BASH

THE BOURNE-AGAIN SHELL

Angel Luis Calvo

ESCUELA IT

En esta sesión:

- Temporización
- AT
- CRON
- Casos prácticos

Angel Luis Calvo

Angel Luis Calvo

angelonx@gmail.com

ESCUELA IT

Temporización

Angel Luis Calvo

ESCUELA IT

Temporización

La automatización asegura una importante liberación de tareas administrativas, y a la larga, un mejor aprovechamiento de los recursos disponibles (siempre que esté bien hecha)

Uno de los pilares fundamentales reside en la temporización de trabajos, la programación de tareas a horarios y fechas establecidas.

Temporización

Para las temporizaciones habrá que tener en cuenta la importancia de los relojes de los sistemas y su sincronización.

Para ello contamos con el protocolo NTP, que permite a un equipo cliente sincronizarse con un servidor horario.

Con el comando date se puede establecer la hora, pero no es lo más recomendable.
date +%Z → para conocer la zona horaria del sistema

Temporización

En distribuciones como Ubuntu se utiliza timedatectl, permite establecer huso horario y sincronizar por red la hora entre otras cosas.
Se puede usar el comando ntp (instalándolo) para acciones más avanzadas.
Su archivo de configuración es /etc/ntp.conf
En ese archivo figura una lista de servidores, que podría ser en España:

server 0.hora.roa.es
server 1.hora.rediris.es
...

Temporización

Si se quiere se puede incluir una sentencia server en ese archivo:
server *192.168.24.15* → señalando a un servidor de la empresa que esté sincronizado con el exterior

Comandos ilustrativos:
ntpq -p → para ver estado de los servidores horarios
ntpdate -d *servidor* → para forzar la sincronización con un servidor.
tzselect → para seleccionar otra zona horaria
watch ntpq -p → lista de servidores, el * es con el que está sincronizado, *reach* indica la conectividad hacia el servidor.

Temporización

OJO:

- ¡La hora del reloj hardware, marcado por la BIOS, no tiene por qué coincidir con la del sistema!
- Esto puede crear problemas (muchos logs usan ese reloj para sus marcas)

Solución: sincronizar ambos, normalmente la del sistema será la buena, o debería ser :-)

hwclock --show → para ver la hora del hardware

hwclock -w → para que coja la del sistema

Ejercicio

Crear un script consulta.sh, que consulte las zonas horarias de dos equipos y calcule la diferencia de horas entre ambos. (Deberá autenticarse en ssh a no ser que lo tenga ya sin pedir contraseña)

```
#!/bin/bash
valor_mi_zona=$(date +%z)
mi_zona=$(date +%Z)
matriz=($(ssh jefe@192.168.15.24 "bash -c" "'date +%z; date +%Z'"))
#bash -c asegura que los comandos permanezcan en nuestra terminal
valor_otra_zona=${matriz[0]}
su_zona=${matriz[1]}
diferencia=$((${valor_mi_zona:1}-${valor_otra_zona:1})) #El 1 es para eliminar el + que introduce date
echo La diferencia de horas es $diferencia horas
echo Nosotros estamos en la zona $mi_zona y ellos en $su_zona
echo FIN
```

Temporización

Para automatizar algo puede echarse mano de comandos conocidos. como *sleep*.

- Este comando introduce retardos temporales que podemos aprovechar, junto con comandos condicionales.
- Por ejemplo, generando un bucle infinito del que se salga cuando se produzca un evento.

