

AUTOMATIZACIÓN DE TAREAS

Eduardo de Lamo Téllez

UT 05: Actividades automatización de tareas

1.- Describe las ventajas de la automatización de las tareas repetitivas en el sistema (teórica). En Microsoft hay dos conceptos clave implicados en la programación de una tarea ¿cuáles son?

- 1.- Permitir la ejecución de tareas repetitivas de forma automática y desatendida.
- 2.- Ahorro de tiempo y recursos.

El task scheduler de Windows se puede utilizar desde la línea de comandos con el programa `schtasks.exe`, pero lo más habitual es usar la interfaz gráfica de usuario hasta llegar a Programador de tareas. Los 2 conceptos clave implicados en la programación de una tarea en Windows son el desencadenante y la acción.

2.- Consulta en los apuntes facilitados el servicio **cron**. Después estudia la utilidad de determinados valores predefinidos tales como **@yearly**. Posteriormente investiga sobre la orden **run-parts** y para finalizar, estudia la utilidad de determinados directorios relacionados como **/etc/cron.hourly**.

Cron es el nombre del programa que permite a los usuarios de sistemas Unix y derivados planificar la ejecución de órdenes o programas de shell de manera automática en una fecha y tiempo específicos.

@yearly -> se ejecuta una vez al año.

@monthly -> una vez al mes.

@weekly -> una vez a la semana.

@daily -> una vez al día.

La orden `run-parts` sirve para ejecutar todos los scripts y programas situados en el directorio especificado.

```
antonio@xubuntu:~$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
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 )
#
```

/etc/cron.hourly -> cada hora

/etc/cron.daily -> cada día

/etc/cron.weekly -> cada semana

/etc/cron.monthly -> cada mes

UT 05: Actividades automatización de tareas

Como administradores del sistema, podemos guardar un archivo ejecutable por el shell (guión de shell, programa compilado, etc.) dentro de cualquiera de estos directorios para que sea ejecutado una vez cada hora, día, semana o mes, respectivamente, a la hora configurada en el fichero `/etc/crontab`.

3.- En el servidor Windows, automatiza la tarea que permita realizar copias de seguridad básicas con el comando **ROBOCOPY**.

Deberás crear un script que se ejecute todos los miércoles a las 9h y haga un **espejo** de esta estructura de directorios y ficheros.

Observación: suponemos que estamos en D: y el espejo va a E:

```
INFORMATICA
  propuestas.doc (4 KiB de tamaño)
  ASIR
    ASIR1
      modulos_asir1.txt (2 KiB)
    ASIR2
      modulos_asir2.txt (4 KiB)
      fct.txt (1 KiB)
  tríptico_asir.jpg (60 KiB)
  tríptico_dam.jpg (100 KiB)
  DAM
    DAM1
    DAM2
```

En su forma más básica solo copia los ficheros que encuentra en la primera carpeta donde le indiquemos.

```
C:\Users\Administrador>robocopy E:\INFORMATICA F:\DESTINO

ROBOCOPY      ::      Herramienta para copia eficaz de archivos

Inicio: lunes, 25 de febrero de 2019 9:47:03
Origen : E:\INFORMATICA\
Destino : F:\DESTINO\

Archivos: *.*
Opciones: *.* /DCOPY:DA /COPY:DAT /R:1000000 /W:30

100%      Nuevo dir          3      E:\INFORMATICA\
100%      Nuevo arch         4000      propuestas.doc
100%      Nuevo arch        60000      tríptico_asir.jpg
100%      Nuevo arch       100000      tríptico_dam.jpg
```

Opciones más interesantes:

`/S` -> Copia todas las carpetas que contengan ficheros en su interior.

`/E` -> Copia la estructura completa respetando los archivos que hay en destino.

`/MIR` -> Hace un espejo de la estructura origen en destino (si en destino hay algún fichero que no está en el origen se lo cepilla).

UT 05: Actividades automatización de tareas

```

C:\Users\Administrador>robocopy E:\INFORMATICA F:\DESTINO /MIR

ROBOCOPY      ::      Herramienta para copia eficaz de archivos

-----
Inicio: lunes, 25 de febrero de 2019 9:55:58
Origen  : E:\INFORMATICA\
Destino : F:\DESTINO\

Archivos: *.*

Opciones: *.* /S /E /DCOPY:DA /COPY:DAT /PURGE /MIR /R:1000000 /W:30

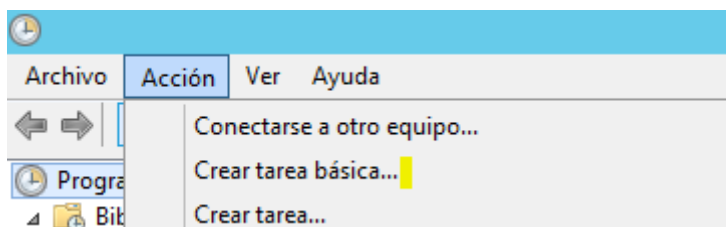
-----
3      E:\INFORMATICA\
0      E:\INFORMATICA\ASIR\
1      E:\INFORMATICA\ASIR\ASIR1\
2      E:\INFORMATICA\ASIR\ASIR2\
0      E:\INFORMATICA\DAM\
0      E:\INFORMATICA\DAM\DAM1\
*Archivo EXTRA      5      horario_dam1.txt
0      E:\INFORMATICA\DAM\DAM2\
  
