Introducción a los Sistemas Operativos

Trabajo para la evaluación Final (20% de la nota)

Parte I. Entrada/Salida y Multiusuario (12% de la nota)

La parte I se compone de dos utilidades a desarrollar, (1) y (2).

Para cada utilidad se deben realizar (en orden) las siguientes tareas.

Tarea 1. Describid el detalle de la **especificación** de la utilidad. Utilizad como ejemplo la especificación de la utilidad **`mypackzip`** (Actividad A04.6 de eGela).

Tarea 2. Describid el "**algoritmo**" de la funcionalidad que cumple la especificación.

Las tareas 1 y 2 se deben entregar en un mismo documento PDF.

Tarea 3. **Desarrollad el código C** de la utilidad **utilizando llamadas al sistema**. **Verificad** su correcto funcionamiento y cread el documento con el **test de verificación**. Utilizad como plantilla de una verificación la que aparece eGela en la Sección de *Documentación Variada*: *Documento para la verificación de programas*.

IMPORTANTE. Para la verificación de las utilidades cread dos directorios de pruebas (***`TabajoFinal/Util1`*** y ***`TabajoFinal/Util2`***). En ambos directorios añadid una serie de ficheros para realizar las pruebas y verificaciones de cada utilidad. Estos ficheros se utilizarán para las correspondientes verificaciones por el profesor. Recordad que se trata de que el test de verificación garantice el correcto funcionamiento de las dos utilidades en cada uno de los posibles supuestos. El contenido de los dos directorios (previo a las pruebas) debe añadirse a la entrega (en formato tar.gz).

Utilidad (1): Construid una utilidad que realice las siguientes tareas.
Dado un directorio como primer argumento (*`Dir1`*), un fichero de salida como segundo argumento (*`Resultado.dat`*) y dos números como tercer y cuarto argumento (*`Pos1`* y *`Pos2`*) construid una utilidad que realice la siguiente tarea:

Por cada fichero regular y que no sea de enlace simbólico que se encuentra en el directorio *`Dir1`*, añade al final del fichero *`Resultado.dat`* ciertos datos del fichero encontrado. Los datos que hay que añadir son los que se encuentran entre la posición *`Pos1`* y *`Pos2`* del fichero encontrado. Hay que tener en cuenta todas las diferentes posibilidades en cuanto a que el tamaño del fichero no sea superior a *`Pos1`* o *`Pos2`*.

Utilidad (2): Construid una utilidad que realice las siguientes tareas.
Dado un directorio (*`Dir1`*), cambiar los permisos de cada elemento que se encuentra en el directorio dependiendo de si es un directorio o un elemento de otro tipo:

a) A los elementos que sean directorios se les debe añadir los permisos de lectura y ejecución al propietario y al grupo del elemento.
b) Al resto de elementos se les debe eliminar los permisos de escritura y ejecución al grupo del elemento y al resto de los usuarios.

Los criterios de evaluación de esta actividad los podéis encontrar en eGela en la Sección de *Evaluación*: *Criterios de Evaluación: CEval 4, CEval 8 y CEval 9*. Además de estos criterios, se valorará la originalidad de las soluciones.

Parte II. Multiprogramación y Comunicación (8% de la nota)

La parte II se compone de dos utilidades a desarrollar, (3) y (4).

Para cada utilidad se deben realizar (en orden) las siguientes tareas.

Tarea 1. Describid el detalle de la **especificación** de la utilidad. Utilizad como ejemplo la especificación que aparece en el documento "Ejemplo de especificación" en eGela.

Tarea 2. Describid el "**algoritmo**" de la funcionalidad que cumple la especificación.

Las tareas 1 y 2 se deben entregar en un mismo documento PDF.

Tarea 3. **Desarrollad el código C** de la utilidad. **Verificad** su correcto funcionamiento.

Utilidad (3): En un centro de investigación se utilizan programas muy complejos, y se desea crear una nueva utilidad para detectar errores en sus resultados. Un programa que normalmente se ejecutaría de esta manera:

```
programa <fichero_entrada >fichero_salida
```

con la nueva utilidad lo ejecutaremos de la siguiente manera:

```
modo_seguro programa fichero_entrada fichero_salida
```

Cuando se utiliza esta utilidad, el programa especificado como argumento se ejecutará dos veces de manera concurrente, y si los resultados de ambas ejecuciones son iguales, entonces los resultados se escribirán en el fichero `fichero_salida`. Si, por el contrario, los resultados no son iguales, entonces no se creará el fichero de salida y se escribirá un mensaje de error en el canal de error estándar.

Al ejecutar el programa dos veces de manera concurrente, si transcurridos 10 segundos desde el final de la ejecución de uno de ellos el otro no hubiera finalizado, entonces la utilidad forzará su finalización y escribirá un mensaje de error en el canal de error estándar (sin crear el fichero de salida).

Construid la utilidad **`modo_seguro`**.

Utilidad (4): Construid la utilidad **`ISO_connected_ordered`**, que escribe en la salida estándar la lista de usuarios de la asignatura que se encuentran conectados, ordenados alfabéticamente por el nombre de usuario. Recordad que los usuarios de la asignatura son aquellos cuyo nombre de usuario está entre `acaf0000` y `acaf0099`.

Se valorarán la originalidad y la modularidad de las soluciones.