Ejercicio

Crear un script copia_con_aviso.sh, que copie un archivo en otro, siempre que el destino exista, en caso contrario debe mandar un aviso de que el destino no está disponible, y mandar un mensaje cada 5 minutos mientras dure esta situación, hasta que por fin ya pueda hacerlo.

```
#!/bin/bash
#copiar un archivo en otro, si el destino existe, si no esperar 5 min.
while : ; do
  if [ -f "$2" ] ; then
            cp -v "$1" "$2"
            echo Archivo copiado
            exit 0 #Aquí rompemos el bucle infinito
      else
            echo No se ha podido hacer aún la copia!
            sleep 5m
  fi
done
```


BASH
THE BOURNE-AGAIN SHELL

ESCUELA IT

AT

Angel Luis Calvo

ESCUELA IT

AT

El comando *at* nos permitirá programar una tarea para que se ejecute en un determinado momento.
Consideraciones:

- No está pensado para realizar programaciones periódicas.
- No está pensado para su uso en archivos de configuración.
- No suele venir pre-instalado → sudo apt install at

AT

Al ejecutar el comando se introduce en un prompt de configuración.
Comandos:
at 11:00 → se debe especificar al menos la hora de ejecución, luego se indican los comandos
at>touch prueba # a las 11:00 se ejecutará esto, cuando se termina de indicar comandos hay que pulsar CTRL-D
atq → lista de trabajos programados
atrm → para eliminar programados

AT

Ejemplos:

```
at 12:00 PM tomorrow #dispone de diversos comandos temporales
at> cp *.txt carpeta1 #se pueden poner comandos línea a línea
at> touch archivo1 ; rm borrar.txt #o varios en una
at> echo hola > saludo.txt #al ejecutarse en otro proceso, los mensajes
                                    por consola se pierden
at> mi_script.sh #se puede programar la ejecución de un script
```

Ejercicio

Crear un script at_ejemplo.sh, que copie un archivo at_prueba.txt, en una subcarpeta del curso llamada temporal. En caso de que se produzca un error, se desea que programe automáticamente una tarea para que realice la copia a las 13:00.


```
#!/bin/bash
#para copiar archivos especificos, que si da un error, programe una tarea
#con at, para hacerlo más tarde
cp -v at_prueba.txt /home/$USERNAME/curso/temporal/ >> error.log

if [ $? -ne 0  ] ; then
        echo Hubo un error de copia >> error.log
        echo lo intentaremos a las 13:00
        echo "$0" | at 13:00
else
        echo Copia realizada >> error.log
fi
```

ESCUELA IT

AT

Si consultamos los procesos en ejecución, podremos encontrar el de at, si intentamos matarlo se genera otro nuevo.
Debe cancelarse con atrm *nº*

```
ps aux | grep atd
daemon 6716 0.0 0.0  3792  2544 ?        Ss   10:42   0:00 /usr/sbin/atd
…
```

Lo que sí se puede es parar el servicio atd con systemctl stop atd, para que deje de funcionar

BASH

THE BOURNE-AGAIN SHELL

CRON

Angel Luis Calvo

ESCUELA IT

CRON

Se trata de un *daemon* que lanza las tareas programadas.
Se ejecuta continuamente en 2º plano.
Hay dos formas de configurarlo:

- Por usuario
- En general

Qué es un daemon:
https://es.wikipedia.org/wiki/Daemon_(inform%C3%A1tica)

CRON

Todas las tareas las ejecuta en 2º plano.

Para configurarlo en general editaríamos (con permisos de root) el archivo /etc/crontab
Es un archivo de texto, donde se pueden ir incluyendo líneas, en cada una de las cuales habría una tarea.

Además de ser un archivo, *crontab* también es un comando.

CRON

Para hacer una configuración por usuario, hay que utilizar el comando crontab -e durante la sesión de dicho usuario.

- Genera un archivo de configuración en el directorio /var/spool/cron/crontabs/ con el nombre del usuario (protege el principal)
- Abre el archivo en un directorio temporal, si sucede algo inesperado no se guardará.

