

## EJERCICIOS DE LINUX.

### Relación – 2

1.- Introduce un comando para dar permisos de ejecución a todos los usuarios en todos los ficheros de \$HOME. Este comando debe guardar los resultados en el fichero res1 y los errores en el fichero err2. Además debe lanzarse en background.

```
chmod +R a+x $HOME >res1 2>err2 &
```

2.- Introduce un comando para almacenar la información detallada de todos los procesos del sistema en el fichero procesos\_1. Los procesos deben estar ordenados por nombre de comando.

```
ps -ef | tr -s ' ' | sort -t ' ' -k8 >procesos_1
```

3.- Introduce un comando para ejecutar en segundo plano un comando que se quede en pausa durante 60 minutos. Introduce un comando para verificar si el comando que has lanzado sigue activo en memoria.

```
Sleep 60m &  
ps -ef | grep sleep
```

4.- Introduce un comando para localizar el PID del proceso anterior. Seguidamente introduce otro comando para eliminar dicho proceso.

```
ps -ef | grep sleep  
kill -9 PID
```

5.- Introduce un comando para almacenar la información sobre la memoria del sistema, en el fichero memoria\_1.

```
free > memoria_1
```

6.- Introduce un comando para que el sistema ejecute mañana a las 20 horas y 10 minutos, el una orden para enviar al fichero res1, el contenido de los ficheros del directorio \$HOME, mayores de 1KBytes.

```
nano orden  
find $HOME-size +1k -exec cat {} > $HOME/res1
```

Cerramos el fichero y ejecutamos:

```
at -f orden 20:10 021509 # suponiendo que estemos a 14/2/2009
```

7.- Introduce un comando para que el sistema ejecute el día 30 de este mes, una tarea para borrar los ficheros del directorio \$HOME, mayores de 1GB La salida de error de este comando debe almacenarse en el fichero /root/errores\_copia, sin sobrescribir su contenido previo.

```
nano orden  
find $HO -size +5G -exec rm {} + 2> > /root/errores
```

Cerramos el fichero y ejecutamos:

```
at -f orden 20:00 033016 # suponiendo que estemos a 14/3/2016
```

8.- Introduce el comando necesario para mostrar la cola de tareas programas con el comando at.

Atq

9.- Crea una tarea periódica que almacene el estado de la memoria del sistema en un fichero llamado memoria\_2, sin eliminar su contenido previo.

La tarea debe ejecutarse todos los viernes de cada mes a las 22:00.

Nano orden

```
00 22 * * 5 free >> $HOME/memoria_2
```

Cerramos el fichero y ejecutamos:

crontab orden

10.- Crea una tarea periódica que se ejecute todos los días a las 8:00 y almacene en el fichero /root/usuarios\_conectados, la lista de usuarios que han entrado al sistema.

Nano orden

```
00 08 * * * who >> /root/usuarios_conectados
```

Cerramos el fichero y ejecutamos:

crontab orden

11.- Crea una tarea periódica que se ejecute cada 10 minutos y almacenen en el fichero /root/procesos, la lista de procesos existentes en memoria, ordenada por tiempo de ejecución.

Nano orden

```
00,10,20,30,40,50 * * * * ps -ef | tr -s ' ' | sort -t ' ' -k7
```

Cerramos el fichero y ejecutamos:

crontab orden

12.- Explica qué es lo que hace el comando kill.

Mirar el manual de comandos.

13.- Muestra en pantalla un monitor con la información detallada de los procesos que se están ejecutando en el sistema.

top

14.- Muestra un árbol con la jerarquía de los procesos de memoria.

pstree

15.- Introduce comandos para realizar las siguientes tareas:

- Desde una consola del modo gráfico, lanza un proceso en segundo plano,

que espere durante 1 hora.  
Sleep 1h &

- Desde otra consola gráfica, envía la señal SIGTERM sobre el proceso anterior.

Kill -15 PID

- Desde la segunda consola manda una señal SIGTERM sobre la primera consola. ¿Que ocurre?. ¿Como y por qué, podríamos cerrar dicha consola por medio de una señal?

kill -9 PID

16.- Introduce un comando para mostrar en pantalla la información detallada de todos los procesos de root existentes en el sistema.

```
ps -f -u root
```

17.- Introduce un comando para mostrar en pantalla el nombre y PID de los procesos lazados en primer plano.

```
ps -ef | grep + | tr -s ' ' | cut -d ' ' -f2,8
```

18.- Introduce un comando para obtener en pantalla el PID de los procesos de sistema en estado suspendido.

```
ps -e u | tr -s ' ' | grep S | cut -d ' ' -f2
```

19.- Introduce un comando para mostrar el PID de todas las consolas modo texto existentes en el sistema.

```
ps -ef | grep tty | tr -s ' ' | cut -d ' ' -f2
```

20.- Introduce un comando para mostrar los procesos de tu usuario ordenados por tiempo de ejecución.

```
ps -f -u root | tr -s ' ' | sort -t ' ' -k7 -r
```

21.- Introduce una tarea periódica que almacene cada 20 minutos el espacio libre y ocupado de la memoria en el fichero /root/memoria\_libre, sin sobrescribir su contenido.

Nano orden

```
00,20,40 * * * * free >> /root/memoria_libre
```

Cerramos el fichero y ejecutamos:

crontab orden