

ETSII: IC

SISTEMAS OPERATIVOS: MEMORIA PRÁCTICA 1

SANTIAGO RAMOS GÓMEZ

YAGO NAVARRETE MARTÍNEZ

ÍNDICE

AUTORES	2
DESCRIPCIÓN DEL SCRIPT	3
COMENTARIOS PERSONALES	4

AUTORES

Santiago Ramos Gómez:

Segundo año de Ingeniería de Computadores.

Yago Navarrete Martínez:

Segundo año de Diseño y Desarrollo de Videojuegos + Ingeniería de Computadores.

DESCRIPCIÓN DEL SCRIPT

Al comienzo del Script se crea la variable **"salida"**, igualada a **"\$?"**. Esto facilitará a lo largo del programa la gestión de errores.

El Script comienza al realizarse una comprobación de los argumentos de entrada, buscando que sean dos o más (directorios destino y origen). A su vez se comprueba que el primero de dichos argumentos sea necesariamente un directorio. En caso de que no se cumplan estas condiciones se notificará del **"ERROR (1)"**, se explica el motivo de su aparición y se cambia el valor de salida a **"1"**.

Tras comprobar los parámetros de entrada se entra en el primero de los bucles **"for"**, en que se comenzará a iterar sobre cada uno de los argumentos. El primero de dichos argumentos es introducido en una variable **"destino"** para, de nuevo, facilitar el seguimiento del código y que este sea más legible. Los parámetros pasados tras este serán los **"directorios origen"**. Se comprobará a continuación si se trata de directorios y en caso de que alguno no lo sea se lanzará un mensaje de error, notificando que dicho argumento no cumple los requisitos del programa. Si ese fuese el caso, la salida toma el valor **"2"** y se continuaría con los siguientes argumentos pasados.

Una vez dentro de cada directorio pasado por argumento, se realiza el siguiente bucle **"for"** en el que se itera sobre todos los elementos de dicho directorio, facilitados por el comando previamente ejecutado **"ls \$i"**, donde **"\$i"** corresponde al argumento que está siendo ejecutado. Una vez obtenida la lista de elementos del directorio, se itera sobre cada elemento, comprobando si se trata de un fichero. En caso negativo se continúa con el siguiente elemento.

Teniendo ya el nombre del fichero, se comprueba si existe un directorio con el mismo nombre en el directorio destino. En caso afirmativo, se imprime por pantalla un mensaje de error, no se realiza la copia y se continúa con el siguiente elemento. En caso contrario se ejecuta el comando **"cp -pu "FicheroACopiar" "DirectorioDestino"**". Gracias a **-p** (preserve) se mantienen los atributos del fichero original en el fichero copiado. Por otra parte, **-u** (update) realiza la copia únicamente si no existe otro archivo llamado de la misma manera, o si el fichero a copiar ha sido editado de manera más reciente que otro fichero con el mismo nombre en el directorio destino.

Una vez finalizadas toda las iteraciones, se llega a la salida de la ejecución del programa, expresada como **"exit \$salida"**. Así se finaliza el código con su respectivo código de error.

COMENTARIOS PERSONALES

Problemas encontrados:

Inicialmente nos resultó complicada la gestión de argumentos, puesto que desconocíamos la existencia de variables predefinidas como `$@` (que contiene todos los parámetros pasados por la línea de comando excepto el nombre del script).

Para tratar de comprobar si existía un archivo o directorio con el mismo nombre en direcciones destino y origen tratamos de obtener el nombre de cada elemento y eliminar su extensión. Esto resultó no ser una solución óptima, pues archivos y directorios no poseen extensiones propias como, por ejemplo, los ejecutables.

Crítica constructiva:

Esperábamos una práctica con un mayor nivel de dificultad y más contenido a realizar por el alumno. Añadir apartados o funcionalidades extra a implementar solventaría esto. No obstante, encontrar una práctica amena tampoco ha resultado un inconveniente.

Propuesta de mejoras:

Con el objetivo de mejorar la práctica de cara a otros años proponemos aumentar la variedad de comandos. Tener que investigar cómo funciona cada uno, que parámetros admiten y como cambia su funcionalidad en base a estos resulta muy formativo para el alumno. Si en la práctica se planteasen tareas que requiriesen usar un mayor número de estos sería, en conjunto, más didáctica.

Evaluación del tiempo empleado:

El tiempo total empleado en la práctica y su posterior memoria ronda las diez horas. La cantidad de tiempo escribiendo líneas de código es mucho menor que la empleada en buscar soluciones a diversos problemas logísticos que fueron surgiendo.