

Sistemas Informáticos

Actividad práctica: Gestión Integral y Automatización de un Sistema Linux. (Parte 1)

En esta actividad aprenderemos a crear y gestionar archivos en Linux utilizando comandos algo más complejos. Primero crearemos un pequeño árbol de directorios. Dentro de una de sus carpetas crearemos un archivo de texto, el cual dividiremos en tres partes. Después, las volveremos a unir y enviar a otra carpeta usando un mismo comando.

Por último, programaremos una tarea para actualizar ese mismo archivo de manera automatizada cada cinco minutos.

Configuración del entorno

Comenzaremos por crear los directorios y gestionar los permisos de cada uno de ellos. Después crearemos el archivo de texto que más adelante dividiremos en tres partes.

Lo primero que hacemos es operar desde root para operar como administrador. Esto no es fundamental pero así no habrá necesidad de introducir el comando “**sudo**” constantemente.

Para ello introducimos el comando **sudo passwd root**. Esto nos servirá para cambiar la contraseña. Se nos pedirá introducir una nueva.

```
donguillermo@DonGuillermo-VirtualBox:~$ sudo passwd root
[sudo] contraseña para donguillermo:
Nueva contraseña:
```

Una vez introduzcamos la nueva contraseña dos veces, si cumple con los requisitos mínimos, podremos operar desde el directorio raíz introduciendo **su root** y escribiendo la contraseña.

```
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
donguillermo@DonGuillermo-VirtualBox:~$ su root
Contraseña:
```

Ahora mismo antes de escribir cada comando debería aparecer como **root@usuario-VirtualBox:/home/usuario#**. Escribimos **mkdir proyecto_empresa/confidencial proyecto_empresa/publico proyecto_empresa/procesos**. Si escribimos **tree** debería mostrarnos todo el árbol de directorios existentes, incluidos los que acabamos de crear. Es posible que Linux nos pida instalar la herramienta para el comando tree. De ser así, solo hay que introducir el comando que el propio Linux nos pondrá en pantalla.

```

root@DonGuillermo-VirtualBox:/home/donguillermo# mkdir proyecto_empresa/confidencial proyecto_empresa/publico proyecto_empresa/proce
sos
root@DonGuillermo-VirtualBox:/home/donguillermo# tree
.
├── Descargas
├── Documentos
├── Escritorio
├── Imágenes
├── Música
├── Plantillas
└── proyecto_empresa
    ├── confidencial
    ├── procesos
    └── publico

```

Ahora vamos a usar el comando **ls -l** para listar la lista de directorios con todos sus permisos dentro de **proyecto_empresa**. Recordemos: podemos usar **cd** para acceder a dicho directorio, o escribir la ruta después de **ls -l**. En la imagen, ya estamos dentro del directorio. Podemos ver los permisos **drwxr-xr-x** en todos ellos. Esto significa que todos los usuarios tienen permiso de lectura.

```

root@DonGuillermo-VirtualBox:/home/donguillermo/proyecto_empresa# ls -l
total 12
drwxr-xr-x 2 root root 4096 nov 29 14:37 confidencial
drwxr-xr-x 2 root root 4096 nov 29 14:37 procesos
drwxr-xr-x 2 root root 4096 nov 29 14:37 publico

```

Vamos a cambiar los permisos de cada directorio usando:

chmod 700 confidencial para conceder acceso exclusivo a la carpeta confidencial solamente al usuario actual.

chmod 755 publico para conceder permiso de lectura a la carpeta publico a todos los usuarios, que es como estaba antes.

chmod 770 procesos para conceder acceso de lectura y escritura a la carpeta procesos al grupo propietario.

