Carlos Segundo Nieto
Héctor Ullate Catalán

EJERCICIOS DE LA PRÁCTICA 2

Ejercicio 1: Comparar el tiempo de ejecución de la copia de un fichero de gran tamaño (> 1MB) para diferentes valores de `BLOCKSIZ`. Explicar los valores obtenidos. Para ello ejecutar el programa con una orden como la siguiente:

```
time ./copy example1.pdf example1_1024.pdf 1024
```

```
usuarioso@ssoo:~/Escritorio/Práctica 2 SO/Examples/1_Copy$ time ./copy ./ficheroPDF ./copia 1024
Starting COPY
Ends of COPY

real    0m0,010s
user    0m0,000s
sys     0m0,004s
usuarioso@ssoo:~/Escritorio/Práctica 2 SO/Examples/1_Copy$ time ./copy ./ficheroPDF ./copia 512
Starting COPY
Ends of COPY

real    0m0,011s
user    0m0,000s
sys     0m0,005s
usuarioso@ssoo:~/Escritorio/Práctica 2 SO/Examples/1_Copy$ time ./copy ./ficheroPDF ./copia 128
Starting COPY
Ends of COPY

real    0m0,028s
user    0m0,000s
sys     0m0,011s
usuarioso@ssoo:~/Escritorio/Práctica 2 SO/Examples/1_Copy$ time ./copy ./ficheroPDF ./copia 1
Starting COPY
Ends of COPY

real    0m0,902s
user    0m0,092s
sys     0m0,794s
usuarioso@ssoo:~/Escritorio/Práctica 2 SO/Examples/1_Copy$
```

Podemos observar que a menor tamaño de bloque, mayor es el tiempo de ejecución del programa. Esto es debido a que si tenemos un tamaño de bloque pequeño vamos a tener que acceder al disco a por los datos más veces que en caso de tener un tamaño de bloque grande.

Ejercicio 2: En el ejemplo 2, si el fichero es un enlace simbólico, los atributos mostrados se refieren al *nodo-i* del fichero al que apunta el enlace. Incorporar una opción (`-L`) al programa anterior para que, cuando se especifique, haga que la consulta de atributos para un enlace simbólico se refiera al fichero apuntado, y cuando no se dé tal opción, se refiera a los atributos del propio enlace.

En resumen, para implementar la nueva opción necesitamos utilizar la llamada al sistema `lstat()`. En la carpeta de ejercicios se adjunta el archivo C con la resolución del ejercicio.

Ejercicio 3: Mostrar los resultados de aplicar el programa con ambas opciones al directorio `$HOME` del usuario.

```
usuarioso@ssoo:/home$ sudo ./usuarioso/Escritorio/P9_Pr2_Lab3/Examples/3_Distribution/distribution -t -n
Results for the directory ..
Range           Total   Percentage
[0K, 10K]       5328    63.090586%
[10K, 20K]      988     11.699230%
[20K, 30K]      359     4.251036%
[30K, 40K]      220     2.605092%
[40K, 50K]      200     2.368265%
[50K, 60K]      137     1.622262%
[60K, 70K]      108     1.278863%
[70K, 80K]      90      1.065719%
[80K, 90K]      54      0.639432%
[90K, 100K]     75      0.888099%
Greater         886     10.491415%
=======         =====   =======
TOTAL FILES     8445    100.000000%

Range           Total   Percentage
[ 0,  10]       1861    93.658782%
[10,  20]       65      3.271263%
[20,  30]       17      0.855561%
[30,  40]       18      0.905888%
[40,  50]       7       0.352290%
[50,  60]       0       0.000000%
[60,  70]       3       0.150981%
[70,  80]       3       0.150981%
[80,  90]       0       0.000000%
[90, 100]       2       0.100654%
More than 100   11      0.553598%
=======         =====   =======
TOTAL DIRECT.   1987    100.000000%

usuarioso@ssoo:/home$
```

Ejercicio 4: ¿Dónde están y cómo podemos obtener los datos correspondientes al fichero 2 usando `hexdump`? consulta el manual de `hexdump` para ello.

En nuestro sistema de ficheros asignamos los bloques de datos por orden y a partir del octavo, por tanto el fichero 2 comenzará en el bloque de datos siguiente al último bloque de datos del fichero 1.

Para leer correctamente el contenido del fichero 2 utilizaremos el argumento `-s` para posicionarnos al comienzo del bloque del fichero 2, y el argumento `-n` para indicar los bytes a leer. Como el inicio del fichero 1 esta a 32768 bytes del inicio del fichero, sumamos el tamaño de un bloque (4096 bytes) para obtener el inicio del fichero 2.

El comando que nos mostraría los datos correspondientes al fichero 2 es el siguiente:

```
hexdump virtual-disk -C -s 36864 -n 15
```