



UNIVERSITEIT VAN AMSTERDAM

OPERATING SYSTEMS

Assignment 4

Authors:

Abe WIERSMA

Bram van den AKKER

May 26, 2013

Contents

1	Introduction	2
2	Overview of the FAT12 file system	2
3	FAT12 file extraction	2
4	Data corruption in FAT12	2
4.1	Inconsistency in the two file allocation tables	2
4.2	Empty assigned clusters	2
4.3	More clusters used then assigned	2
4.4	Double cluster references	3
4.5	Looping references	3
4.6	The root and parent folders	3
4.7	File names	3
4.8	Folder sizes	3
4.9	Deleted data	3
5	Solutions to corruptions	3
5.1	Inconsistency in the two file allocation tables	3
5.2	Empty assigned clusters	4
5.3	More clusters used then assigned	4
5.4	Double cluster references	4
5.5	Looping references	4
5.6	The root and parent folders	4
5.7	Filenames	4
5.8	Folder sizes	4
5.9	Deleted data	5
6	Results	5
6.1	Image 1	5
6.2	Image 2	5
6.3	Image 3	5
6.4	Image 4	5
6.5	Image 5	5
6.6	Image 6	5
6.7	Image 7	5
6.8	Image 8	6
7	Conclusion	6

1 Introduction

This report is for the course Operating Systems(UvA), we will discuss FAT12 file systems extracting FAT12 partitions and finding and solving data corruptions. For the assignment we got a skeleton code, to which we added extraction functionalities and improved data corruption detection within the code. Solving the data corruptions was not part of the assignment, we will only be discussing theoretical solutions.

2 Overview of the FAT12 file system

FAT12 was introduced in august 1980, and was designed to be a floppy disk file system, but was later used as a hard disk file system. The newest version of the FAT file system, FAT32 is widely used on USB drives.

3 FAT12 file extraction

The extraction begins with going through the FAT12 table and buffering the found files, then the buffers are written to correct folders. When a new folder is found in this process it is created. Corrupted files will not be handled and a message is returned with the type of corruption and the name of the corrupted file or the locations of the clusters.

4 Data corruption in FAT12

4.1 Inconsistency in the two file allocation tables

Every FAT file system has two file allocation tables that should be identical, when this is not the case one of the two is inconsistent and there is a high risk of corrupt data.

4.2 Empty assigned clusters

When there are more clusters assigned to a file then needed, whilst retrieving a file, this indicates missing file data.

4.3 More clusters used then assigned

When the length in the DIRENTRY is exceeded but there are still more references to other clusters, this is an indication of corrupted administration.

4.4 Double cluster references

When there are two DIRENTRIES referring to one set of clusters this indicates a administration corruption. Multiple references to a set of clusters is not supported.

4.5 Looping references

When a cluster references to a previously read cluster, this would create a infinite loop, to avoid getting stuck in the loop, the program stops when the the number of clusters to go through is reached.

4.6 The root and parent folders

Every folder in a fat file system has a . and a .. folder reference. The . goes to the root of the file system and de .. goes to the parent of the current folder. These two references should not be present in the root folder.

4.7 File names

File names should not contain unsupported or illegal characters and should have a positive length.

4.8 Folder sizes

Folders have a size of zero, different sizes indicate corruption.

4.9 Deleted data

Files or folders could be marked as deleted even when they should not. This could lead to unwanted data loss.

5 Solutions to corruptions

5.1 Inconsistency in the two file allocation tables

Search for corruptions in both the file allocation tables and keep the one that has none.

5.2 Empty assigned clusters

The clusters must be recovered with the help of data recovery tools, chances of full data recovery are very small.

5.3 More clusters used then assigned

Ignore the length in the DIRENTRY and try to find the end of the clusters.

5.4 Double cluster references

In this case there are two possibilities: One file it's direntry is overridden by another and therefor cannot be located on the disk. The other one is that there are just two direntries for one file. The first case cannot be solved unless there is information available about the file and the clusters are manually searched. The second case not solvable. The file exists in two places. The user should decide which one to keep, no data is lost.

5.5 Looping references

Build in safeguards to prevent systems crashes caused by infinite loops when reading or extracting the files. The only possible solution is manually searching for the remaining clusters.

5.6 The root and parent folders

When . or .. folders are found in the root folders these should be deleted. When they are not found in sub folders they should be added. Parent folders can be found by searching through the rest of the system.

5.7 Filenames

Corrupted file names should be fixed by using placeholders or querying the user for a new file name.

5.8 Folder sizes

When a folder is found with another size of then zero the size should be simply changed to zero and any linked clusters should be deleted.

5.9 Deleted data

Files and folders that are marked as deleted could be revised by the user to check if they should really be marked as deleted.

6 Results

6.1 Image 1

Image 1 has two inconsistent file tables. The files folder is also marked as deleted which might be a direct result of this inconsistency.

6.2 Image 2

Image 2 has 8 inconsistent file table entries. The file shirt-bl.png has an loop and the file stick.gif is bigger then the amount of available clusters.

6.3 Image 3

Image 3 is the correct image.

6.4 Image 4

In image 4 the file shirt-bl.png is extracted twice and there are 3 broken files. The files are bigger than the amount of clusters assigned.

6.5 Image 5

The File allocation table of image 5 has one inconsistent entry. Spinnato.pdf is also inconsistent, there is a bad reference at the end.

6.6 Image 6

The File allocation table of image 6 has two inconsistent entries. There is one broken file and one broken folder due to bad references.

6.7 Image 7

There are two broken files in image 7: One because the file is bigger then the assigned clusters and one because of a bad cluster reference. Readme.txt is extracted twice.

6.8 Image 8

Image 8 contains a lot of deleted files. This could be counted as an inconsistency but cannot be directly marked as inconsistent. Readme.txt is also extracted twice in image 8.

7 Conclusion

Fat12 file systems are far from perfect. But they give great perspective in how file systems work. The corruptions found can be caused by a lot of things and could occur on other file systems as well. Solving these corruptions can be a pain in the ass, preventing data loss by regularly making backups is a lot easier then recovering lost data.