

PRÁCTICA 1:

Llamadas al sistema relacionadas con ficheros (I).

Objetivo:

Implementación de programas en C bajo UNIX, que utilicen las siguientes llamadas al sistema relacionadas con ficheros:

<i>creat</i>	<i>open</i>	<i>read</i>
<i>write</i>	<i>lseek</i>	<i>close</i>

Desarrollo:

- Se implementarán los programas según el enunciado correspondiente a cada grupo de prácticas (A, B, C, D, E, F, G, H, I, J).
- Se controlará el mayor número posible de errores:
 - Número correcto de parámetros en la línea de comandos (*argc*).
 - Parámetros correctos en la línea de comandos (*argv[1]*, *argv[2]*, ..., *argv[argc-1]*).
 - Errores al realizar las llamadas al sistema (retornan el valor -1), indicándose el tipo de error:
 - Error de creación de fichero.
 - Error de creación de fichero existente.
 - Error de apertura de fichero.
 - Error de escritura de fichero.
 - Error de posicionamiento de fichero.
 - Error de cierre de fichero.
 - Cualquier otro tipo de errores que el alumno considere oportuno controlar.
- En caso necesario, se tendrá en cuenta que siempre que se crea un proceso, se le asocian tres descriptores de fichero estándar:
 - 0 para la entrada estándar (teclado).
 - 1 para la salida estándar (pantalla).
 - 2 para la salida de error.
- Se utilizará la llamada al sistema *exit* (*estado*) para terminar la ejecución del programa con la condición de salida indicada por *estado*, donde 0 es la terminación normal y los demás valores son distintas condiciones de error.

Enunciados:**Grupo A**

Implementar un programa que mueva o renombre un fichero, de tal manera que en la línea de comandos se le pasen dos parámetros, correspondientes a dos ficheros con su ruta de acceso. El primero se refiere al fichero origen que se quiere mover o renombrar, y el segundo al fichero destino renombrado (o nueva ruta de acceso a la que se quiere mover el fichero origen). Después de la ejecución, sólo existirá el fichero destino.

Grupo B

Implementar un programa al que se le pasen tres parámetros en la línea de comandos. El primer parámetro es el nombre de un fichero origen, el segundo, un número de líneas (p. e. “-15”), y el tercero el nombre de un fichero destino.

El programa debe copiar las últimas n líneas del fichero origen (donde n se refiere al número de líneas pasadas en el segundo parámetro) en el fichero destino.

En caso de no introducir fichero destino en la línea de comandos, se copiarán las líneas en la salida estándar (pantalla).

En caso de no introducir número de líneas en la línea de comandos, el número de líneas por defecto será 10.

Grupo C

Implementar un programa que realice el copiado de un fichero, de tal manera que en la línea de comandos se le pasen dos parámetros, correspondientes a dos ficheros con su ruta de acceso. El primero se refiere al fichero origen del que se quiere realizar una copia, y el segundo al fichero destino o copia del primero.

En el caso de que el segundo parámetro sea “-d”, habrá un tercer parámetro que será la ruta donde vamos a almacenar el fichero destino. En este caso, el fichero destino tendrá el mismo nombre que el fichero origen pero estará ubicado en un directorio diferente, indicado por la ruta de acceso pasada como parámetro.

Grupo D

Implementar un programa al que se le pase como parámetro en la línea de comandos el nombre de un fichero de texto. El programa mostrará en pantalla la información contenida en el fichero, pero de forma inversa, de modo que el primer carácter que aparece es el último, a continuación el penúltimo, y así sucesivamente hasta que al final se visualiza el primer carácter del fichero.

Grupo E

Implementar un programa que visualice por pantalla la información contenida en el fichero de texto cuyo nombre es pasado como primer parámetro en la línea de comandos.

En el caso de pasar un segundo parámetro (“-m”), la visualización se realizará pantalla a pantalla, siendo necesario pulsar alguna tecla para pasar a la pantalla siguiente. Si la tecla pulsada es una concreta (p. e. la “q”), se terminará la ejecución sin visualizar el resto de la información contenida en el fichero, pero cerrándolo adecuadamente.

Grupo F

Implementar un programa al que se le pasen dos parámetros en la línea de comandos. El primer parámetro es el nombre de un fichero origen y el segundo, un número de líneas (p. e. “-15”). El programa debe mostrar por pantalla las primeras n líneas del fichero origen (donde n se refiere al número de líneas pasadas en el segundo parámetro). En caso de no introducir número de líneas en la línea de comandos, el número de líneas por defecto será 10.

Grupo G

Implementar un programa que, dado un fichero de texto cuyo nombre es pasado como primer parámetro en la línea de comandos, cuente el número de líneas y el número de caracteres en el mismo. El programa mostrará en pantalla un mensaje indicando cada uno de estos valores.

Grupo H

Implementar un programa que admita como parámetros el nombre de un fichero origen, el nombre de un fichero destino y un número, especificado en la forma “-num” (por ejemplo, “-15”). El programa debe copiar en el fichero destino todas aquellas líneas del fichero origen que tengan más de *num* caracteres.

Grupo I

Implementar un programa que admita como parámetros el nombre de un fichero de texto origen y el nombre de un fichero de texto destino. El programa debe copiar en el fichero destino las líneas pares del fichero origen. Por ejemplo, si ejecutamos **prog forigen fdestino:**

forigen:

fdestino:

Línea 0
Línea 1
Línea 2
Línea 3

Grupo J

Implementar un programa que admita como parámetros el nombre de un fichero de texto origen, a continuación dos parámetros numéricos *num1* y *num2* (especificados en la

forma "-15") y finalmente el nombre de un fichero de texto destino. El programa debe copiar en el fichero destino *num2* líneas del fichero de origen a partir de la línea *num1* (incluida ésta). Por ejemplo, si ejecutamos **prog forigen -2 -3 fdestino:**

forigen:

Línea 0
Línea 1
Línea 2
Línea 3
Línea 4
Línea 5
Línea 6
Línea 7

fdestino:

Línea 2
Línea 3
Línea 4

