DOI: 10.5769/C2015010 or http://dx.doi.org/10.5769/C2015010

Estudo de Rótulos de Tempo em Sistemas de arquivos HFS+

Arelian Monteiro Maia, Felippe Pires Ferreira e Lindeberg Pessoa Leite

***Resumo*— Para análise pericial de um sistema de arquivo, os metadados armazenam dados relevantes, principalmente os rótulos de tempo. Dessa modo, este trabalho objetiva determinar o comportamento dos rótulos de tempo do sistema de arquivos HFS+ em diversos cenários na plataforma OS X, versões Mavericks e Yosemite. Em uma máquina virtual Mavericks e Yosemite, realizaram-se simulações de operações comuns com o intuito de entender o comportamento dos metadados de rótulos de tempo no sistema de arquivo HFS+. Para exposição dos resultados dos experimentos, foram elaboradas tabelas que apresentam o mapeamento entre a ação executada e as alterações nos rótulos de tempo.**

***Palavras-Chave*— Sistema de arquivo, Metadados, Rótulos de tempo, HFS+**

***Abstract*— For expert analysis of a file system, metadata store relevant data, especially the labels of time. In this way, this study aims to determine the behavior of the HFS+ File System timestamps in diverse scenarios in OS X platform, versions Mavericks e Yosemite. In a virtual machine Mavericks e Yosemite, there were simulations of common operations in order to understand the behavior of timestamps metadata in the HFS+ file system. To display the results of the experiments, tables were prepared presenting the mapping between the action taken and the changes on the labels of time.**

***Keywords*— File system, Metadatas, Timestamp, HFS+**

I. Introdução

Ao realizar uma análise pericial, informações temporais de um arquivo, como por exemplo, data de criação, data de modificação e data de acesso são elementos essenciais para criar uma linha do tempo (*timeline*). Entretanto, devido aos rótulos de tempo serem influenciados por vários fatores como o sistema de arquivo, *hardware*, o sistema operacional em execução e suas configurações, normalmente a extração das informações temporais não são diretas [1]. Desse modo, conhecer como são utilizados os registro de rótulos de tempo é fundamental para subsidiar Laudos periciais.

A demanda por exames periciais em dispositivos da Apple na Perícia da Polícia Federal vem crescendo, consequência do aumento de sua presença no mercado. Portanto, o estudo do funcionamento e utilização do sistema de arquivos destes dispositivos é uma necessidade.

O Hierarchical File System Plus (HFS+) é o principal sistema de arquivos da linha de produtos da Apple, substituindo o Hierarchical File System (HFS) em sistema Mac OS X e também é um dos formatos utilizados em sistema iOS. [7]

Arelian Monteiro Maia, Felippe Pires Ferreira e Lindeberg Pessoa Leite Mestrandos do Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília-DF, Brasil. E-mails: arelianmaia@gmail.com, felippepipipe@gmail.com, lindpessoa@gmail.com.

A. Objetivo

Este trabalho possui o objetivo de determinar o comportamento dos rótulos de tempo do sistema de arquivos HFS+ em diversos cenários na plataforma OS X, versões Mavericks e Yosemite.

B. Método

O método adotado neste trabalho foi executar em máquinas virtuais a plataforma OS X, nas versões Mavericks e Yosemite, realizando operações comuns de usuários como criar, copiar, mover, compactar, entre outras. A análise dessas operações na estrutura do sistema de arquivos HFS+ servirá de base para fazer afirmações acerca das alterações que os metadados de arquivos e diretórios sofrem e auxiliar a criação de uma *timeline*.

Foram criados dois cenários de teste. No primeiro cenário foi utilizado a interface gráfica para realizar as operações. Enquanto no segundo cenário foi utilizado linhas de comando para executar as operações. Com auxílio da interface gráfica do Mac OS X foram criadas duas partições HFS+. Em uma partição, criaram-se dez pastas, onze arquivos de texto e um arquivo de imagem. As operações foram executadas sobre esses objetos conforme descritas na tabela na seção Resultados.

