

|                                    |                              |              |
|------------------------------------|------------------------------|--------------|
|                                    | © FLORIDA Centre de Formació |              |
| Cognoms i Nom / Apellidos y Nombre |                              | Data / Fecha |

## **Ejercicio de la 1ª evaluación**

Para superar la primera evaluación de ASI-II se plantea un ejercicio práctico que cada alumno deberá preparar y presentar al profesor la solución adoptada, respondiendo a las preguntas que este pudiera plantear sobre el mismo, y modificarlo para satisfacer las necesidades que el profesor estime oportunas durante la presentación/defensa.

### **El contexto:**

Hemos “heredado” un equipo con multitud de copias de seguridad desordenadas y repartidas por diferentes directorios del ordenador. Como resultado de esta situación, cuando intentamos buscar una imagen encontramos varias copias idénticas en diferentes ubicaciones; por lo tanto están consumiendo mucho espacio en nuestro disco y resulta complicado encontrar los originales deseados.

Para solucionar esta situación queremos consolidar todas estas copias de seguridad en una única copia, evitando los duplicados y desechando los archivos que no correspondan a imágenes interesantes. Para ello disponemos de un disco extraíble nuevo donde realizar esta copia. Por otra parte, queremos que esta situación no se reproduzca en un futuro y para ello tenemos que pensar en alguna forma de evitar que se generen múltiples copias de seguridad.

Por otra parte se desea preparar un segundo script que se ejecutará cada viernes a las 12:00 que se encargará de añadir a la copia de seguridad todos aquellos ficheros nuevos de los que no se tenga copia.

### **El ejercicio:**

Se requiere crear un proceso que permita generar una copia de todas las imágenes presentes en nuestro disco duro. Además se quiere que se ejecute de forma periódica todos los viernes a las 23:59. Y se desea que se evite la copia de duplicados (existe la posibilidad de que una misma imagen esté en varios directorios de nuestro disco). También se evitará la copia de archivos pequeños que corresponde a iconos o imágenes de muy baja calidad.

Con esta situación se plantearán dos necesidades: un programa que realice la copia de forma periódica y otro que nos permita realizarla de forma manual e incluya algunas funcionalidades.

## Requisitos:

Para atender estas dos necesidades crea dos scripts:

Uno para que se ejecute de forma periódica y otro para ser ejecutado de forma manual.

En ambos casos:

- Se debe configurar el tamaño mínimo de los archivos para ser copiados.
- Se copiará los ficheros "jpg" de todo el disco local al directorio correspondiente.
- No se copiarán duplicados de un mismo fichero, pero se generará un log donde se indiquen las copias rechazadas por ser copias ya existentes
- Se considerará que un fichero es idéntico a otro, si se cumplen estas dos condiciones:
  - Tienen el mismo nombre
  - Tienen el mismo tamaño
- En el caso de dos archivos con el mismo nombre, pero diferente tamaño se indicará en un segundo log, y se copiará en otro directorio para ser chequeado posteriormente por el usuario (NO se detendrá la ejecución del script por este motivo).

En el caso del script que vaya a ser ejecutado de forma manual, debes respetar los siguientes requisitos:

- Solo un usuario concreto podrá ejecutarlo. (Se deberá indicar cual, justificar tu decisión e implementar este control)
- El directorio destino de la copia será configurable. (Se deberá indicar la forma de permitir esta configuración e implementarla).
- Antes de iniciar la copia debemos asegurar de que no habrá problemas. (Se deberán indicar las comprobaciones a realizar e implementarlas).

En el caso del script que se ejecutará de forma periódica

- La ejecución de este script se programará para todos los viernes a las 23:59.
- Antes de realizar ninguna copia se comprobará que el disco externo está montando en una ruta predeterminada. (Para esto se puede generar un fichero de marca y chequear su existencia)

## Ayudas técnicas y de diseño:

- El comando `locate` genera un listado de todos los ficheros que respetan un determinado patrón.
  - Por ejemplo `"locate jpg"`, genera un listado de todos los ficheros que contienen esta cadena.
- Una forma de comprobar que un dispositivo externo está presente en el sistema es buscar en él un "fichero marca" que habremos copiado allí anteriormente con tal propósito. Estos ficheros marca suelen tener nombres ocultos (que empiezan por `."`, por ejemplo `".marca_copia_pau"`) y están ubicados en el directorio raíz del dispositivo.
- La mejor forma de configurar un valor que será utilizado por diferentes scripts (o programas) es indicarlo en una variable global de una librería que será compartida por ambos scripts.
- `"basename"` es otro comando que te devuelve el nombre del fichero de una ruta.
  - `FILE=$(basename $RUTACOMPLETA)`
- `"dirname"` es otro comando que devuelve el nombre del directorio de una ruta.
  - `DIR=$(dirname $RUTACOMPLETA)`
- En este enlace explica otras formas de separar el nombre de un fichero de su ruta:
  - <http://diariolinux.com/2007/04/01/separar-nombre-y-extension-de-fichero-en-bash/>
- El comando `"stat -c %s file.txt"`, nos indicará el tamaño del fichero `file.txt`
  - `FSIZE=`

## Evaluación:

Recuerda que se valorará:

La utilización de **funciones** con parámetros

La utilización de **variables** locales y/o globales

La **claridad** del código del script principal

Generar **librerías** con las funciones

La **corrección** del código

Y la obtención del **resultado** esperado

## Ayuda:

Piensa!!!!