Al usar el comando **ls -l** debería aparecer como en la imagen,

```

root@DonGuillermo-VirtualBox:/home/donguillermo/proyecto_empresa# chmod 700 confidencial
root@DonGuillermo-VirtualBox:/home/donguillermo/proyecto_empresa# chmod 755 publico
root@DonGuillermo-VirtualBox:/home/donguillermo/proyecto_empresa# chmod 770 procesos
root@DonGuillermo-VirtualBox:/home/donguillermo/proyecto_empresa# ls -l
total 12
drwx----- 2 root root 4096 nov 29 14:37 confidencial
drwxrwx--- 2 root root 4096 nov 29 14:37 procesos
drwxr-xr-x 2 root root 4096 nov 29 14:37 publico

```

A continuación crearemos el archivo **clientes.txt** en el directorio confidencial. Para ello usamos **nano confidencial/clientes.txt**. En el archivo escribiré una lista de nombres de clientes.

```
root@DonGuillermo-VirtualBox:/home/donguillermo/proyecto_empresa# nano confidencial/clientes.txt
```

Usando **cat confidencial/clientes.txt** mostraremos el contenido del archivo. De forma intencional he repetido la misma lista de diez nombres tres veces para mostrar algo más adelante.

```
root@DonGuillermo-VirtualBox:/home/donguillermo/proyecto_empresa# cat confidencial/clientes.txt
Inés Perado
Elena Nito
Segundo Toro de la Tarde
Olga Itero
Guillermógenes de Tarso
Mac Intosh
Aitor Reznor
Erica Rica Tura
Armando Bronca
Francisco Teca
Inés Perado
Elena Nito
Segundo Toro de la Tarde
Olga Itero
Guillermógenes de Tarso
Mac Intosh
Aitor Reznor
Erica Rica Tura
Armando Bronca
Francisco Teca
Inés Perado
Elena Nito
Segundo Toro de la Tarde
Olga Itero
Guillermógenes de Tarso
Mac Intosh
Aitor Reznor
Erica Rica Tura
Armando Bronca
Francisco Teca
```

División y reunificación del archivo

Ahora dividiremos el archivo en tres partes y usaremos un comando para reunificarlo de nuevo. Lo interesante de esto es la forma en que lo haremos: Redirigiendo la salida de este comando desde el directorio **confidencial** al directorio **público**.

Lo primero que haremos es introducir **split -l 10 clientes.txt**. Esto dividirá el documento entre diez líneas. Debido a que el original tiene 30, esto hará que aparezcan tres documentos nuevos. Si usamos **ls** deberíamos ver tres nuevos archivos junto al de clientes.txt. En este caso, se llaman **xaa**, **xab**, **xac**.

```
root@DonGuillermo-VirtualBox:/home/donguillermo/proyecto_empresa/confidencial# split -l 10 clientes.txt
root@DonGuillermo-VirtualBox:/home/donguillermo/proyecto_empresa/confidencial# ls
clientes.txt xaa xab xac
```

Ahora vamos a reunificar el archivo pero redirigiendo la salida de esta acción a la carpeta **público**. Para ello usaremos **cat /home/donguillermo/proyecto_empresa/confidencial/xa* | sort | uniq > /home/donguillermo/proyecto_empresa/publico/resumen_publico.txt**.

Hay varias cosas que notar acerca de este comando. Para empezar, **cat** sirve tanto para crear como para mostrar el contenido de un archivo. Después de la ruta, notamos **xa***. Esto significa que el comando aplica a todo lo que empiece con las letras xa, así que nos ahorramos tener que escribir xaa, xab y xac.

| **sort** | sirve para ordenar el contenido.

| **uniq** | eliminará todas las líneas repetidas. (Por esto repetí los mismos diez nombres tres veces)

Por último, **>** seguido de la ruta hacia el directorio publico redirigirá la salida del comando hasta allí. Así, creamos el archivo resumen_publico.txt.

```
root@DonGuillermo-VirtualBox:/home/donguillermo/proyecto_empresa/confidencial# cat /home/donguillermo/proyecto_empresa/confidencial/xa* | sort | uniq > /home/donguillermo/proyecto_empresa/publico/resumen_publico.txt
```

Usamos **ls** seguido de la ruta hasta el directorio **público** y deberíamos ver el archivo **resumen_publico.txt**.

