

Sistemas Informáticos

UD4. Actividad 5

Programación de tareas Linux

INDICE

<u>Introducción</u>	3
Desarrollo de la práctica	
<u>Paso 1: Comprender cron</u>	4
<u>Paso 2: Conocer la estructura de cron</u>	4
<u>Paso 3: Agregar una tarea programada</u>	5
<u>Paso 4: Programar tres tareas normales</u>	6
<u>Paso 5: Programar tres tareas periódicas</u>	7
<u>Conclusión</u>	9
<u>Bibliografía</u>	10

Introducción

El objetivo es aprender a utilizar el programa *cron* en sistemas Linux/Ubuntu para automatizar tareas periódicas. *Cron* es una herramienta poderosa que permite a los usuarios ejecutar comandos automáticamente en momentos específicos. Conocer su sintaxis y cómo programar tareas adecuadamente es esencial para la administración de sistemas en entornos Linux. En actividad, se explorarán las tareas programadas a través de *cron*, incluyendo tareas sencillas como la creación de archivos y copias de seguridad, así como tareas más complejas como la actualización del sistema y el reinicio programado.

Paso 1: Comprender cron

Cron es una herramienta que se usa en sistemas como Linux (Ubuntu) para hacer tareas automáticamente, sin necesidad de que tú las hagas cada vez. Es como si le dijeras a la computadora: "haz esto todos los días a las 9 de la mañana", y ella lo hace por ti.

Paso 2: Conocer la estructura de cron

Las tareas que cron ejecuta se guardan en un archivo llamado *crontab*. Cada usuario en el sistema tiene su propio archivo de crontab. Este archivo tiene una estructura especial que le indica a cron cuándo y qué tareas hacer.

m h dom mon dow command

m Minutos (0 a 59)

h Hora (0 a 23)

dom Día del mes (1 a 31)

mon Mes (1 a 12)

dow Día de la semana (0 y 7 son domingo)

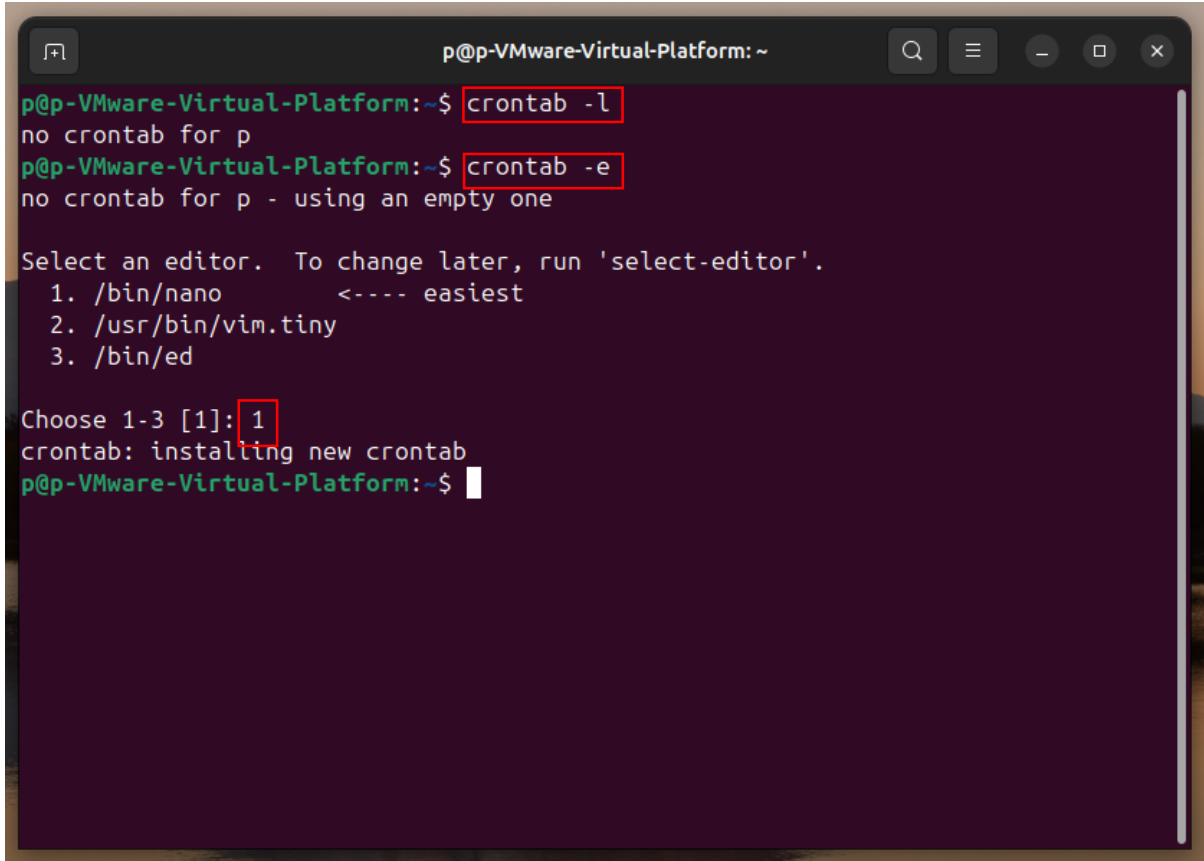
command El comando que quieres que se ejecute