As partições foram espelhadas e por meio do FTK Imager 3.2.0.0 [1] e do software HFSExplorer 0.23 [2] foram registrados os valores das datas de cada pasta e arquivo. Posteriormente, realizaram-se as operações de copiar, colar, mover, compactar, etc. Após isso, gerou-se uma nova imagem das partições para serem realizadas as comparações entre as datas antes e depois das operações de cada arquivo e pasta.

No segundo cenário, foi elaborado um script com os comandos de manipulação de arquivos e pastas. Antes da execução de cada comando de manipulação, os arquivos e pastas eram submetidos ao comando stat para registrar as datas antes da manipulação, e posteriormente as datas eram novamente chegadas para fins comparativos. Os dois cenários foram implementados com a utilização de discos rígidos e mídias removíveis.

[1]Forensic Toolkit, ou FTK, é um software de computação forense criado pela AccessData. Ele é capaz de processar um dispositivo, como um disco rígido, em busca de informações diversas. [9]

[2]É uma aplicação para visualizar e extrair arquivos de um volume HFS + ou em um volume HFSX, que estão localizados em uma unidade física, como uma imagem de um disco .dmg, ou em um dump de sistema de arquivos em formato raw[10]

C. Trabalhos correlatos

Um estudo do comportamento de rótulos de tempo em sistemas NTFS foi realizado por Junior, Cleber Scoralick [1]. Seu trabalho baseou-se em Chow et al [2], que apresentaram regras gerais baseadas em alguns dos rótulos de tempo existentes. Ele também inspirou-se em Bang et al. [3] e Bang, Yoo e Lee [4] que avaliaram mais rótulos de tempo e um número maior de operações. Este artigo busca realizar um estudo semelhante, mas aplicado ao sistema de arquivo HFS+.

II. Aspectos Técnicos

O HFS+, também chamado Mac OS Extended, foi introduzido em 1998 para superar os problemas da HFS e se tornar o sistema de arquivos principal usado em computadores Mac. HFS+ é uma versão melhorada do HFS suportando arquivos e volumes maiores pelo o uso de endereços de blocos de alocação de 32 bits e Unicode para nomes de arquivos. Ele também suporta múltiplos atributos para arquivos, como *journaling*, registros de textitinline attribute data, lista de controle de acesso baseado em arquivos de segurança e compatibilidade com os modelos de permissão de arquivo em outra plataformas como Windows. [5]

Assim como HFS, HFS+ divide o volume em setores de 512 bytes e agrupa-os em blocos de alocação, normalmente 8, e atribui a um arquivo. Blocos de alocação são endereçados por ponteiros de 32 bits [6]. Para reduzir a fragmentação do arquivo, blocos de alocação contíguas chamados *Clumps* são atribuídos aos arquivos. O número de blocos de alocação por *Clump* é fixa e é especificado em Volume Header. Os primeiros 1024 bytes e últimos 512 bytes de volume são reservados. O *Volume Header* está localizado imediatamente após primeiros 1024 bytes e é fixo. O *Alternate Volume Header* que é réplica do *Volume Header* está localizado nos 1024 bytes antes do final do volume e também é fixo. [7]

O *Volume Header* armazena rótulos do tempo, o número de arquivos sobre o volume, localização de outras estruturas sobre o volume, tamanho de blocos de alocação, tamanho de aglomerados, etc. [7]

Um volume HFS+ tem cinco arquivos especiais que são utilizados para organizar o espaço do volume utilizado para armazenar pastas, arquivos e atributos. São eles:

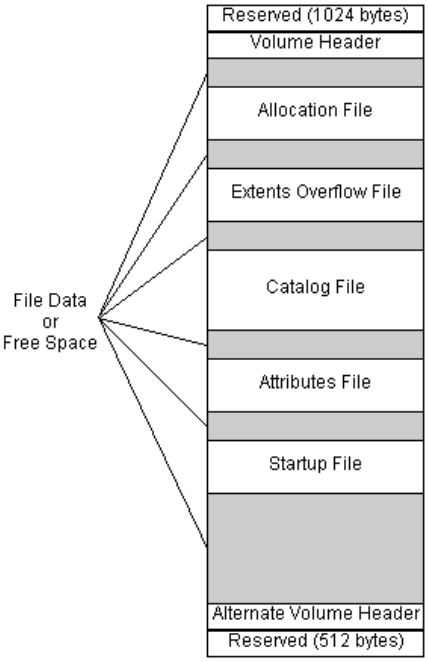

Fig 1. Layout do HFS+ [5]

A. Allocation File

O *Allocation File* é um bitmap que controla a utilização dos blocos de alocação. Ele mantém o controle do que está livre e em uso pela representação de cada bloco por um bit. Isso é equivalente ao *Bitmap Volume* do HFS. A principal diferença entre *Bitmap Volume* e *Allocation File* é que este é um arquivo comum que pode existir em qualquer lugar no volume, podendo diminuir ou aumentar de tamanho e não precisa ser contíguo. Enquanto aquele sempre reside em área reservada e seu tamanho é fixo. A localização do primeiro extent do *Allocation File* é armazenado em *Volume Header*. Esta arquitetura de *File Allocation* induz a flexibilidade no sistema de arquivos HFS+ não encontrada em HFS. [7]

B. Catalog File

O *catalog file* é uma árvore-B que armazena a hierarquia de pastas e arquivos. Ele descreve todos os arquivos e pastas do volume incluindo os arquivos especiais e da hierarquia no volume. É semelhante ao *Catalog File* do HFS. O *Catalog File* é organizado em uma árvore-B para permitir pesquisas rápida e eficientes por meio de uma grande hierarquia. Este arquivo contém informações vitais sobre cada arquivo e pasta juntamente com as informações do catálogo. A principal diferença entre os registros em HFS e HFS+ é que no *Catalog File* em HFS+ os nós da árvore-B relativo aos arquivos e pastas contêm mais informações e podem ter diferentes tamanho ao contrário do HFS. A localização do primeiro extent do *Catalog File* é armazenado no *Volume Header*. *Catalog File* contém nó de cabeçalho, nós índices, nós folhas e, se necessário, mapa de nós. Cada arquivo ou pasta do *Catalog File* é identificado por um único *Catalog Node ID* (CNID). Para pasta, CNID é chamado FolderID e para arquivos FileID. [7]

C. *Extent Overflow File*

O *Extents Overflow File* é utilizado para mapear os extents (áreas contíguas de um arquivo) extras dos arquivos que contêm mais que oito extents. Os primeiros oito extents são listados no registro correspondente ao arquivo no *catalog file*. Está estruturado como uma árvore-B. [7]

D. *Bad Block File*

Bad Block File é usado para marcar e registrar as áreas do volume que contém blocos danificados. O *Extent Overflow File* é usado para armazenar informações sobre os extents de *Bad Block File*. [7]

E. *Attributes File*

O *attribute file* contém metadados adicionados em pastas e arquivos pelas aplicações. Ele é um arquivo especial que não possui uma entrada no *Catalog File*. Um volume não pode ter *Attributes File* em caso de sua descrição no *Volume Header* para alocação de blocos seja 0. *Attributes File* é uma Arquivo B-tree estruturado onde os nós podem conter registros conhecidos como atributos. Um *Attribute File* pode ter 3 tipos de atributos [7]:

- *Inline Data Attributes* que contêm pequena atributos;
- *Attributes Data Fork* que contêm referências para um máximo de 8 extents;
- *Extended Attributes* que contêm referências a mais 8 extents para os atributos de dados;