Finalmente, usamos **cat** seguido de la ruta donde se encuentra el archivo que acabamos que comprobar que existe, y veremos que solamente aparecen los diez nombres una vez. Esto se debe a | **uniq** |. Las tres partes del archivo han sido unificadas, pero dado que las tres eran exactamente iguales, este comando ha eliminado las líneas repetidas.

```

root@DonGuillermo-VirtualBox: /home/donguillermo/proyecto_empresa/confidencial# ls /home/donguillermo/proyecto_empresa/publico
resumen_publico.txt
root@DonGuillermo-VirtualBox: /home/donguillermo/proyecto_empresa/confidencial# cat /home/donguillermo/proyecto_empresa/publico/resumen_publico.txt
Aitor Reznor
Armando Bronca
Elena Nito
Erica Rica Tura
Francisco Teca
Guillermógenes de Tarso
Inés Perado
Mac Intosh
Olga Itero
Segundo Toro de la Tarde

```

Final: Script automatizado

Para la última parte de esta tarea, crearemos un script dentro de la carpeta **procesos** con un comando programado para que copie automáticamente cualquier archivo nuevo desde el directorio público al directorio procesos cada 5 minutos.

Usamos **nano** seguido de la ruta que lleva hasta el directorio procesos seguido de **/copiar_nuevos.sh** para crear el archivo donde programaremos el script.

```

root@DonGuillermo-VirtualBox: /home/donguillermo/proyecto_empresa/confidencial# nano /home/donguillermo/proyecto_empresa/procesos/copiar_nuevos.sh

```

A continuación mostramos el contenido del script:

```

root@DonGuillermo-VirtualBox: /home/donguillermo/proyecto_empresa/confidencial
GNU nano 6.2 /home/donguillermo/proyecto_empresa/procesos/copiar_nuevos.sh *
#!/bin/bash

DIR_ORIGEN== "/home/donguillermo/proyecto_empresa/publico"
DIR_DESTINO== "/home/donguillermo/proyecto_empresa/procesos"

find "$DIR_ORIGEN" -type f -cmin -5 -exec cp {} "$DIR_DESTINO" \;

```

#!/bin/bash especifica que este es un script de bash.

DIR_ORIGEN== “ruta de origen” define la ruta de origen. En este caso **“/home/donguillermo/proyecto_empresa/publico”**

DIR_DESTINO== “ruta de destino” define la ruta de destino. En este caso **“/home/donguillermo/proyecto_empresa/procesos”**.

find “\$DIR_ORIGEN” encuentra archivos en el directorio de origen.

-type f se refiere a que sean archivos (f de file).

-cmin -5 significa que hayan sido creados en los últimos cinco minutos.

exec cp ejecuta el comando en el directorio de destino.

Usamos **chmod +x** seguido de la ruta del archivo hasta `/copiar_nuevos.sh` para hacerlo ejecutable.

```
root@DonGuillermo-VirtualBox:/home/donguillermo/proyecto_empresa/confidencial# chmod +x /home/donguillermo/proyecto_empresa/procesos/copiar_nuevos.sh
root@DonGuillermo-VirtualBox:/home/donguillermo/proyecto_empresa/confidencial# ls /home/donguillermo/proyecto_empresa/procesos/copia
```

Usamos **cat** seguido de la ruta otra vez para mostrar el contenido del archivo.

```
root@DonGuillermo-VirtualBox:/home/donguillermo/proyecto_empresa/confidencial# cat /home/donguillermo/proyecto_empresa/procesos/copiar_nuevos.sh
#!/bin/bash

DIR_ORIGEN="/home/donguillermo/proyecto_empresa/publico"
DIR_DESTINO="/home/donguillermo/proyecto_empresa/procesos"

find "$DIR_ORIGEN" -type f -cmin -5 -exec cp {} "$DIR_DESTINO" \;
```

Ahora introducimos el comando **crontab -e** para entrar en la herramienta crontab. Esto nos permite ejecutar tareas repetitivas para que se ejecuten de manera automática en segundo plano. Pulsamos **1** para elegir el editor nano, que es el más fácil de usar.

```
root@DonGuillermo-VirtualBox:/home/donguillermo/proyecto_empresa/confidencial# crontab -e
no crontab for root - 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
```

En el fondo escribimos **`* /5 * * * * /bin/bash`**
`/home/donguillermo/proyecto_empresa/procesos/copiar_nuevos.sh`.

