WRITING TOPIC THREE

FILE SYSTEMS

November 28, 2016

Steven Silvers
Oregon State University
CS 444 Operating Systems II

I. Introduction

A file system is the way that a particular operating system uses methods and data structures to manage and keep track of files on the disk[1]. There are numerous file systems available for use, from Windows' New Technology File System(NTFS) to the popular ext3 file system that was built on ext2, keeping most of the file system the same but added journaling.

Journaling is a file system feature that is most helpful when the computer crashes or has a power failure. File systems that make use of journaling track desired but not yet committed changes to the file system in a data structure, typically a circular log[2]. This data structure of uncommitted changes is referred to as the journal, and when the system crash or power failure occurs, the data stored in the journal is used to help bring the system back online quicker than if it did not have a journal, and with a lower likelihood that the data will be corrupt. It is for this reason and almost this reason alone that ext3 has become the default file system in Linux, because it is a journaled file system on top of the already popular and feature packed ext2[1]. There are a few options when using a journaled file system, for example you could opt to use a journaled file system that only tracks metadata which would improve the overall performance of the file system but would be more at risk to memory corruption during a crash than using a journal that tracks stored data as well as metadata.

II. File Systems in Linux

The Linux operating system supports a wide variety of file systems that all have pros and cons based on what you need out of your system. As Linux is based on UNIX, everything in Linux is considered a file unless it is a process. This means that apart from processes, everything in Linux must be managed by a file system in one way or another. FreeBSD is also based on UNIX meaning there will be many similarities in regards to file system between Linux and FreeBSD.

In a traditional file system, such as ext3, the file system is not aware of the structure of the disks mounted to the kernel. This means that for every disk you had mounted, you would also need one file system mounted to manage that disk. There is however one file system that can manage multiple disks all by itself, and that is the Zettabyte File System or ZFS developed by Sun Microsystems as part of Solaris, but is now owned by Oracle.

ZFS is unique in that it combines the volume manager with the file system, which allows ZFS as a file system to see the structure of the disks[3]. This means that ZFS can manage multiple file systems that can work across multiple disks, instead of the usual one file system to one disk system. The biggest advantage of using ZFS is that since it is aware of the physical disks, existing file systems can be grown simply by adding another disk to the memory pool[3].

ZFS is packed with many other features that set it apart from typical file systems, such as the fact that ZFS is a 128-bit file system. This means that theoretically the upper bound on the size of a single file is 16 exbibytes which is 2^{64} bytes[3]. ZFS also checksums the entire file tree to ensure data integrity. Each block of memory is checksumed, and the checksum value is then saved with the pointer to the block, not at the actual block itself. This continues on until even the root itself is checksumed creating a Merkle Tree or hash tree of the entire file system, so that the entire pool of memory can be verified. Since the checksum value of a block is stored in its parent block as a pointer, that block can still be verified if good or corrupt regardless of what someone tries to do to modify it[3].

REFERENCES

[1] S. S. A. W. Lars Wirzenius, Joanna Oja, "Linux system administrators guide." http://www.tldp.org/LDP/sag/html/filesystems.html. Accessed: 11-26-2016.
[2] T. Jones, "Anatomy of linux journaling file systems." http://www.ibm.com/developerworks/library/l-journaling-filesystems/. Accessed: 11-26-2016.
[3] T. F. D. Project, "Freebsd handbook, chapter 19." https://www.freebsd.org/doc/handbook/index.html. Accessed: 11-26-2016.