

NO SE PUEDE... en un... es, en el "trabajo"

Se valorará la sencillez, elegancia, y la programación de la actuación adecuada ante las distintas situaciones de error que puedan ocurrir.

### EJERCICIO 1 [5\*]

Construya un programa en C que genere un escenario en que se provoque que un proceso hijo es heredado por el proceso hijo (de pid 1), deberá mostrar la información en pantalla oportuna para poder concluir tal hecho.

### EJERCICIO 2 [5\*]

Genere un escenario en que se provoque que un proceso recibe la señal SIGPIPE. Deberá mostrar la información en pantalla oportuna para poder concluir tal hecho.

**EJERCICIO 3 [15\*]** Construya un programa en C que se ejecutará pasándole dos argumentos los nombres de dos directorios (dentro del directorio de trabajo) que contienen diverso número de archivos regulares. Procesaremos únicamente los archivos regulares que cuelguen directamente de cada uno de esos directorios. Queremos generar un directorio llamado ALBUM (también en el directorio de trabajo) que contendrá la unión de los archivos de los dos directorios anteriores. Pero queremos ordenar este conjunto unión según la fecha de última modificación; a cada archivo se le dará como nombre la fecha de última modificación (campo st\_mtime de la estructura stat) seguida del nombre del directorio donde está ubicado. Para copiar un archivo de **archivo-fuente** en el directorio **dir-destino** ejecutaremos la orden

**cp archivo-fuente dir-destino**

**EJERCICIO 4 [20\*]** Siendo **ruta** la ruta de un archivo regular, sabemos que la orden **du -h ruta** proporciona el número de bloques (de 1024 bytes) asignados a **ruta**. Por otra parte, la orden **ls -l ruta** proporciona el número de bytes que ocupa realmente el archivo **ruta**. Construir programa en C que tome de la entrada estándar un literal (que será una ruta), y utilizando las dos órdenes anteriores escriba en la salida estándar el número de bytes desaprovechados en esta asignación de espacio al archivo **ruta** (es decir, el primer número por 1024 menos el segundo número).

Entenderemos que existe el carácter "\n" que sirve de delimitador indicando que ha terminado una ruta y comienza la siguiente. Puede optar por la simplificación siguiente: suponer que cada ruta que recibe de la entrada estándar está formada por 10 caracteres exactamente.

Si la ruta recibida no es un archivo regular deberá proporcionar un mensaje de error y leer la siguiente ruta.

El programa producirá los mensajes de error en la salida de error estándar.

*M. Ame*