Al ejecutarlo puede ofrecerte elegir un editor, por defecto suele usar vim.
(se almacena en ~/.selected_editor)

CRON

El archivo /etc/crontab tendrá una línea por cada tarea programada con la siguiente estructura:
5 campos de tiempo (minuto - hora - día del mes - mes - dia de la semana) seguido del usuario que realiza esa tarea y el comando a ejecutar (que puede ser un programa o un script)
Ejemplo:
30 22 8 2 sat root rm /tmp/temporal.txt #El sábado 8 de febrero a las 22:30 se borrará el archivo temporal.txt usando la cuenta de root

CRON

Ejemplos especiales:

20 9 1,15,30 * * #Esta combinación de valores de tiempo hacen referencia a todos los días 1, 15 y 30 de cualquier mes, a las 9:20 horas

*/15 * * * 1,3,5 #Se ejecuta los lunes, miércoles y viernes cada 15 minutos

* 5 * * mon #Cada minuto de la hora 5 de cada lunes

00 11 * 7-9 0 #A las 11:00 de todos los domingos desde julio a septiembre

00 * 1-31/2 dec * #Cada hora en punto de cada dos días de diciembre

CRON

Todas las tareas se ejecutarán mientras los usuarios tengan permisos sobre ellas.
Si el equipo está apagado, cron no puede retomar la tarea.
Para retomar tareas, fuera de plazo (asíncronas) está *anacron*.

OJO: Los scripts deben probarse antes de automatizarlos

CRON

Consideraciones:

- Si se utiliza el comando crontab -e, no hay que especificar el usuario
- Los errores de cron se pueden analizar en /var/log/syslog, o en /var/log/cron

Ejemplo de crontab -e

```
# 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 * * * * /home/jefe/Escritorio/scripts/muestro.sh
```

CRON

Más consideraciones:

- En /etc/crontab hay predefinidas tareas para que podamos poner scripts en carpetas especiales, para que se ejecuten diaria, semanal y mensualmente, y a cada hora (/etc/cron.hourly/, etc.)
- Podemos poner enlaces ahí, teniendo en cuenta que en al menos Debian y derivados no admiten nombres con punto. Esto es debido al comando run-parts.

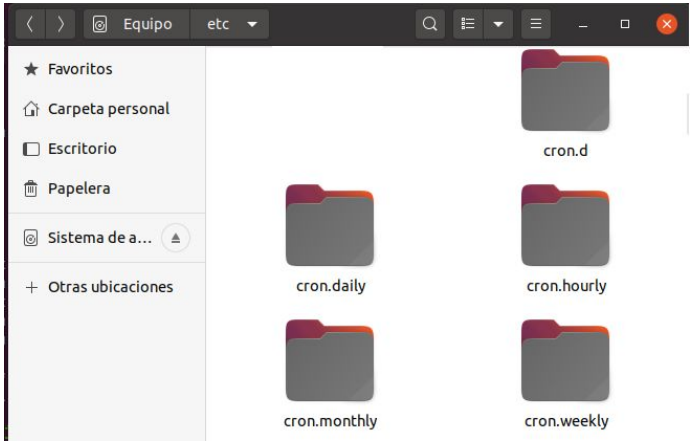
Ejemplo de /etc/crontab

```
# |  |  |  |  |
# *  *  *  *  *  user-name command to be executed
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
```

CRON

- Deberíamos usar el principal /etc/crontab solo para tareas que requieren el usuario root.
- El directorio /etc/cron.d/ también nos sirve para incluir archivos crontab, evitando tocar el principal.

Carpetas vistas desde Nautilus



CRON

- Ojo a las rutas de los archivos, pueden acabar apareciendo los resultados en el home del usuario.
- Ojo a los permisos de usuario de los archivos que se generen en esas tareas.
- No es necesario reiniciar el servicio cada vez que se configura.

