# Ejercicios de Ficheros

1.

- Si se utilizan 6 bytes para almacenar números de bloque entonces en cada bloque de datos se pueden almacenar: (8*1024) / 6 números de bloque. Es decir, 1365 números de bloque.
- **Veamos ahora cuantos bloques puede tener como máximo un fichero**:
  7 bloques → Por las 7 entradas directas
  +
  1365*1365 bloques → Por la entrada indirecta doble
  +
  2 * 1365*1365*1365 bloques → Por las dos entradas indirectas triple

  **Lo que nos da un total de 7+1.863.225+5.086.604.250 = 5.088.467.482 bloques**

- **Como cada bloque son 8KiB, el tamaño máximo en bytes de un fichero es:**

**5.088.467.482 * 8 * 1024 = 41.684.725.612.544 bytes**

2.

Vamos a asumir que hay que contar lo que hace a partir de que ya haya sido abierto el fichero (es decir solo lo que hace para la operación de lectura):

- Lo primero es saber a qué bloque quiere acceder, es decir, a qué bloque corresponde el desplazamiento 10.058.

  En este método todo el bloque se usa para datos (3KiB), por tanto, para saber el ordinal del bloque al que quiere acceder sería realizar la siguiente división: 10.058/(3*1024) = 3  10.058 MOD (3*1024) = 842

  La parte entera indica el ordinal del bloque al que hay que acceder y el resto indicaría el desplazamiento, dentro de ese bloque, al que habría que acceder.

  Es decir, tiene que acceder al CUARTO bloque (el primero tiene posición 0), y dentro de ese cuarto bloque al desplazamiento 842.

- Veamos ahora cuáles son los cuatro primeros bloques del fichero:
  101 102 103 104
  Por tanto, hay que leer el bloque número 104 y dentro de él acceder al desplazamiento 842.

- **Finalmente, indicamos los pasos que haría el sistema operativo:**
  1. **Accede al BDF del fichero, al campo correspondiente al primer bloque del fichero, y obtendría que el primer bloque es el 101**.
  2. **Accedería a la entrada 101 de la tabla en MP y obtendría el valor 102** (segundo bloque).
  3. **Accedería a la entrada 102 de la tabla en MP y obtendría el valor 103** (tercer bloque)
  4. **Accedería a la entrada 103 de la tabla en MP y obtendría el valor 104** (cuarto bloque)
  5. **El SO ordena leer el bloque número 104 del disco**.
  6. **Finalmente, el SO accedería al desplazamiento 842 de dicho bloque para leer la información solicitada**.

3.

Vamos a asumir que hay que contar lo que hace a partir de que ya haya sido abierto el fichero (es decir solo lo que hace para la operación de lectura):

- Lo primero es saber a qué bloque quiere acceder, es decir, a qué bloque corresponde el desplazamiento 8.733.

  En este método parte del bloque se usa para almacenar el siguiente bloque (6 bytes), por tanto, se usan para datos: (2,5*1024)-6 = 2554 bytes. Para saber el ordinal del bloque al que quiere acceder sería realizar la siguiente división: 8.733/2554 = 3 8.733 MOD 2554 = 1.071

  La parte entera indica el ordinal del bloque al que hay que acceder y el resto indicaría el desplazamiento, dentro de ese bloque, al que habría que acceder.

  Es decir, tiene que acceder al CUARTO bloque (el primero tiene posición 0), y dentro de ese cuarto bloque al desplazamiento 1.071.

- Veamos ahora cuáles son los cuatro primeros bloques del fichero:
  101 125 102 103
  Por tanto, hay que leer el bloque número 103 y dentro de él acceder al desplazamiento 1.071.

- **Finalmente, indicamos los pasos que haría el sistema operativo:**
  1. **Accede al BDF del fichero, al campo correspondiente al primer bloque del fichero, y obtendría que el primer bloque es el 101**.
  2. **El SO ordena leer el bloque número 101 del disco**
  3. **El SO accede al campo correspondiente del bloque leído en el paso anterior y obtiene el número de bloque siguiente: el 125** (Segundo bloque)
  4. **El SO ordena leer el bloque número 125 del disco**
  5. **El SO accede al campo correspondiente del bloque leído en el paso anterior y obtiene el número de bloque siguiente: el 102** (Tercer bloque)
  6. **El SO ordena leer el bloque número 102 del disco**
  7. **El SO accede al campo correspondiente del bloque leído en el paso anterior y obtiene el número de bloque siguiente: el 103** (Cuarto bloque)
  8. **El SO ordena leer el bloque número 103 del disco**
  9. **Finalmente, el SO accedería al desplazamiento 1.071 de dicho bloque para leer la información solicitada**.

4.

**Los pasos que haría el sistema operativo serían:**

1. **Obtener el BDF del directorio raíz desde la memoria principal.**
2. **Cargar temporalmente, de uno en uno, los bloques de datos del directorio raíz y buscar en estos bloques de datos (que son pares: Nombre, Nº inodo) el nombre *home***
   Si no lo encuentra error de ruta incorrecta (*home* no existe) y sale de la operación
3. **Carga temporalmente el BDF (nº inodo) del directorio *home* a MP (si no estuviera ya)**
4. **Cargar temporalmente, de uno en uno, los bloques de datos del directorio *home* y buscar en estos bloques de datos (que son pares: Nombre, Nº inodo) el nombre *usuario1***
   Si no lo encuentra error de ruta incorrecta (*usuario1* no existe) y sale de la operación
5. **Carga temporalmente el BDF (nº inodo) del directorio *usuario1* a MP (si no estuviera ya)**
6. **Cargar temporalmente, de uno en uno, los bloques de datos del directorio *usuario1* y buscar en estos bloques de datos (que son pares: Nombre, Nº inodo) el nombre *practica1.cc***
   Si no lo encuentra error de ruta incorrecta (*practica1.cc* no existe) y sale de la operación
7. **Crea una entrada (si no estuviera ya) en la tabla de BDFs en MP y carga en ella el BDF (nº inodo) del fichero *practica1.cc* y poner el contador de referencias a 1. Si la entrada ya estuviera sólo se añadiría 1 al contador de referencias.**
8. **Crea una entrada en la tabla de ficheros abiertos del sistema en MP, con los campos: Puntero de lectura (a 0), Nº Compart: -1, y un puntero a la entrada correspondiente de la tabla del punto anterior**
9. **Crea una entrada en la tabla de descriptores del proceso (tabla de ficheros abierto del proceso) y dicha entrada apuntaría a la entrada creada en el punto anterior**
10. **Retorna el número de entrada (descriptor del fichero) del punto anterior al proceso como valor de retorno de la operación *open*.**