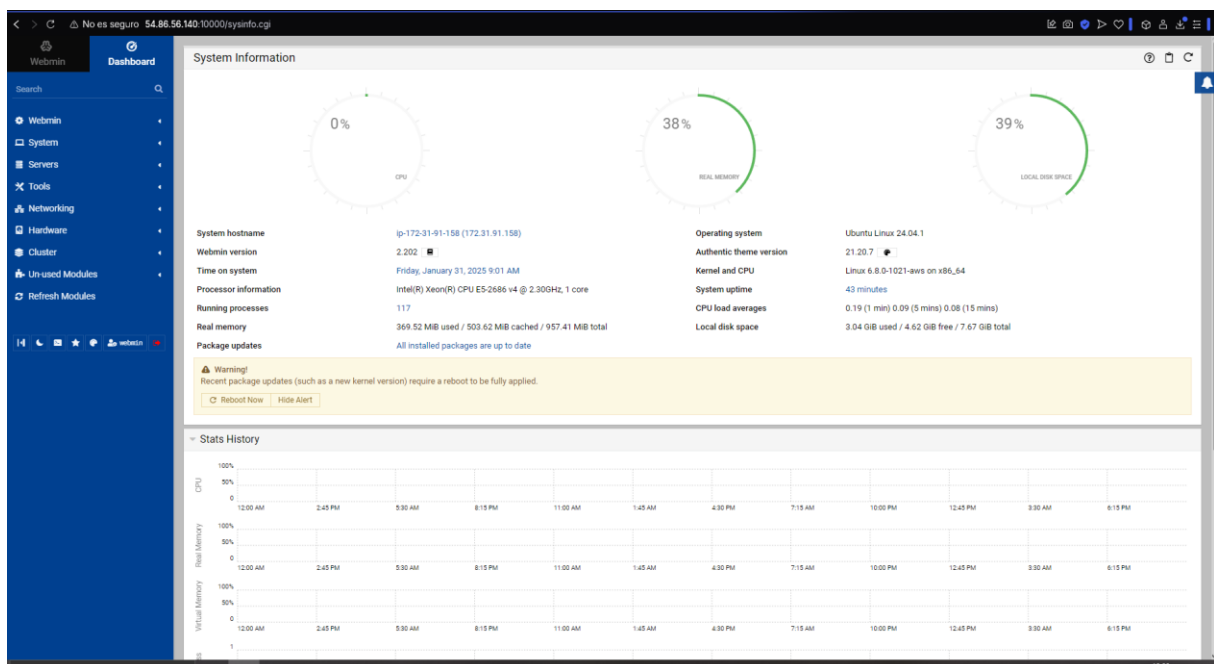
```
echo "/home/script.sh" | at midnight
```

34. Vamos a realiza la programación de las siguientes dos tareas de manera gráfica. Para ello vamos a emplear dos herramientas gráficas: Webmin y gnome-schedule. Instalamos la aplicación. Webmin ya la conoces. Vamos a instalar gnome-schedule.

```
apt-get install gnome-schedule
```

gnome-schedule fue descontinuado.

<https://github.com/GNOME/gnome-schedule.git>



Tareas a configurar (hazlo con las dos herramientas):

- Hacer un backup del directorio de usuario

←

Create Cron Job

Execute cron job as

webmin

Active?

• Yes

No

Command

tar -czf /backup/usuario_backup_\$(date +%F).tar.gz /home/usuario

Input to command

Description

When to execute

Simple schedule ... Hourly

• Times and dates selected below ...

Minutes

All

Selected ..

0

12

24

36

48

1

13

25

37

49

2

14

26

38

50

3

15

27

39

51

4

16

28

40

52

5

17

29

41

53

6

18

30

42

54

7

19

31

43

55

8

20

32

44

56

9

21

33

45

57

10

22

34

46

58

11

23

35

47

59

Hours

All

Selected ..

0

12

1

13

2

14

3

15

4

16

5

17

6

18

7

19

8

20

9

21

10

22

11

23

Days

All

Selected ..

1

13

25

2

14

26

3

15

27

4

16

28

5

17

29

6

18

30

7

19

31

8

20

9

21

10

22

11

23

12

24

Months

All

Selected ..

January

February

March

April

May

June

July

August

September

October

November

December

Weekdays

All

Selected ..

Sunday

Monday

Tuesday

Wednesday

Thursday

Friday

Saturday

Note: Ctrl-click (or command-click on the Mac) to select and de-select minutes, hours, days and months.

Date range to execute

• Run on any date

Only run from

Jan

to

Jan

Create

- Reiniciar el sistema a las 11:15 de la mañana

Execute cron job as

webmin

Input to command

Description

Job Details

Active?

☒ Yes
 ☐ No

Command

sudo shutdown -r now

When to execute

☐ Simple schedule ...

Hourly

☒ Times and dates selected below ...

Minutes

☐ All
 ☒ Selected ...

| | | | | |
|----|----|----|----|----|
| 0 | 12 | 24 | 36 | 48 |
| 1 | 13 | 25 | 37 | 49 |
| 2 | 14 | 26 | 38 | 50 |
| 3 | 15 | 27 | 39 | 51 |
| 4 | 16 | 28 | 40 | 52 |
| 5 | 17 | 29 | 41 | 53 |
| 6 | 18 | 30 | 42 | 54 |
| 7 | 19 | 31 | 43 | 55 |
| 8 | 20 | 32 | 44 | 56 |
| 9 | 21 | 33 | 45 | 57 |
| 10 | 22 | 34 | 46 | 58 |
| 11 | 23 | 35 | 47 | 59 |

Hours

☐ All
 ☒ Selected ...

| | |
|----|----|
| 0 | 12 |
| 1 | 13 |
| 2 | 14 |
| 3 | 15 |
| 4 | 16 |
| 5 | 17 |
| 6 | 18 |
| 7 | 19 |
| 8 | 20 |
| 9 | 21 |
| 10 | 22 |
| 11 | 23 |

Days

☐ All
 ☒ Selected ...

| | | |
|----|----|----|
| 1 | 13 | 25 |
| 2 | 14 | 26 |
| 3 | 15 | 27 |
| 4 | 16 | 28 |
| 5 | 17 | 29 |
| 6 | 18 | 30 |
| 7 | 19 | 31 |
| 8 | 20 | |
| 9 | 21 | |
| 10 | 22 | |
| 11 | 23 | |
| 12 | 24 | |

Months

☒ All
 ☐ Selected ...

| |
|-----------|
| January |
| February |
| March |
| April |
| May |
| June |
| July |
| August |
| September |
| October |
| November |
| December |

Weekdays

☒ All
 ☐ Selected ...

| |
|-----------|
| Sunday |
| Monday |
| Tuesday |
| Wednesday |
| Thursday |
| Friday |
| Saturday |

Note: Ctrl-click (or command-click on the Mac) to select and de-select minutes, hours, days and months.