F. *Startup File*

O *startup file* é um arquivo especial que facilita o boot em computadores não-Mac. O *boot loader* pode encontrar o arquivo *startup File* sem total conhecimento do formato de uma partição HFS+. Em vez disso, o *Volume Header* contém a localização dos primeiros oito extents do *startup File*. Este arquivo pode conter mais de oito extents, os quais serão colocados no *Extents Overflow File*. [7]

G. *Rótulo de tempo no HFS+*

HFS+ armazena rótulos de tempo em várias estruturas de dados, incluindo *Volume Header* e registros de catálogo. Estas datas são armazenados em inteiros de 32 bits sem sinal (UInt32) contendo o número de segundos desde 01/jan/1904 00:00:00 GMT, tendo como data máxima 06/fev/2040 06:28:15 GMT. Como as datas registradas estão entre 1900 e 2100, não se considera os segundos bissextos. [5]

A implementação é responsável por converter esses tempos para o formato esperado pelo software cliente. Por exemplo, o gerenciador de arquivos do Mac OS converte rótulos de tempo para hora local; a implementação Mac OS HFS+ converte a hora local para GMT, quando apropriado. [5]

A documentação oficial da Apple sobre a implementação do HFS+ [8], encontram-se descritos rótulos de tempo no *volume header/alternate volume header* e nas informações sobre pastas e arquivos no *Catalog File*:

Volume Header e *Alternate Volume Header* [7]

- *createDate* - Rótulo de tempo de quando o volume foi inicializado.
- *modifyDate* - Rótulo de tempo de quando o volume foi modificado pela última vez.
- *backupDate* - Rótulo de tempo de quando foi feito o último backup do volume. Deve ser atualizado por alguma aplicação de backup e não pelo sistema.
- *checkedDate* - Rótulo de tempo de quando foi realizado a última verificação de consistência no volume. Tipicamente alterado na utilização de ferramentas de checagem de disco como *Disk First Aid*.

Registro de pasta no *catalog file* [7]

- *createDate* - Rótulo de tempo de quando a pasta foi criada. Diferentemente da *createDate* do *volume header*, essa data está armazenada em GMT.
- *contentModDate* - Rótulo de tempo da última modificação do conteúdo da pasta, isto é, a última vez em que um arquivo ou pasta foi criado ou deletado dentro dessa pasta, ou quando um arquivo ou pasta foi movido para outra pasta.
- *attributeModDate* - Rótulo de tempo da última vez em que qualquer campo no registro de catálogo da pasta foi alterado.
- *accessDate* - Rótulo de tempo em que o conteúdo da pasta foi lido pela última vez. Criado para compatibilidade do Mac OS X com o POSIX, tem o valor de zero quando criado pelo Mac OS tradicional.
- *backupDate* - Rótulo de tempo de quando foi feito o último backup da pasta. Deve ser atualizado por alguma aplicação de backup e não pelo sistema.

Registro de arquivos no *Catalog File* [7]

- *createDate* - Rótulo de tempo de quando o arquivo foi criado. Diferentemente da *createDate* do *volume header*, essa data está armazenada em GMT.
- *contentModDate* - Rótulo de tempo em que o conteúdo do arquivo foi modificado. Entenda-se conteúdo não apenas os dados do arquivo, mas qualquer informação associada a ele (resource) como ícone, vídeo *QuickTime*, som, e outros.
- *attributeModDate* - Rótulo de tempo da última vez em que qualquer campo no registro de catálogo do arquivo foi alterado.
- *accessDate* - Rótulo de tempo em que o conteúdo do arquivo foi lido pela última vez. Criado para compatibilidade do Mac OS X com o POSIX, tem o valor de zero quando criado pelo Mac OS tradicional.
- *backupDate* - Rótulo de tempo de quando foi feito o último backup da pasta. Deve ser atualizado por alguma aplicação de backup e não pelo sistema.

III. Resultados obtidos

As tabelas abaixo mostram os resutados obtidos no OS X, versões Mavericks 10.9.0 e Yosemite 10.10.3 com o sistema de arquivo HFS+, versão 4. Os softwares FTK Imager 3.2 e HSFExplorer 0.23 foram utilizados para acessar a estrutura do sistema de arquivos.

