



Steps on search for file ICEBERGS.TEX in disk3.IMA (image could be found at: webhome.csc.uvic.ca/~wkui/Courses/CSC36O/disk3.IMA)

Step 1: Find the location of the root directory: offset 0x0002600 (Note that the root directory starts from the 19-th sectors and each sector has 512 bytes)

Step 2: Search the root directory and find an entry with the name matching ICEBERGS and its extension matching TEX. Note that each entry in the directory has 32 bytes. The location of this entry is at: 0x00026a0.

Step 3: Find the first Logical cluster of this file (note that 1 sector per cluster): at the locations: $0x00026a0 + (0x1a) = 0x00026ba$. The value is: 0x 65 00 (in little endian); this value is $6 \times 16 + 5 = 101$.

Step 4: Calculate the file size (the last four bytes in the entry): The value is 0x 5043 0000 (in little endian), which is $0x43 \times 256 + 0x5 \times 16 = 17232$ bytes. So we need 34 sectors to store the file.

Step 5: The first 512 bytes (1 sector) will be stored in the physical sector 132 (since $33 + 101 - 2 = 132$). Sector 132 starts at the location 0x 0010800. Read in 512 bytes from that place.

Step 6: We need to search the next sector where the file is stored. To do so, we need to search the 101-th entry in the FAT table. Since each entry in the FAT table has 12 bits and the FAT table starts from 0x0000200, we can use the calculation introduced in "FAT packing" (refer to An overview of FAT12) to find the value in the 101-th entry. It is the eight bits at the offset $(0x0000200 + 1 + 3 \times 101/2 = 0x0000298)$, which is 0x 06, combined with the high four bits at the offset $(0x0000200 + 3 \times 101/2 = 0x0000297)$, which is 0x 6. Put them together, we get the next logical sector storing the file, which is: $0x066 = 6 \times 16 + 6 = 102$. The corresponding physical Sector is: $33 + 102 - 2 = 133$. Sector 133 starts at location 0x 0010a00; read in another 512 bytes from that place.

Step 7: Repeat Step 6 to search for the next sector where the file is store, until the entry in the FAT table has a value 0xFFF (the end of the file).

Note that since the file length (17232) is not divisible by 512, the Last 512 bytes are padded with 0s. For the last sector, you only need to read in: 336 bytes ($= 17232 - 512 \times 33$).