Para especificar todos los valores posibles de una variable se utiliza un asterisco (*).

Ejemplo:

```
30 10 * * 1 /usr/bin/who >> /var/log/usuarios.txt
```

Paso 3: Agregar una tarea programada

A terminal window titled 'p@p-VMware-Virtual-Platform: ~' with standard window controls. The terminal shows the following sequence of commands and output: 1. Command: `crontab -l` (highlighted with a red box). Output: 'no crontab for p'. 2. Command: `crontab -e` (highlighted with a red box). Output: 'no crontab for p - using an empty one'. 3. Prompt: 'Select an editor. To change later, run 'select-editor''. 4. List of options: '1. /bin/nano <---- easiest', '2. /usr/bin/vim.tiny', '3. /bin/ed'. 5. Prompt: 'Choose 1-3 [1]:' (highlighted with a red box). Input: '1' (highlighted with a red box). 6. Output: 'crontab: installing new crontab'. 7. Prompt: `p@p-VMware-Virtual-Platform:~$` with a cursor.

```
p@p-VMware-Virtual-Platform:~$ crontab -l
no crontab for p
p@p-VMware-Virtual-Platform:~$ crontab -e
no crontab for p - using an empty one

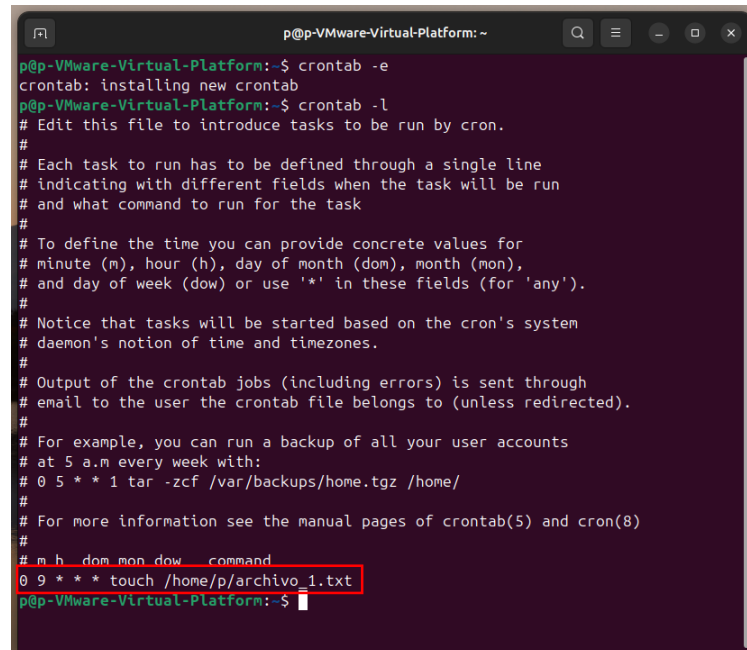
Select an editor. To change later, run 'select-editor'.
 1. /bin/nano      <---- easiest
 2. /usr/bin/vim.tiny
 3. /bin/ed

Choose 1-3 [1]: 1
crontab: installing new crontab
p@p-VMware-Virtual-Platform:~$
```

No sale nada después de `crontab -l` porque no tenemos ningún *crontab* asignado. Y si es la primera vez que asignamos un *cron* son va a preguntar que editor de texto queremos utilizar.

Paso 4: Programar tres tareas normales

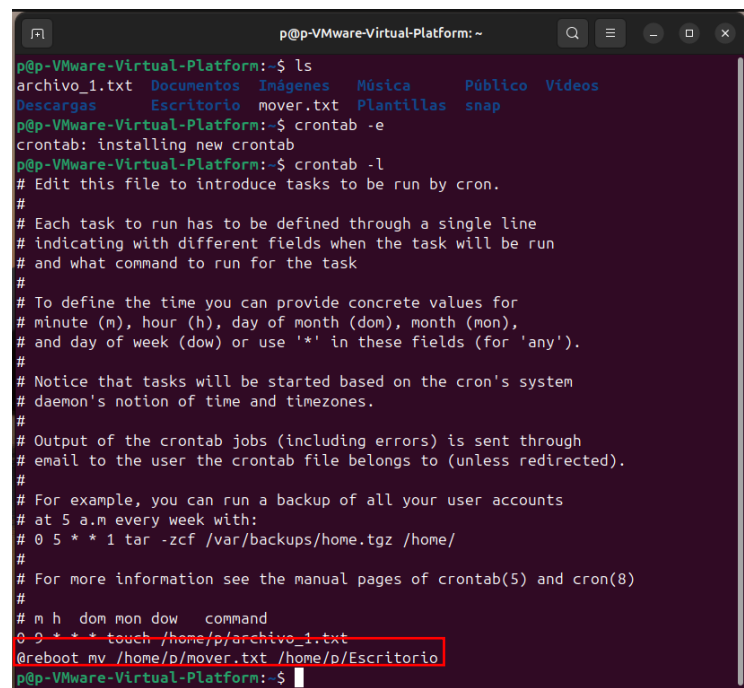
Tarea Normal 1:

A terminal window titled 'p@p-VMware-Virtual-Platform: ~' showing the installation and editing of the crontab. The user runs 'crontab -e', which opens a file with instructions. The line '0 9 * * * touch /home/p/archivo_1.txt' is highlighted with a red box.

```
p@p-VMware-Virtual-Platform:~  
p@p-VMware-Virtual-Platform:~$ crontab -e  
crontab: installing new crontab  
p@p-VMware-Virtual-Platform:~$ crontab -l  
# Edit this file to introduce tasks to be run by cron.  
#  
# Each task to run has to be defined through a single line  
# indicating with different fields when the task will be run  
# and what command to run for the task  
#  
# To define the time you can provide concrete values for  
# minute (m), hour (h), day of month (dom), month (mon),  
# and day of week (dow) or use '*' in these fields (for 'any').  
#  
# Notice that tasks will be started based on the cron's system  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow _ command  
0 9 * * * touch /home/p/archivo_1.txt  
p@p-VMware-Virtual-Platform:~$
```

Esto hará que cada día a las 9:00 AM, se cree un archivo llamado *archivo_1.txt* en tu carpeta.

Tarea Normal 2:

A terminal window titled 'p@p-VMware-Virtual-Platform: ~' showing the user's directory contents and the setup of a cron job. The user runs 'ls' showing files like 'archivo_1.txt', 'mover.txt', and 'Escritorio'. Then they run 'crontab -e' and the line '@reboot mv /home/p/mover.txt /home/p/Escritorio' is highlighted with a red box.

```
p@p-VMware-Virtual-Platform:~$ ls  
archivo_1.txt  Documentos  Imágenes    Música       Público     Videos  
Descargas     Escritorio  mover.txt   Plantillas  snap  
p@p-VMware-Virtual-Platform:~$ crontab -e  
crontab: installing new crontab  
p@p-VMware-Virtual-Platform:~$ crontab -l  
# Edit this file to introduce tasks to be run by cron.  
#  
# Each task to run has to be defined through a single line  
# indicating with different fields when the task will be run  
# and what command to run for the task  
#  
# To define the time you can provide concrete values for  
# minute (m), hour (h), day of month (dom), month (mon),  
# and day of week (dow) or use '*' in these fields (for 'any').  
#  
# Notice that tasks will be started based on the cron's system  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow _ command  
0 9 * * * touch /home/p/archivo_1.txt  
@reboot mv /home/p/mover.txt /home/p/Escritorio  
p@p-VMware-Virtual-Platform:~$
```

Esto moverá un archivo de una carpeta a otra cuando inicies sesión.

Tarea Normal 3:

```
p@p-VMware-Virtual-Platform: ~  
p@p-VMware-Virtual-Platform:~$ crontab -e  
crontab: installing new crontab  
p@p-VMware-Virtual-Platform:~$ crontab -l  
# Edit this file to introduce tasks to be run by cron.  
#  
# Each task to run has to be defined through a single line  
# indicating with different fields when the task will be run  
# and what command to run for the task  
#  
# To define the time you can provide concrete values for  
# minute (m), hour (h), day of month (dom), month (mon),  
# and day of week (dow) or use '*' in these fields (for 'any').  
#  
# Notice that tasks will be started based on the cron's system  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow  command  
0 9 * * * touch /home/p/archivo_1.txt  
@reboot mv /home/p/mover.txt /home/p/Escritorio  
0 18 * * 5 echo "Bienvenido al fin de semana!" | wall  
p@p-VMware-Virtual-Platform:~$
```

Esto mandará un mensaje a todos los usuarios del sistema.

