Flash Storage Analysis

Taylor Futral, Sabrina Tsui, and Leah Langford
UCSC Storage Systems Research Center
tfutral@ucsc.edu, sctsu@ucsc.edu, llangfor@ucsc.edu

September 20, 2017

1 Topic

Different file systems have different access patterns and ways of dealing with flash and disk memory. The disk drive and flash drive is a type of storage hardware that is used to store data as explained in the background. The goal of this project is to use different types of forensics software to observe where each file system writes to the free list on a disk and flash device.

The aim of this project is to analyze where (in terms of sectors and offsets) data is written on the free list, and where on disk or flash device that is. By looking at the physical locations in memory things are being written to, we hope to find places in the free list where files can potentially be hidden without being overwritten. The systems that this analysis will focus on are the New Implementation of a Log-structured File System (NILFS) for the disk drive, the YAFFS2 (Yet Another Flash File System), and potentially The Journaling Flash File System (or JFFS) for the flash device. Each of these should reveal different patterns of accesses and hopefully will allow us to interpret which file system would be the best for hiding information in the free list.

To make this analysis for YAFFS2 we will be using Sleuth Kit which is a library that exists within Unix and Windows systems that allow for forensic analysis of computer systems. Assuming Sleuth Kit works with our hardware and virtual machines, it should be the perfect tool for the job. We plan to use VirtualBox as our virtual machine to host the operating systems containing the file systems of interest. This means that the parameters and settings for one set-up should match the set ups for the other systems too. For JFFS and NILFS we are still figuring out the best way to collect data.

Our intention with this project is not only to learn where and how each operating system is writing to in the free list but also to learn more about the disk and file systems in general. This learning opportunity should be significantly helpful to improving our knowledge and command as computer science students.

2 Background

Storage Disks are mechanisms dedicated to storing data via electrical, optimal, magnetic, and mechanical manipulations to the disk itself. It is rotated in order to store said data in different segments of the disk. For the case of digital disk drives in the use of Log file systems, the disk is partitioned into multiple "blocks" or clusters (as seen in Figure 1).

Flash memory is a non-volatile, which means that it will retain data whether the flash device is on or off. This is an advantage because most memory devices need power to retain data. Another advantage is that flash memory works without any mechanical parts, data is stored in individual cells in the flash chip and protected by the floating gate. Content in a flash memory device cannot be overwritten instead it has to be erased in order for data to be stored. In order for data to be erased a flash is sent to that cell and the data is cleared. This is one of the downfalls of flash memory, because if a device keeps on getting data erased it corrupts the memory device leading to the device eventually failing. Image break down of bad block management of NAND Flash Memory (Figure 2).

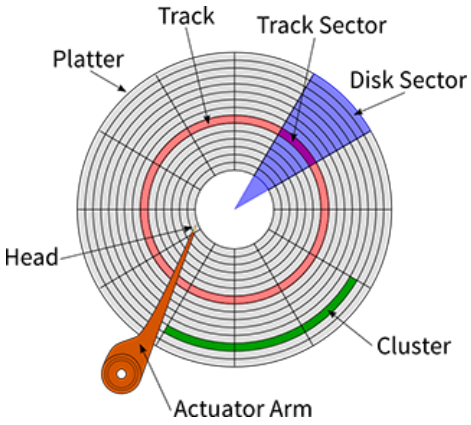

Figure 1: Image of disk partitioned into blocks

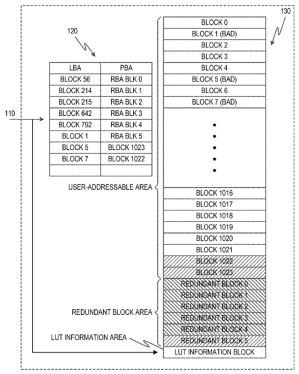

FIG. 2

Figure 2: Image of Flash NAND Memory break down

A free list is a data structure that allows for dynamic memory allocation. Typically this is created and maintained by having each node in a free list point to the next however sometimes the term “free list” is used inappropriately and refers to free memory in general. The significance of the free list for this project is that the different file system’s patterns of accessing it will be observed.

Common types of flash file systems:

- JFFS/2
- YAFFS/2
- ZFS
- UBIFS
- LogFS
- F2FS
- FAT

Sleuth Kit is an open source forensic analysis command line tool. Sleuth Kit can be used to determine operations and accesses within a file system. This functionality can be used to observe where in the free list and on the disk that memory is being written to.

3 Procedure

3.1 Part 1

1. Write test files to memory and find where the data is written to on the disk and free list on these operating systems:

    - Linux/Windows → YAFFS2
    - Linux → NILFS, JFFS

2. Find patterns of where things are written (understand where the computer "decides" to write things).

3. Find spots in the free list where they are less likely to be overwritten.

3.2 Part 2

1. Compare where the data is written with the aforementioned operating systems

    - Are there spots in the free list that are common to both file systems?

4 Schedule

Winter Quarter

Week 1: Pivot Project
Week 2: Write new Proposal
Week 3: read chapter 1.3.2, 1.5.2, and 1.5.3 Modern Operating Systems
Week 4: read chapter 3.2 of Modern Operating Systems
Week 5: read 3.5.9, 4.3.5, and 4.3.6
Week 6: Read about Sleuth kit and test software for YAFFS2
Week 7: Set up environment for YAFFS2, and design test cases. A Flash Memory Based File System
Week 8: Run Sleuth kit to determine where the OS is writing for the YAFFS2 file system, record results
Week 9: continue tests on the YAFFS2 file system, Read 5.1.4 MOS
Week 10: Research which forensic software to use for NILFS

Spring Quarter

Week 1: Read about and test forensic software for NILFS.
Week 2: Set up environment for NILFS, and design test cases.
Week 3: Run forensics software to determine where the OS is writing for the NILFS file system, record results, read 13.1.3
Week 4: continue tests on the NILFS file system
Week 5: Synthesize and Analyze data from both tests
Week 6: Draw conclusions based on analysis that would be useful for matryoshka

Presentation Prep

Week 7: Compile procedure/data/results for a presentation
Week 8: write presentation
Week 9: Edit, revise and practice presentation
Week 10: Give presentation

References

[Bie16] Tim Bielawa. The Linux Sysadmins Guide to Virtual Disks. 2016.

[Kaw95] Atsuo Kawaguchi. A Flash-Memory Based File System. 1995.

[Mic11] Micron. Bad Block Management in NAND Flash Memory. 2011.

[Ros92] Mendel Rosenblum. The Design and Implementation of Log-Structured File System. *ACM Transactions on Computer Systems (TOCS)*, 1992.

[Sel93] Margo Seltzer. An Implementation of a Log-Structured File System for Unix. 1993.