Ejercicio

Plantear un script y una tarea, para que todos los días, a las horas en punto y a y media, se compruebe una carpeta y si contiene archivos los mueva a otra ubicación.

```
#!/bin/bash
#Comprobaremos la carpeta curso/comprobar/ y si tiene archivos los llevaremos hacia curso/depósito
#Fijamos variables de directorios para facilitar el manejo
directorio=/home/jefe/curso/comprobar/
destino=/home/jefe/curso/deposito/
#Comprobación
vacio=$(ls "$directorio")
if [ "$vacio" = "" ] ; then
        echo Directorio vacío
  else
        echo Directorio ocupado
        mv "$directorio"* "$destino"
        echo Movidos
```


---------- Continúa en la siguiente----------

BASH
THE BOURNE-AGAIN SHELL

ESCUELA IT

Ejercicio

---------- Viene de la anterior ----------

```
fi
```

Configuración de *cron*: (haciendo crontab -e)

```
00,30 00-23 * * * /etc/cron.d/mover.sh     #En los minutos valdría */30
```

ESCUELA IT

CRON

Para la configuración existen opciones abreviadas:
@reboot → Se ejecuta una vez, al arrancar el sistema
@yearly → Se ejecuta sólo una vez al año, sería como: 0 0 1 1 *
@hourly → Al primer minuto de cada hora: 0 * * * *

Y algunas más

Más info:
https://manpages.debian.org/buster/cron/crontab.5.en.html

CRON

Las tareas cron pueden estar configuradas en múltiples sitios.
Para comprobar cuáles hay:
cron -l → lista las del usuario actual
sudo cron -u *usuario* -l → lista las de otro usuario
sudo ls -1 /var/spool/cron/crontabs → para saber qué usuarios tienen tareas
Y al final, comprobar los directorios /etc/cron.d y compañeros

Ejercicio

Hacer un script que liste todas las tareas programadas por usuarios mediante cron.

```
#!/bin/bash
#Para ver tareas programadas de todos
matriz_user=($(ls -1 /var/spool/cron/crontabs))

for ((i=0; i<${#matriz_user[*]}; i++))
do
        echo Usuario ${matriz_user[$i]} >> encontrados
        crontab -u ${matriz_user[$i]} -l | grep -v \# >> encontrados
done
echo Las de usuarios son:
cat encontrados
echo El resto
matriz_carpetas=(/etc/cron.*)
```


---------- Continúa en la siguiente----------

ESCUELA IT

Ejercicio

---------- Viene de la anterior ----------

```
for ((j=0; j<${#matriz_carpetas[*]}; j++))
do
        ls -l ${matriz_carpetas[$j]}
done
echo FIN
```

ESCUELA IT

CRON: anacron

Cuando los equipos están apagados, cron no puede hacer nada.
Al volver a encenderlos, cron no se preocupa.
Habrá tareas de cron dedicadas a lanzar anacron, para que éste recupere tareas perdidas.

Se configura en /etc/anacrontab
Su sintaxis:
Periodo - tiempo de espera - identificador - comando

CRON: anacron

Ejemplo de anacrontab:

```
1     5     cron.daily    run-parts --report /etc/cron.daily
7     10    cron.weekly       run-parts --report /etc/cron.weekly
@monthly    15    cron.monthly      run-parts --report /etc/cron.monthly
```

Los periodos indican días, salvo las cláusulas con @. El tiempo de espera indica por ejemplo que si una tarea diaria no se ejecutó, se espera 5 minutos desde el arranque para hacerlo.

CRON: anacron

Las fechas de la última ejecución se almacenan en /var/spool/anacron

anacron -f → para forzar a que se ejecuten las tareas aunque no estén pendientes

/etc/cron.d/anacron → aquí cron define la tarea para que anacron se ejecute todos los días.