Ação	Data de Criação	Data de Modificação	Data de Alteração de Atributos	Data de Acesso	Data de Backup
Mover diretório na mesma partição					
Mover arquivo na mesma partição				x	
Copiar diretório na mesma partição			x	x	
Copiar arquivo na mesma partição			x	x	
Mover diretório para partição diferente			X (diretório na partição de destino alterado)	X (diretório na partição de origem e de destino alterados)	
Mover arquivo para partição diferente			X (arquivo na partição de destino alterado)	X (arquivo na partição de origem e de destino alterados)	
Copiar diretório para partição diferente			X (diretório na partição	X (diretório na partição	

Tabela 1: Pela interface gráfica do sistema operacional

Ação	Data de Criação	Data de Modificação	Data de Alteração de Atributos	Data de Acesso
Alterar permissões do arquivo: chmod xxx			X	
Alterar permissões de pasta: chmod xxx			X	
Arquivo oculto Chflags hidden			X	
Pasta oculta Chflags hidden			X	

Tabela 3: Linha de comando para manipulação de atributos

Ação	Data de Criação	Data de Modificação	Data de Alteração de Atributos	Data de Acesso
Cat				X
Sed				x

Tabela 4: Linha de comando para leitura de arquivo

Ação	Data de Criação	Data de Modificação	Data de Alteração de Atributos	Data de Acesso
Compactar Pasta: tar -cf				x
Compactar Arquivo: tar -cf				x
Descompactar Pasta: tar –xf	x		x	X
Descompactar Arquivo: tar –xf			x	x
Arquivo .tar após descompactação				x
Compactar Arquivo: gzip			x	
Descompactar Arquivo: gzip -d			x	x
Compactar Pasta: zip				
Compactar Arquivo: zip				x
Descompactar Pasta: unzip	x		x	
Descompactar Arquivo: unzip	x		x	x
Arquivo .zip após descompactação				x

Tabela 2: Linha de comando para compactação e descompactação

Ação	Data de Criação	Data de Modificação	Data de Alteração de Atributos	Data de Acesso
Ls				X
Cd				
Criação de diretório filho (mkdir) alteração no diretório pai		x	x	
Remoção de arquivo (rm) alteração no diretório pai				x
Remoção de diretório filho (rm -rf) alteração no diretório pai		x	x	
Cópia de arquivo (cp) na mesma partição:				
Arquivo na origem				x
Arquivo no destino	x	X	X	x
Diretório de origem		X	X	
Diretório de destino		x	x	
Cópia de arquivo (cp) em outra partição:				
Arquivo na origem				x
Arquivo no destino	X	X	X	x
Cópia de arquivo (cp -a) preservando atributos:				
Arquivo na origem				x
Arquivo no destino			x	
Cópia de diretório (cp -a) preservando atributos:				
Diretório de origem				x
Diretório de destino			x	x
Cópia de diretório (cp -r) na mesma partição:				
Diretório de origem				
Diretório de destino				
Diretório filho	x	x	x	x

Tabela 5: Linha de comando para manipulação

IV. Conclusões

Este artigo objetivou determinar o comportamento dos rótulos de tempo do sistema de arquivo HFS+ em diversos cenários na plataforma OS X, versões Mavericks e Yosemite.

É possível observar comportamentos semelhantes em alguns comandos executados pela interface gráfica e pela interface de linha de comando. Entretanto, não é possível afirmar isso em todos os casos, visto que a implementação da ação pode executar instruções diferentes ou sequências diferentes. Um dos comportamentos semelhantes observados foi a execução de compactação e descompactação utilizando o formato .zip. tempo.

Algumas ações semelhantes tiveram comportamentos diferentes, como a manipulação de arquivos e diretórios. Enquanto na utilização de interface gráfica a data de criação foi preservada, em alguns casos de execução de comandos pelo terminal esta data sofreu alterações.