```

Script.

```

espejo.cmd: Bloc de notas
Archivo Edición Formato Ver Ayuda
@echo off
ROBOCOPY E:\INFORMATICA F:\DESTINO /MIR
  
```

Programación del script con el programador de tareas de Windows.



Asistente para crear tareas básicas

Crear una tarea básica

Use este asistente para programar rápidamente una tarea común. Para obtener más configuraciones u opciones avanzadas, como desencadenadores o acciones de varias tareas, use el comando Crear tarea del panel Acciones.

Crear una tarea básica

Desencadenar

Acción

Finalizar

Nombre: Copia con ROBOCOPY

Descripción: Copia de seguridad programada con ROBOCOPY.

UT 05: Actividades automatización de tareas



Desencadenador de tarea

Crear una tarea básica	¿Cuándo desea que se inicie la tarea?
Desencadenar	<input checked="" type="radio"/> Diariamente
Acción	<input type="radio"/> Semanalmente
Finalizar	<input type="radio"/> Mensualmente
	<input type="radio"/> Una vez
	<input type="radio"/> Al iniciarse el equipo
	<input type="radio"/> Al iniciar sesión
	<input type="radio"/> Cuando se registre un evento específico



Diariamente

Crear una tarea básica	Inicio: 25/02/2019	10:11:21	<input type="checkbox"/> Sincronizar zonas horarias
Desencadenar			
Diariamente	Repetir cada: 1	días	
Acción			
Finalizar			

Crear una tarea básica	¿Qué acción desea que realice la tarea?
Desencadenar	
Diariamente	
Acción	<input checked="" type="radio"/> Iniciar un programa
Finalizar	<input type="radio"/> Enviar un correo electrónico (desusado)
	<input type="radio"/> Mostrar un mensaje (desusado)

Crear una tarea básica	Programa o script:	C:\Users\Administrador\Documents\espejo.cmd	Examinar...
Desencadenar			
Diariamente			
Acción			
Iniciar un programa	Agregar argumentos (opcional):		
Finalizar	Iniciar en (opcional):		

UT 05: Actividades automatización de tareas

Observación: Todos los scripts que hagas a continuación los diseñarás con root y los almacenarás en `/usr/local/bin`

4.- Hemos visto en Linux tres posibles estrategias sobre copias de seguridad que pueden ser implementadas con tar. Realizamos la estrategia primera: **copia completa diaria**.

Implementa las otras dos estrategias:

1ª: copia completa semanal los domingos y copia diferencial diaria.

2ª: copia completa semanal los domingos y copia incremental diaria.

Observación: para hacer pruebas vamos a crearnos un directorio en el que iremos almacenando información y que será el que respaldemos, por ejemplo `/home/antonio/datos`. La copia irá a `/copias`.

```
antonio@UD-01:~/datos$ ls -l
total 16
drwxr-xr-x 2 antonio antonio 4096 feb 17 22:26 asir1
drwxr-xr-x 2 antonio antonio 4096 feb 17 22:26 asir2
drwxr-xr-x 2 antonio antonio 4096 feb 17 22:26 dam1
drwxr-xr-x 2 antonio antonio 4096 feb 17 22:26 dam2
-rw-r--r-- 1 antonio antonio    0 feb 17 22:26 f1
-rw-r--r-- 1 antonio antonio    0 feb 17 22:26 f2
-rw-r--r-- 1 antonio antonio    0 feb 17 22:26 f3
```

Ayuda: consultar comando find y entre otras opciones, `-mtime` y `-newer`

Copia completa.

```
completa.sh x  diferencial.sh x  incremental.sh x
1  #!/bin/bash
2
3  cd /
4
5  tar czf /copias/datos$(date +%u).tgz datos
```

Copia diferencial (con la opción `-N` de tar).

```
diferencial.sh x
1  #!/bin/bash
2
3  cd /copias
4
5  # el formato adecuado para la opción -N de tar es Ej: 09-feb-15
6  # ||| buscar en man date para la salida correcta||| +%d-%b-%y'
7
8  # la completa es siempre la del domingo
9  fecha_copia_completa=`date +%d-%b-%y' -r datos7.tgz` 2>/dev/null
10
11  tar -czf /copias/datos$(date +%u).tgz /datos -N $fecha_copia_completa
```