Ejercicio

Establecer una tarea que compruebe el espacio en disco, generando un registro con el tamaño ocupado cada hora.

```
#!/bin/bash
#Este script comprueba el espacio libre en disco
#Lo ejecutaremos en el minuto 21 de cada hora
#Haremos un crontab en /etc/cron.d/ para ello
espacio=$(df -h | grep sda1 | tr -s " " | cut -d " " -f 4)
echo Queda libre $espacio >> /home/jefe/Escritorio/scripts/espacio.log
date >> /home/jefe/Escritorio/scripts/espacio.log
```


En ***cron_espacio***:

```
21 00-23 * * * jefe /home/jefe/Escritorio/scripts/espacio_disco.sh
```

Casos prácticos

Angel Luis Calvo

ESCUELA IT

Casos prácticos

Para la realización de diversos casos prácticos es necesario conocer otras herramientas y procedimientos.

- Sistemas de archivo
- Envío de correos
- Unidades de almacenamiento
- Comandos de redes

Casos prácticos: mount

El comando mount se utiliza para montar sistemas de ficheros en la estructura de directorios de Linux.

- Para GNU/Linux las unidades físicas se manejan como archivos
- Unidades: discos, CD, USB, particiones, unidades de red,...
- Cada cosa debe ir montada en un directorio
- Solo hay un raíz (/) en el árbol de directorios del sistema

Casos prácticos: mount

Sintaxis:
mount [-*parámetros*] [-t *tipo_sistema_archivo*] [-o *opciones*] *dispositivo punto_de_montaje*

Tipos de sistemas de archivo (habituales):
ext4 → muy utilizado en particiones Linux
ntfs → NTFS de Windows
nfs → sistema de red
smbfs ó cifs → sistema de red samba
tmpfs → sistema de disco en RAM

Casos prácticos: mount

Ejemplo:
Carpeta compartida por un servidor samba, llamada datos
Punto de montaje en nuestro sistema: ~/servidor

```
sudo mount -t cifs -o username=angel,password=angel //192.168.10.12/datos ~/servidor
```

Si accedemos a nuestro directorio servidor, veremos lo que hay compartido en el servidor samba.

Casos prácticos: mount

Para comprobar las unidades que hay montadas se ejecuta mount.
Para desmontar usamos umount.

Ojo: las unidades montadas directamente con este comando, se desmontan automáticamente al reiniciar el equipo.

Solución:

- Crear un script que arranque con el equipo.
- Usar el archivo de sistema /etc/fstab

Ejercicio

Mediante un script, copiar los archivos seleccionados como parámetro en el servidor samba, montando previamente la unidad. (Debe ejecutarse con sudo y tener creado previamente el punto de montaje)

```
#!/bin/bash
#copiaremos el primer parámetro en el servidor samba
#antes montaremos la unidad
if [ $# -eq 0 ]
   then
        echo No pusiste parámetro
        exit 1
fi
mountpoint /home/jefe/servidor #Comprobamos si ya está montado para evitar duplicidades
if [ $? -eq 0 ]
   then
        echo Ya está montado
```


---------- Continúa en la siguiente----------

ESCUELA IT

Ejercicio

```
-------------------------------- Viene de la anterior ------------------------------------
  else
        echo No está montado aún
        mount -t cifs -o username=angel,password=angel //192.168.15.24/datos /home/jefe/servidor
                if [ $? -ne 0 ]
                then
                        echo Error de montaje
                        exit 2
                fi
fi
while [ $# -gt 0 ]
  do
        cp -v "$1" /home/jefe/servidor
        shift
done
```

ESCUELA IT

Casos prácticos: mount

Sintaxis de /etc/fstab
Unidad a montar - Punto de montaje - Sistema de archivo - Opciones (solo lectura, usuario, etc.) - Usar *dump* (0 ó 1) - Uso de *fcsk* (0,1,2)

Ejemplo:
//192.168.20.10/carpeta_remota /home/usuario/punto_de_montaje cifs username=pepe,password=pepe,rw,iocharset=utf8 0 0

Si se añade una entrada se puede cargar sin reiniciar mediante:
mount -a

Casos prácticos: disco RAM

Un disco RAM es un trozo de RAM que se utilizará como si fuese una partición de disco duro normal.

