

Q&A: Assignment 3

1. How to read a subdirectory?

Ans: In the root directory, first, check if an entry in the directory is a file or a subdirectory. If the entry is a subdirectory, find the logical cluster number of this subdirectory (denoted the first logical cluster number as x), and then find the corresponding physical cluster number (which is $x+31$). Read in 512 bytes using this physical cluster number. Every 32 bytes in this 512 bytes represent a directory entry, which points to either a file or another layer of subdirectory. Repeat the above procedure recursively until no subdirectory can be found.

Of course, you also need to check x -th entry in the FAT table to find out if the subdirectory uses more than 512 bytes. If the x -th entry in the FAT table is not 0xFF8-0xFFF, you need to read the next 512 bytes. This is exactly the same steps used when you read the content of a regular file.

2. How is the disk image created? Can I create my own disk image for testing?

Ans: We used software WinImage to create disk image. You can download the software (I use the 9.0 version) from this link: <https://www.winimage.com/download.htm>

Using winImage, you can generate and customize any types of new FAT-12 images to test your program (e.g., insert new files or create new folders within the image). A short tutorial is attached.

3. Where can I find the label of the disk information?

Ans: According to the specification of FAT-12, this information should be stored both in boot sector and the root directory. However, WinImage does not strictly follow this specification for the aspect, and you may find it's a blank string within the boot sector on this field in the disk image created with WinImage. Therefore, you need to get the disk label information from root directory instead of the boot sector. In the root directory, find the directory entry with attribute 0x08, e.g., CSC360 in disk.ima

4. Where can I find OS Name?

Ans: OS Name: boot section (starting byte 3, length 8 bytes, e.g., MSDOS5.0 in disk.ima)

5. How can I debug my code?

Ans: There are many ways. One simple way is to follow FATsearchsteps.pdf, print out intermediate steps, and check if your intermediate outputs match the results shown in FATsearchsteps.pdf.

6. How should we handle the case where we want to insert a file in the root directory, but there are no free entries there?

Ans: Ideally, your code should handle this situation: when there is no free entry for the file, you should print out the problem and exit. Nevertheless, we will not test this boundary case, so you will be safe if your program does not handle this special case.

7. What should I do if I put a file whose name has existed in the disk?

Ans: Output an error message that “There is a file of the same name in the disk” and return.

8. Should directory names given in disk_put be case-sensitive? That is, should we treat a.c and A.C differently?

Ans: FAT-12 filename is not case-sensitive. So a.c and A.C should be treated as the same file.

9. Should we care about the upper and lower case of file name? For example, I put a file a.c into the disk, the disk will get a file named a.c or A.C?

Ans: unlike linux file system, FAT-12 is not case-sensitive. For consistency, FAT-12 directory entries store the file name in upper case.

10. Since FAT-12 filename is case-insensitive but Linux filename is case-sensitive, when we use diskget to read a file (e.g., ANS.PDF) from the disk image to Linux, what file name should we use in Linux?

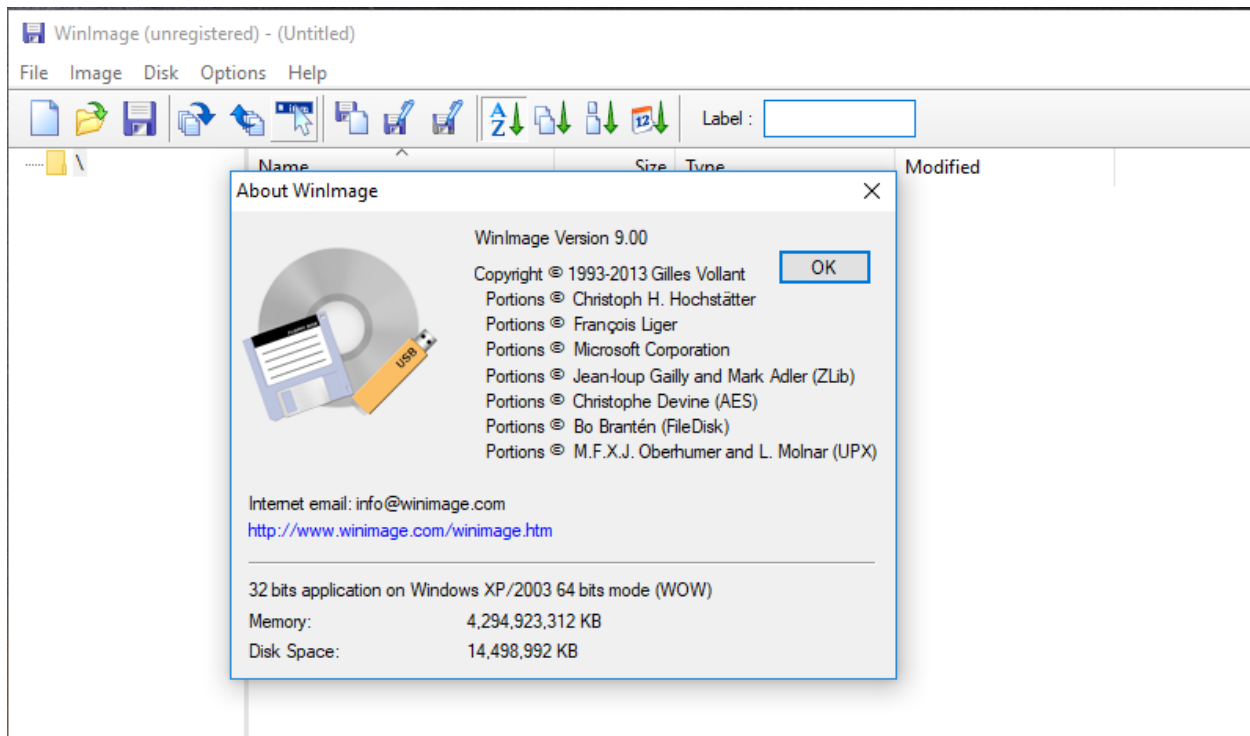
Ans: The filename in Linux can use either upper or lower case. Either way is ok.

