

Memoria Practica 1 SSOO

Alberto Lozano 100495914

Pablo Moreno 100451061

Indice

- 1) Mywc
 - a) Descripción del código
 - b) Pruebas realizadas
- 2) Myls
 - a) Descripción del código
 - b) Pruebas realizadas
- 3) Myishere
 - a) Descripción del código
 - b) Pruebas realizadas
- 4) Conclusión

Mywc

- Descripción del código

El propósito del programa es leer el numero de bytes, lineas y palabras que tiene un fichero. Simula el comportamiento del comando de linux wc.

El algoritmo de funcionamiento de programa es el siguiente

1. Declaración de variables
 - a. bytesLeidos => Guarda el número de bytes leídos por la función read()
 - b. contadorBytes => Guarda el numero de bytes del fichero a leer
 - c. contadorLineas => Guarda el numero de lineas del fichero a leer
 - d. contadorPalabras => Guarda el numero de palabras del fichero a leer
 - e. tipoPrevio => Guarda el byte leído anterior al que se esta leyendo ahora
2. Comprobación de argumentos => Comprobamos que se han pasado 2 argumentos
3. Apertura de fichero objetivo con la función open()
 - a. Guardamos el descriptor del fichero en una variable fd
 - b. Si se retorna -1 lanzamos error
4. Bucle
 - a. Leemos el siguiente byte del fichero
 - i. Si retorna -1, lanzamos error
 - b. Si no se ha leído un byte, terminamos el bucle
 - c. Incrementamos el contador de bytes
 - d. Si se ha leído un carácter normal (no es espacio, tab, nueva linea)
 - i. Se mira el byte leído anterior y se actualizan los contadores correspondientemente
 - ii. Se guarda el byte leído actual en tipoPrevio
 - e. Si el tipo previo es un salto de lineas
 - i. Se incrementa el contador de lineas
 - ii. NO se actualiza el tipo previo
 - iii. Esto se hace para que no se cuenten espacios y tabs después de un salto de linea, pero si se cuenten saltos de linea sucesivos
 - f. En otros casos, se actualiza la variable tipoPrevio con el byte leído actual

5. Al finalizar el bucle, comprobamos si el ultimo carácter es un salto de linea
 - a. Esto se hace porque solo se actualiza el salto de linea al alcanzar un 'carácter normal', por lo que si un archivo acaba en una linea nueva, no se leerá en el bucle. Es necesario actualizar el contador mediante este condicional en ese caso
6. Imprimimos el resultado
7. Cerramos el directorio

- Pruebas realizadas

Prueba realizada	Resultado obtenido
Fichero vacío	0 0 0
Fichero una linea de texto y una palabra	Se lee 1 linea y una palabra y el numero de bytes correspondientes
Fichero con varias lineas. Una palabra por linea	El numero de lineas leídas es igual al número de palabras
Fichero con espacios consecutivos	Se lee el numero de palabras correcto, omitiendo espacios sucesivos
Fichero con todas las lineas vacías menos la ultima	Se leen las lineas vacías
Fichero con solo espacios en una linea	Se lee 1 linea, 0 palabras y el numero de bytes
Fichero con linea vacía al final	Se lee la linea vacía correctamente
Fichero vacío con solo saltos de linea	Se leen las lineas y 0 palabras
Se introduce como parámetro un fichero no existente	Error de open()

Myls

- Descripción del código

El propósito del programa es abrir un directorio y leer y imprimir el contenido de dicho directorio. Simula el comportamiento del comando de linux ls -f -1

El algoritmo de funcionamiento de programa es el siguiente

1. Declaración de variables
 - a. Path_actual => Guarda la ubicación del directorio cuyo contenido queremos obtener
 - b. lectura_actual => Estructura dirent que guarda el return de readdir()
2. Comprobación de argumentos
 - a. Si se pasa el path como argumento, se usa eso
 - b. Si no se pasa el path como argumento, obtenemos el path actual con getcwd()
3. Abrimos el directorio objetivo con opendir()
 - a. Lo guardamos en la variable dir
4. Bucle
 - a. Leemos el siguiente subdirectorio/fichero con readdir()
 - b. Si no se ha leído un subdirectorio/fichero, terminamos el bucle
 - c. Imprimimos el nombre del subdirectorio/fichero obtenido
5. Cerramos el directorio

- Pruebas realizadas

Prueba realizada	Resultado obtenido
Directorio vacío	Se imprime solo .. y .
Directorio con archivos/subdirectorios	Se imprimen los archivos/subdirectorios y .. y .
Intentar abrir directorio inexistente	Error de opendir()
No pasar parámetro de path	Se imprime el contenido del directorio actual
Pasar parámetro path de directorio existente	Se imprime el contenido del directorio dado

Myishere

- Descripción del código

El propósito del programa es comprobar si un fichero/subdirectorio se encuentra contenido en otro

El algoritmo de funcionamiento de programa es el siguiente

1. Declaración de variables
 - a. lectura_actual => Estructura dirent que guarda el return de readdir()
 - b. fichero_encontrado => Indica si se ha encontrado el fichero/subdirectorio objetivo
2. Comprobación de argumentos
 - a. Si hay menos de 3 argumentos lanzamos error
3. Abrimos el directorio objetivo con opendir()
 - a. El directorio a abrir sera el que indique primer parámetro
 - b. Si no se abre un directorio, lanzamos error
4. Bucle
 - a. Leemos el siguiente subdirectorio/fichero con readdir()
 - b. Si no se ha leído un subdirectorio/fichero, terminamos el bucle
 - c. Comparamos el subdirectorio/fichero leído con el que buscamos
 - i. Si encontramos en fichero, ponemos fichero_encontrado a verdadero
 - d. Si el fichero ha sido encontrado, acabamos el bucle
5. Imprimimos si hemos encontrado el directorio o no
6. Cerramos el directorio

- Pruebas realizadas

Prueba realizada	Resultado obtenido
Se busca un archivo arbitrario en un directorio vacío	No se encuentra el archivo
Se busca un archivo inexistente en un directorio con otros archivos	No se encuentra el archivo
Intentar abrir un directorio inexistente	Error de opendir()
Se busca un archivo existente en un directorio con otros archivos	Se encuentra el archivo
No se pasan suficientes parámetros	Error de "No hay suficientes parámetros"

Conclusión

La única complicación encontrada fue realizar la lógica de lectura del programa mywc debido al gran número de casos que puede encontrar este programa (saltos sucesivos, espacios sucesivos, ficheros vacíos, etc.). Por lo demás, la realización de la práctica fue relativamente simple.