UT 05: Actividades automatización de tareas

Copia incremental (con la opción -g de tar).

```
completa.sh x incremental.sh x diferencial.sh x
1 #!/bin/bash
2
3 cd /
4
5 tar czf /copias/datos$(date +%u).tgz -g /copias/incremental.snap datos 2>/dev/null
```

Programación de las copias con cron (sudo crontab -e).

```
.----- minuto (0-59)
| .----- hora (0-23)
| | .----- día del mes (1-31)
| | | .----- mes (1-12) o jan,feb,mar,apr,may,jun,jul... (meses en inglés)
| | | | .--- día de la semana (0-6) (domingo=0 ó 7) o sun,mon,tue,wed,thu,fri,sat (días en inglés)
| | | | |
* * * * * comando a ejecutar
```

Estrategias:

1ª: copia completa semanal los domingos y copia diferencial diaria.

```
# m h dom mon dow command
0 4 * * 0 /usr/local/bin/completa.sh
0 22 * * 1-6 /usr/local/bin/diferencial.sh
```

2ª: copia completa semanal los domingos y copia incremental diaria.

```
# m h dom mon dow command
0 4 * * 0 /usr/local/bin/completa.sh
0 22 * * 1-6 /usr/local/bin/incremental.sh
```

5.- Realizar un script, **alta_grupos.sh** que permita crear grupos a partir de un fichero que contiene el nombre del grupo.

Fichero grupos.txt

alumnado
profesorado
direccion
pas
secretaria
jefatura

Observación: sólo lo puede ejecutar root (UID = 0), el script debe recibir sólo un argumento y el fichero debe existir.

UT 05: Actividades automatización de tareas

```
alta_grupos.sh x
1  #!/bin/bash
2
3  # Comprobaciones
4
5  # Comprobar que el scrip está siendo ejecutado por root
6  if [ $UID -ne 0 ]
7  then
8      echo "Debes ser root"
9      exit 1
10 fi
11 # Que recibe un argumento
12 if [ $# -ne 1 ]
13 then
14     echo "Debe introducir 1 argumento."
15     exit 2
16 fi
17 # El fichero debe existir y ser un fichero regular
18 if [ ! -f $1 ]
19 then
20     echo "$1 no existe o no es un fichero."
21     exit 3
22 fi
23
24 # Algoritmo
25 for linea in `cat $1`
26 do
27     /usr/sbin/groupadd $linea
28 done
```

```
edu@US-ST:/usr/local/bin$ sudo alta_grupos.sh fichero_grupos.txt
```


UT 05: Actividades automatización de tareas

6.- Realizar un script, **alta_usuarios.sh** que permita crear usuarios a partir de un fichero (formato csv) que contiene el nombre de usuario, el grupo principal y la clave.

Fichero usuarios.csv

ricardo,alumnado,inves

julian,alumnado,inves

pepe,profesorado,inves

Observación: estudiar opciones -m -g del comando **useradd**

Observación: sólo lo puede ejecutar root (UID = 0), el script debe recibir sólo un argumento y el fichero debe existir.

```
1  #!/bin/bash
2
3  # Comprobaciones
4
5  # Comprobar que el scrip está siendo ejecutado por root
6  if [ $UID -ne 0 ]
7  then
8      echo "Debes ser root"
9      exit 1
10 fi
11 # Que recibe un argumento
12 if [ $# -ne 1 ]
13 then
14     echo "Debe introducir 1 argumento."
15     exit 2
16 fi
17 # El fichero debe existir y ser un fichero regular
18 if [ ! -f $1 ]
19 then
20     echo "$1 no existe o no es un fichero regular."
21     exit 3
22 fi
23
24 for usuario in $(cat $1)
25 do
26     nombre=`echo $usuario | cut -d"," -f1` #login
27     grupo=`echo $usuario | cut -d"," -f2` #grupo
28     clave=`echo $usuario | cut -d"," -f3` #password
29
30     /usr/sbin/useradd -m -s /bin/bash -g $grupo $nombre
31
32     echo "$nombre:$clave" | chpasswd --md5
33
34 done
```

```
edu@US-ST:/usr/local/bin$ sudo alta_usuarios.sh usuarios.csv
```

UT 05: Actividades automatización de tareas

7.- Especificar el formato de las líneas que habría que incluir en el archivo de crontab de un usuario para ejecutar de manera automática un guion de shell llamado `/usr/local/bin/alerta.sh`:

- a) **Cada día a las 05.00 h.**
`0 5 * * * /usr/local/bin/alerta.sh`
- b) **A las 14.15 h el primer día de cada mes.**
`15 14 1 * * /usr/local/bin/alerta.sh`
- c) **A las 22.00 h los días entre semana.**
`0 22 * * 1-5 /usr/local/bin/alerta.sh`
- d) **Cada 5 minutos.**
`*/5 * * * * /usr/local/bin/alerta.sh`
- e) **Cada dos horas.**
`* */2 * * * /usr/local/bin/alerta.sh`