Ventajas:

- Velocidad
- Menor desgaste de disco físico

Desventajas:

- Volatilidad
- Tamaño limitado

Casos prácticos: disco RAM

Para conseguirlo basta con realizar el montaje.
En el fstab:
El dispositivo puede tener cualquier nombre, lo habitual es montarlo en /tmp, hay que crear la subcarpeta
discoram /tmp/ram tmpfs defaults,size=*512M*

Para montarlo sin hacerlo permanente:
sudo mount -t tmpfs -o size=512m *discoram /tmp/ram*

Para comprobarlo podemos ejecutar mount

Ejercicio

Desarrollar un script que libere periódicamente el contenido del disco RAM, salvándolo en un servidor remoto, a la 1, a las 5 y a las 13:00.
Lo hará creando directorios con la marca del día, previendo que no se muevan los archivos que estén abiertos.

```
#!/bin/bash
# para mover los archivos de una unidad RAM
# Revisa el contenido de esa unidad y lo pasa a disco duro
ram="/tmp/ram"
servidor="/mnt/servidor" #unidad remota, previamente montada
dia=$(date +%F)
cd $ram
for i in $(ls -1) ; do
        lsof  $i &> /dev/null #comprueba los archivos abiertos para no moverlos todavía
                if [ $? -ne 0 ] ; then #si es cero el archivo está abierto
```


---------- Continúa en la siguiente----------

Ejercicio

---------- Viene de la anterior ----------

```
                mkdir -p $servidor/$dia #-p para que no proteste si ya existe
                mv $i $servidor/$dia
        fi
done

#Para programarlo, a las 1, 5 y 22, por ejemplo todos los días
# en el crontab 00 1,5,22 * * *
```

ESCUELA IT

Casos prácticos: disco RAM

Es importante controlar el rendimiento y el uso de memoria.
Para comprobar que funciona, además de mount, podemos usar free para ver la cantidad de memoria disponible.

Otros comandos interesantes son top y htop

Casos prácticos: e-mail

Es muy interesante poder enviar correos de aviso por incidencias.
Lo podemos hacer vía comando con mail.

La sintaxis, entre otras opciones:
mail -s “asunto” dirección@correo_destino <<< “cuerpo del mensaje”

Se puede tener un texto preconfigurado y redirigirlo:
mail -s “asunto” dirección@correo_destino < Archivo_mensaje.txt

Ejercicio

Script que comprueba por ping el estado de un servidor. Si lo detecta caído avisará por correo. Se ejecutará cada 30 minutos, entre las 5:00 y las 20:00.

```
#!/bin/bash
#Ejecutamos 5 vecesping a intervalos de 2 segundos si falla
#para evitar fallos momentáneos
servidor=192.168.15.24
contador=0
fecha=$(date)
for num in {1..5} ; do
        ping -c 1 "$servidor" &> /dev/null
        if [ $? -eq 1 ]; then
                sleep 2 #falló el ping
                contador=$(($contador + 1))
        fi
done
```


---------- Continúa en la siguiente----------

ESCUELA IT

Ejercicio

```
-------------------------------- Viene de la anterior ------------------------------------
if [ "$contador" -eq 5 ]; then
        echo Servidor caído "$fecha" >> /home/jefe/alerta_servidor
        mail -s "Servidor caído" micorreo@gmail.com <<< "Alerta de caída"
fi

#En el cron: */30 5-20 * * *
```

ESCUELA IT

Casos prácticos: e-mail

Hay que tener en cuenta que se debe configurar el cliente de correo con una cuenta válida, por ejemplo, de *Gmail*. Hay que permitir en ese caso las aplicaciones no seguras.

Es bueno contar con una cuenta que se use solamente para esto.