Paso 5: Programar tres tareas periódicas

Tarea Periódica 1:

```
p@p-VMware-Virtual-Platform: ~  
p@p-VMware-Virtual-Platform:~$ crontab -e  
crontab: installing new crontab  
p@p-VMware-Virtual-Platform:~$ crontab -l  
# Edit this file to introduce tasks to be run by cron.  
#  
# Each task to run has to be defined through a single line  
# indicating with different fields when the task will be run  
# and what command to run for the task  
#  
# To define the time you can provide concrete values for  
# minute (m), hour (h), day of month (dom), month (mon),  
# and day of week (dow) or use '*' in these fields (for 'any').  
#  
# Notice that tasks will be started based on the cron's system  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow  command  
0 9 * * * touch /home/p/archivo_1.txt  
@reboot mv /home/p/mover.txt /home/p/Escritorio  
0 18 * * 5 echo "Bienvenido al fin de semana!" | wall  
0 * * * * rsync -av home/p/Documentos home/p/CopiaSeguridad  
p@p-VMware-Virtual-Platform:~$
```

Esto copiará todo lo que haya en esa carpeta a otro lugar cada hora.

Tarea Periódica 2:

```
p@p-VMware-Virtual-Platform: ~  
p@p-VMware-Virtual-Platform:~$ crontab -e  
crontab: installing new crontab  
p@p-VMware-Virtual-Platform:~$ crontab -l  
# Edit this file to introduce tasks to be run by cron.  
#  
# Each task to run has to be defined through a single line  
# indicating with different fields when the task will be run  
# and what command to run for the task  
#  
# To define the time you can provide concrete values for  
# minute (m), hour (h), day of month (dom), month (mon),  
# and day of week (dow) or use '*' in these fields (for 'any').  
#  
# Notice that tasks will be started based on the cron's system  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow  command  
0 9 * * * touch /home/p/archivo_1.txt  
@reboot mv /home/p/mover.txt /home/p/Escritorio  
0 18 * * 5 echo "Bienvenido al fin de semana!" | wall  
0 * * * * rsync -av home/p/Documentos home/p/CopiaSeguridad  
0 3 * * * apt update && apt update -y  
p@p-VMware-Virtual-Platform:~$
```

Esto actualizará todos los programas y el sistema.

Tarea Periódica 3:

```
p@p-VMware-Virtual-Platform: ~  
p@p-VMware-Virtual-Platform:~$ crontab -l  
# Edit this file to introduce tasks to be run by cron.  
#  
# Each task to run has to be defined through a single line  
# indicating with different fields when the task will be run  
# and what command to run for the task  
#  
# To define the time you can provide concrete values for  
# minute (m), hour (h), day of month (dom), month (mon),  
# and day of week (dow) or use '*' in these fields (for 'any').  
#  
# Notice that tasks will be started based on the cron's system  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow  command  
0 9 * * * touch /home/p/archivo_1.txt  
@reboot mv /home/p/mover.txt /home/p/Escritorio  
0 18 * * 5 echo "Bienvenido al fin de semana!" | wall  
0 * * * * rsync -av home/p/Documentos home/p/CopiaSeguridad  
0 3 * * * apt update && apt update -y  
0 2 * * 0 reboot  
p@p-VMware-Virtual-Platform:~$
```


Esto reiniciará el ordenador/servidor automáticamente

Conclusión

aprendimos cómo usar cron para automatizar tareas en un sistema Linux. Cron es una herramienta muy útil que nos ahorra tiempo y esfuerzo, ya que podemos hacer que la computadora realice tareas automáticamente en momentos específicos o de manera repetitiva. Ahora sabemos cómo agregar tareas simples y tareas que se repiten, como crear archivos, hacer copias de seguridad, actualizar el sistema y reiniciar el ordenador. Usar cron nos ayuda a ser más eficientes y a mantener el sistema funcionando correctamente sin tener que intervenir todo el tiempo.

Bibliografía

[HelpUbuntu](#)

[Wikipedia](#)

[Redeszone](#)