11. Do I need to handle long file names?

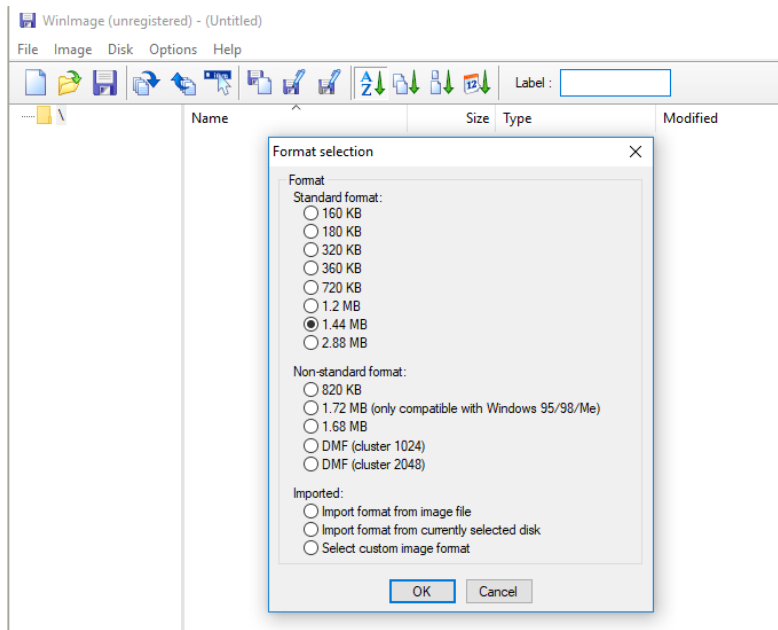
Ans: No. If the Attributes byte of a directory entry is 0x0F, then this directory entry is part of a long file name and can be ignored for purposes of this assignment.

WinImage download link: <https://www.winimage.com/download.htm>

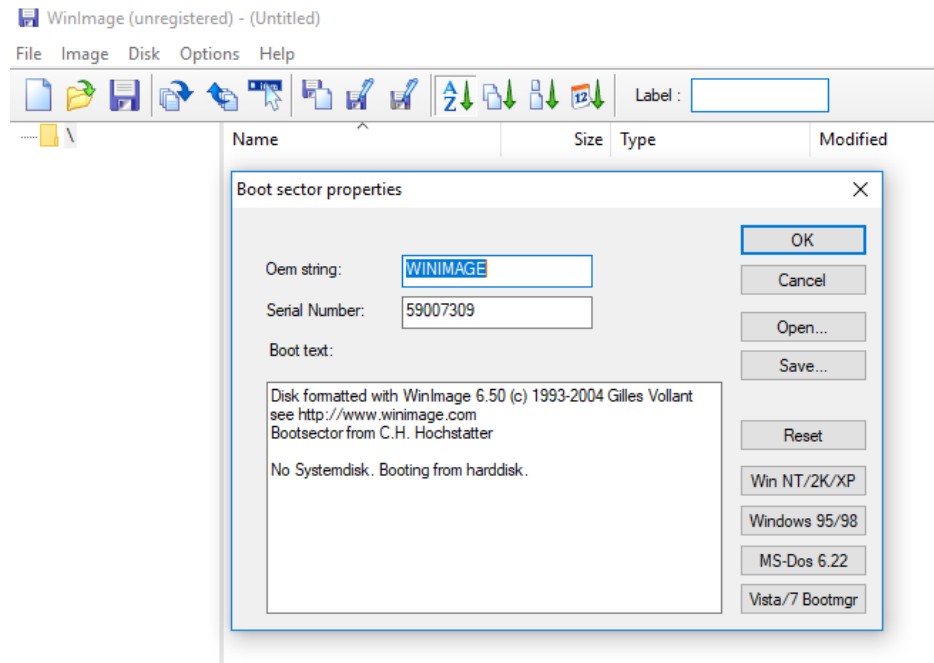
The version I use:



To create a new image, select File->New, and select 1.44MB standard format, then click OK.



Then you can inject new files or create new folders within the image. You can use **Image->Inject** to insert new files to the image, and use **Image->create folder** to create a new directory in the image. You can also customize the **Label of disk** and **OS name** fields, etc.



Once you finish, you can save the image to your local computer:

