

Examen-SO-Modulo-2-Luis-Lopez-Es...



Soso0



Sistemas Operativos



2º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada



MÁSTER EN

Inteligencia Artificial & Data Management

MADRID

Formamos
talento para un futuro
Sostenible

saber más



Esto no son apuntes pero **tiene un 10 asegurado** (y lo vas a disfrutar igual).

Abre la Cuenta NoCuenta con el código **WUOLAH10**, haz tu primer pago y llévate 10 €.

Me interesa



1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos Holandés con una garantía de hasta 100.000 euros por depositante. Consulta más información en [ing.es](https://www.ing.es)

Examen Módulo 2

Profesor: Luis López Escudero (23-24, Grupo A3)

Ejercicio 1 – sumador

Crear el programa *sumador*:

Sintaxis: sumador <dir>

Parámetros:

- <dir> Directorio donde queremos ejecutar el programa sumador.

Funcionamiento:

- El programa leerá de la STDIN un número octal de 3 dígitos. Este número representa una máscara de **permisos**, los cuales incluyen los permisos de propietario, grupo y otros.
- Si el número octal leído es distinto de 0, el programa iterará por cada una de las entradas del directorio <dir> pasado como argumento.
 - Para cada una de las entradas, si se cumple que es un **archivo regular** y tiene **los mismos permisos**, el programa imprimirá por la STDERR información del archivo, con el siguiente formato:

Archivo: <nombre del archivo> | Bloques: <número de bloques> | Tamaño archivo: <tamaño en bytes>

- Tras iterar por todo el directorio, leerá un número de la STDIN. Si el número es 0, el programa mostrará por la STDOUT la suma total en bytes de todos los archivos que cumplieran la condición anterior, y finalizará.
- Adicionalmente, es muy recomendable comprobar los parámetros que se le pasa al programa, además de comprobar la ejecución de todas las llamadas al sistema (los if's)

Notas (no venían en el enunciado, pero son cosas que se deducen y hay que tener cuidado):

- Cuidado con la comparación de los permisos, cuando extraemos los atributos de un archivo en el struct de stat, el campo **st_mode** incluye tanto los permisos como el tipo de archivo, así que la comparación hay que hacerla sin el tipo de archivo:
 - <atributos archivo>.st_mode & !permiso : De esta forma le quitamos a los permisos el bit de tipo de archivo regular, quedando solo los permisos, y podremos comparar sin ningún tipo de problema.
- El profesor incidió mucho en el examen que el nº de bytes que leemos en la segunda lectura de la STDIN es 1, ya que solo leemos un 0 ("0"), mientras que en la primera lectura deben de leerse 3, un byte por cada dígito.

Consulta condiciones aquí



do your thing

WUOLAH

Ejercicio 2 – distribuidor

Crear el programa *distribuidor*:

Sintaxis: distribuidor <dir> <n_hijos> <octal 1> <octal 2> ... <octal n_hijos>

Parámetros:

- <dir> Directorio donde queremos ejecutar el programa.
- <n_hijos> Número de hijos que queremos crear.
- <octal 1> Número octal que se le pasará al primer hijo para ejecutar.
- <octal 2> Número octal que se le pasará al segundo hijo para ejecutar.
- ...
- <octal n_hijos> Número octal que se le pasará al último hijo para ejecutar.

Funcionamiento:

- El programa creará tantos hijos como se especifique en el argumento, y cada uno de ellos ejecutará el programa **sumador** creado en el primer ejercicio, usando <dir> como argumento. Además, el programa se encargará de pasarle los octales que correspondan a cada hijo (el nº 1 al hijo 1, y así sucesivamente). Para pasar cada número, el programa padre empleará un cauce sin nombre para comunicarse con cada hijo.
- El padre, para indicar que ha terminado de mandar el número octal, deberá de escribir **tantos 0 como hijos creados**.
- Los hijos creados consumirán los octales, los cuales recibirán por el cauce desde su padre, y una vez ocurra eso, deberán enviarle a su padre el tamaño de los subtotales, a través de otro cauce sin nombre.
- Una vez haya consumido todos los valores enviados por los hijos, el proceso padre mostrará por STDOUT la suma de todos los subtotales recibidos.
- Es necesario realizar los direccionamientos pertinentes para que el padre mande octales a hijos y cada hijo el subtotal a su padre.

Notas:

Para aclarar un poco el funcionamiento del código (el enunciado lo copié un poco resumido, puede que haya perdido ciertos matices):

- El padre crea **2 cauces**, uno para enviar a todos sus hijos el octal que le corresponde, y otro para recibir el subtotal de sus hijos.
- Cada hijo debe cerrar los fd que hereda de su padre para poder establecer bien el flujo de información (el envío de información en un cauce es unidireccional, cuidado). El padre debe de cerrar los fd que debe para establecer bien el paso de información.
- Una vez establecido el flujo de información, es importante que los hijos redireccionen sus STDIN y STDOUT a los cauces para recibir información y mandar información a su padre de forma correcta.
- El padre manda a cada hijo a través del cauce el numero octal, y además un 0 para indicarle que ha finalizado de mandar el número octal y que puede enseñar el subtotal.