`* /5 * * * *` ejecuta el comando cada cinco minutos.

`/bin/bash /home/donguillermo/proyecto_empresa/procesos/copiar_nuevos.sh`.
 especifica la ruta completa en la que se va a ejecutar el comando bash.


```

GNU nano 6.2 /tmp/crontab.sURm5F/crontab
# 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
*/5 * * * * /bin/bash /home/donguillermo/proyecto_empresa/procesos/copiar_nuevos.sh

```

Esto debería poner en marcha el script. Por último, vamos a comprobar que está funcionando. Esto se hace escribiendo `grep CRON /var/log/syslog` para acceder al fichero donde se guardan todos los sucesos que ocurren en el sistema. Como podemos ver en las partes subrayadas, cada cinco minutos se produce la actividad que hemos programado.

```

root@DonGuillermo-VirtualBox:/home/donguillermo/proyecto_empresa/confidencial# grep CRON /var/log/syslog
Nov 29 13:33:20 DonGuillermo-VirtualBox cron[540]: (CRON) INFO (pidfile fd = 3)
Nov 29 13:33:20 DonGuillermo-VirtualBox cron[540]: (CRON) INFO (Running @reboot jobs)
Nov 29 14:17:01 DonGuillermo-VirtualBox CRON[2555]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)
Nov 29 14:30:01 DonGuillermo-VirtualBox CRON[2740]: (root) CMD ([ -x /etc/init.d/anacron ] && if [ ! -d /run/systemd/system ]; then
/usr/sbin/invoke-rc.d anacron start >/dev/null; fi)
Dec 14 00:17:01 DonGuillermo-VirtualBox CRON[48652]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)
Dec 14 01:17:01 DonGuillermo-VirtualBox CRON[48750]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)
Dec 14 01:25:01 DonGuillermo-VirtualBox CRON[48771]: (root) CMD (/bin/bash /home/donguillermo/proyecto_empresa/procesos/copiar_nuevo
s.sh)
Dec 14 01:25:01 DonGuillermo-VirtualBox CRON[48770]: (CRON) info (No MTA installed, discarding output)
Dec 14 01:30:01 DonGuillermo-VirtualBox CRON[48778]: (root) CMD (/bin/bash /home/donguillermo/proyecto_empresa/procesos/copiar_nuevo
s.sh)
Dec 14 01:30:01 DonGuillermo-VirtualBox CRON[48777]: (CRON) info (No MTA installed, discarding output)
Dec 14 01:35:01 DonGuillermo-VirtualBox CRON[48784]: (root) CMD (/bin/bash /home/donguillermo/proyecto_empresa/procesos/copiar_nuevo
s.sh)
Dec 14 01:35:01 DonGuillermo-VirtualBox CRON[48782]: (CRON) info (No MTA installed, discarding output)
Dec 14 01:40:01 DonGuillermo-VirtualBox CRON[48791]: (root) CMD (/bin/bash /home/donguillermo/proyecto_empresa/procesos/copiar_nuevo
s.sh)
Dec 14 01:40:01 DonGuillermo-VirtualBox CRON[48790]: (CRON) info (No MTA installed, discarding output)
Dec 14 01:45:01 DonGuillermo-VirtualBox CRON[48799]: (root) CMD (/bin/bash /home/donguillermo/proyecto_empresa/procesos/copiar_nuevo
s.sh)
Dec 14 01:45:01 DonGuillermo-VirtualBox CRON[48797]: (CRON) info (No MTA installed, discarding output)
Dec 14 01:50:01 DonGuillermo-VirtualBox CRON[48838]: (root) CMD (/bin/bash /home/donguillermo/proyecto_empresa/procesos/copiar_nuevo
s.sh)
Dec 14 01:50:01 DonGuillermo-VirtualBox CRON[48835]: (CRON) info (No MTA installed, discarding output)

```

Con esto, la tarea está completada.