Para más info:
https://juantrucupei.wordpress.com/2016/07/13/enviar-correo-gmail-con-exim4/

Ejercicios

42_

1.- Crear un script que realice lo siguiente: copiar los archivos indicados como parámetro, en el directorio curso/mis_copias/. Este directorio es un punto de montaje de un servidor, donde siempre hay un archivo índice llamado index.txt. Si no se encontrara este archivo, deberá intentar montar la unidad otra vez. Si falla deberá programar una tarea para que lo vuelva a hacer al cabo de una hora.

2.- En una unidad RAM, además de archivos, hay carpetas con el nombre de la fecha del día en que fueron creadas. Crear un script para que libere la unidad de dichas carpetas, moviéndolas a un directorio distinto. Asegurarse antes de que las carpetas no contienen archivos ocupados. Programar una tarea para que esto se haga cada hora de 12:00 a 20:00.

43_

3.- Crear un script que compruebe el uso de memoria, de tal forma que si es inferior a 512MB libres, vacíe la unidad RAM, poniendo todos sus archivos en el disco, y desmonte dicha unidad. El script debe ser capaz de programar una tarea en cron para que haga esta comprobación cada hora, si es que no estuviese programado ya en /etc/cron.d/ram_cron

44_

BASH
THE BOURNE-AGAIN SHELL

ESCUELA IT

1.-

```
#!/bin/bash
#Copiar los archivos del primer parámetro en curso/miscopias
#Reintentar si falla el montaje al cabo de una hora
if [ $# -eq 0 ]
        then
        echo Faltan parámetros
        exit 1
fi
#variables para ir usando
hora_actual=$(date +%H)
hora=$(($hora_actual + 1))
minuto=$(date +%M)
carpeta=/home/jefe/curso/mis_copias/
#ejecución
if [ -f “$carpeta”index.txt ]; then
        while [ $# -ge 1 ]
```

---------- Continúa en la siguiente----------

ESCUELA IT

---------- Viene de la anterior ----------

```
        do
                echo Copiando archivo
                cp -v $1 $carpeta
                shift
done
  else
        mount -a
        if [ ! -f “$carpeta”index.txt ]; then
                echo "$0" | at $hora:$minuto
        fi
fi
```

ESCUELA IT

2.-

```
#!/bin/bash
# para mover carpetas de una unidad RAM
ram="/tmp/ram"
directorio="/home/jefe/curso"
cd $ram
ls -1d */ > listado.borrable
while read dia
do
        for i in $(ls -1 $ram/$dia*) ; do #movemos los archivos
        lsof $i &> /dev/null
                if [ $? -ne 0 ]; then #0 si archivo abierto
                        mkdir -p $directorio/$dia
```

---------- Continúa en la siguiente----------

---------- Viene de la anterior ----------

```
                    mv $i $directorio/$dia
                fi
        done
        rmdir $dia &> /dev/null #si queda vacía, la borramos
done < listado.borrable
```

En el cron:

```
00 12-20 * * *
```

ESCUELA IT

3.-

```
#!/bin/bash
#Comprobar uso de memoria, para liberar RAM si es necesario
archivo_crontab=/etc/cron.d/ram_cron
touch $archivo_crontab
nombre=${0:2} #eliminamos el ./ de $0
mem_libre=$(free | grep Memoria | tr -s " " | cut -d " " -f 4)
liberar (){
mv /tmp/ram/* /home/jefe/curso/ram
umount /tmp/ram
}
tarea (){
cat $archivo_crontab | grep $nombre > linea_cron
if [ ! -s linea_cron ] ; then #archivo vacío
        echo "00 00-23 * * * root /home/jefe/curso/$nombre" >> $archivo_crontab
fi
}
```

---------- Continúa en la siguiente----------

ESCUELA IT


```
---------------------------------- Viene de la anterior -----------------------------------
if [ $mem_libre -lt 524288 ] #liberar memoria si bajamos de 512MB
   then
        liberar
   else
        echo No es necesario liberar
fi
tarea
```

ESCUELA IT

BASH

THE BOURNE-AGAIN SHELL

GRACIAS!!

Angel Luis Calvo

ESCUELA IT