SISTEMAS OPERATIVOS

Memoria – Práctica 1

28 DE FEBRERO DE 2020

Grupo 80

Iván Aguado Perulero – 100405871

Javier Cruz del Valle – 100383156

Jorge Serrano Pérez - 100405987

Índice

Descripción del código

1. *Mycat.c*

Para el correcto funcionamiento de este programa los pasos principales han sido los siguientes.

Antes de nada, se comprueba si el número de argumentos es válido, es decir, si hay al menos un fichero introducido, ya que si no lo hay habrá que informar al usuario.

A continuación, hemos abierto el fichero con permiso de lectura y ha sido almacenado en nuestra variable fd. A continuación, se comprueba que esta variable no contenga un -1, ya que esto indica que el fichero no se ha abierto correctamente, por lo que se avisa al usuario imprimiendo por pantalla un mensaje de error.

En caso de que no haya ningún problema, mediante un bucle, se va leyendo el fichero con la instrucción "read" y se va escribiendo, utilizando el buffer, el contenido en la salida estándar, utilizando para ello la constante STDOUT_FILENO dada en el enunciado. Este también tiene en cuenta posibles errores de lectura que interrumpirán el proceso de inmediato, haciéndoselo saber al usuario. Este proceso no acaba hasta que nuestra variable "bytes_leídos" sea 0, lo que indicará que no queda contenido en el fichero por ser leído.

Por último, cerramos el fichero y devolvemos "0", comprobando también que este se cierra correctamente e informando al usuario de lo contrario.

Hay que destacar que en todos los casos en los que se produce un error, el programa devolverá "-1" en vez de "0".

2. *Myls.c*

Comenzamos con el caso en el que el directorio no se pasa mediante un argumento. Por lo tanto, lo obtenemos mediante la llamada "*getcwd*" y lo almacenamos en un buffer. A continuación, abrimos ese directorio y lo almacenamos en nuestro puntero "dir", en caso de que este contenga el valor *NULL*, se mostrará por pantalla "error de apertura" y se devolverá -1. Si esto no ha ocurrido, leemos una a una las entradas del directorio mientras nuestra variable , en la que se almacenan, sea distinta de *NULL*, a su vez se muestra por pantalla cada una de ellas.

Por último, cerramos el directorio y devolvemos "0", comprobando como en el caso anterior que este se cierra correctamente.

Si el directorio es pasado por un argumento el proceso es el mismo, salvo que, a la hora de abrirle, se hace directamente con un "opendir" del argumento en cuestión. Al finalizar, lo mismo, se cierra el directorio, se comprueba su cierre y se devuelve "0".

Hay que recordar que en todos los casos en los que hay un error, el programa devolverá "-1" en vez de "0".

3. *Mysize.c*

En primer lugar, comprobamos que no haya un error en los argumentos pasados, informando al usuario de lo contrario, ya que solo podremos introducir un directorio. Si esto no ocurre, mediante getcwd almacenamos el directorio en un buffer.

Después lo abrimos y guardamos su descriptor en "dir" y antes de nada comprobamos que esta variable no sea *NULL*, en cuyo caso se avisa al usuario de un error de apertura y se devuelve -1.

Si no ha habido ningún error, se lee el directorio y se almacena en "dirent", mientras no sea 0 se sigue leyendo entrada a entrada. Cuando el campo "d_type" es igual a la constante DT_REG pasamos a abrir el fichero en modo lectura. A continuación, mediante "lseek" nos desplazamos hasta el final del fichero y obtendremos el valor del puntero. A su vez se mostrará por pantalla el nombre del fichero y el tamaño devuelto por "lseek", con la apariencia indicada. Acabamos cerrando el fichero. Se repetirá el proceso para cada una de las entradas.

En último lugar se cierra el directorio y se devuelve 0, comprobando otra vez su cierre.

Batería de pruebas

1. *Mycat.c*

PRUEBA	RESULTADO ESPERADO	RESULTADO OBTENIDO
Sin argumento de entrada	Sacar por pantalla "Error de argumentos"	Sacar por pantalla "Error de argumentos"
Fichero normal	Contenido del fichero	Contenido del fichero
Fichero con más de 1024 bytes	Contenido del fichero	Contenido del fichero

Fichero con líneas sin nada	Contenido del fichero	Contenido del fichero
Más de un fichero de entrada	Contenido del primer fichero	Contenido del primer fichero
Directorio de entrada	Sacar por pantalla "Error de lectura"	Sacar por pantalla "Error de lectura"
Fichero inexistente	Sacar por pantalla "Error de apertura"	Sacar por pantalla "Error de apertura"

2. *Myls.c*

PRUEBA	**RESULTADO ESPERADO**	**RESULTADO OBTENIDO**
Sin argumento de entrada	Sacar por pantalla los directorios y archivos del directorio actual en el que se ejecuta	Sacar por pantalla los directorios y archivos del directorio actual en el que se ejecuta
Directorio de entrada	Contenido del directorio	Contenido del directorio
Directorio inexistente	Sacar por pantalla "Error de apertura"	Sacar por pantalla "Error de apertura"
Más de un directorio	Contenido del primer	Contenido del primer

de entrada	directorio	directorio
Directorio vacío	Directorios '.' y '..'	Directorios '.' y '..'
Fichero de entrada	Sacar por pantalla "Error de apertura"	Sacar por pantalla "Error de apertura"

3. Mysize.c

PRUEBA	RESULTADO ESPERADO	RESULTADO OBTENIDO
Sin argumento de entrada	Sacar por pantalla los ficheros del directorio en el que se ejecuta	Sacar por pantalla los ficheros del directorio en el que se ejecuta
Directorio de entrada	Sacar por pantalla "Error de argumentos"	Sacar por pantalla "Error de argumentos"
Directorio inexistente	Sacar por pantalla "Error de argumentos"	Sacar por pantalla "Error de argumentos"
Fichero de entrada	Sacar por pantalla "Error de argumentos"	Sacar por pantalla "Error de argumentos"
Más de un directorio de entrada	Sacar por pantalla "Error de argumentos"	Sacar por pantalla "Error de argumentos"
Directorio sin ficheros	No sacar nada	No sacar nada

Conclusiones

El principal problema que hemos encontrado en esta práctica ha sido la familiarización con el entorno y el lenguaje de programación debido a que era totalmente nuevo para nosotros. Debido a esto se ha invertido muchas más horas al inicio de su realización, investigando y probando cosas. Una vez aprendido el manejo de estas nuevas herramientas se aceleró exponencialmente el proceso.

Para su correcta realización nuestra primera toma de contacto con la práctica fue de manera presencial, ahí nos hicimos un poco a la idea sobre cuál era nuestro trabajo e instalamos los programas necesarios. Posteriormente hemos trabajado por llamada online, compartiendo nuestro código y revisando el del compañero con el fin de pulir errores e ineficiencias.

En definitiva, una práctica "simple" para quien ya conoce sobre sus contenidos, sin embargo, puede ser muy costosa para novatos de este ámbito.