Grande parte dos comandos utilizados fazem alterações na data de acesso do objeto, com exceção dos comandos de manipulação de atributo. Logo, ações como leitura, listagem ou cópia fazem alterações neste campo. Uma *timeline* baseada neste campo pode auxiliar na criação de uma trajetória de utilização do sistema de arquivos pelo usuário.

O Backup descrito nos resultados da seção anterior foi realizado através do software *Time Machine* do Mac OS X. Os arquivos e diretórios originais que sofreram backup não tiveram suas datas alteradas, entretanto, os arquivos resultantes do procedimento sofreram alterações em suas datas de modificação de atributos e de acesso. Apesar de existir um atributo de data de backup (*backupDate*) em cada arquivo e pasta, este atributo não foi alterado após a execução do backup.

O experimento foi realizado com discos rígidos e removíveis com o sistema operacional em estudo. Foram simuladas operações de arquivos entre discos rígidos, entre discos removíveis e entre disco rígido e disco removível. Apesar da variação das mídias utilizadas, os resultados obtidos foram os mesmos, independente se eram discos rígidos ou removíveis. O que diferenciava os resultados eram apenas os comandos executados ou as partições envolvidas. Logo, a cópia de um arquivo entre discos rígidos apresentava o mesmo comportamento que a cópia de um arquivo entre discos removíveis. Mas apresentava resultados diferentes a depender se a copia ocorreria entre diferentes partições ou se a cópia ocorria na mesma unidade.

Como trabalho futuro, pode-se usar as datas para criar uma ferramenta de *timeline*. Desse modo, é possível verificar a consistência nas datas, de modo a encontrar possíveis adulterações intencionais. Ademais, repetir os testes no hfsx e em outros sistemas operacionais como o GNU/Linux para verificar se o comportamento descrito neste artigo se mantém quando executado em outras plataformas ou versões do sistema de arquivos. Por fim, é possível ainda analisar o comportamento dos CNID (catalog node ID) e relacioná-lo com as datas de criação.

Referências

[1] Júnior, Cleber Scoralick, *Estudo de Rótulos de Tempo em Sistemas NTFS Baseado em estrutura do Sistema de Arquivos e do Sistema Operacional*. Mestrado Profissional em Informática Forense e Segurança da Informação - UNB, 2012.

[2] CHOW, K. et al, *The rules of time on ntfs File system. Systematic Approaches to Digital Forensic Engineering, IEEE International Workshop on, IEEE Computer Society, Los Alamitos*, 2007

[3] BANG, J. et al, *Analysis of time information for digital investigation. Networked Computing and Advanced Information Management, International Conference on, IEEE Computer Society, Los Alamitos, CA*, 2009

[4] BANG, J.; YOO, B.; LEE, *S. Analysis of changes in File time attributes with File manipulation*, 2011

[5] TN1150, *HFS Plus Volume Format, http://developer.apple.com/*, Acessado em abril de 2015

[6] Mac OS X, *Mac OS Extended format (HFS Plus) volume and file limits, http://support.apple.com/*, Acessado em abril de 2015

[7] Wasim Ahmad Bhat , S. M. K. Quadri, *A Quick Review of On-Disk Layout of Some Popular Disk File Systems*, Global Journal of Computer Science & Technology, 2011

[8] Technical Note TN1150, *HFS Plus Dates, http://dubeiko.com/development/FileSystems/HFSPLUS/tn1150.html*, Acessada em 23/04/2015

[9] FTK Imager, *http://en.wikipedia.org/wiki/Forensic_Toolkit*, Acesso 27/05/2015

[10] HFSExplorer, *http://en.wikipedia.org/wiki/Forensic_Toolkit*, Acessado em 27/05/2015

[11] A Quick Review of On-Disk Layout of Some Popular Disk File Systems, *http://en.wikipedia.org/wiki/Forensic_Toolkit*, Acessado em 27/05/2015