|  |  |
| --- | --- |
|  | **FIPP – SISTEMAS DE INFORMAÇÃO – PRESIDENTE PRUDENTE** |

**ANDRESSA HISAE TSUKASAKI – R.A. 261742078  
GABRIEL MIGUEL NAVAS – R.A. 261741888  
GUSTAVO DILLIO LEITÃO – R.A. 261741748**

**ATIVIDADE DE PESQUISA SOBRE EXT3 X NTFS**

Presidente Prudente - SP

2019

TRABALHOS SEM REFERENCIA BIBLIOGRÁFICA SERÃO DESCOSIDERADOS - PELO MENOS UM LIVRO DEVE SER UTILIZADO (Recomendo fortemente a leitura das seções referentes a sistemas de arquivos, dos capítulos 10 e 11, do livro Sistemas Operacionais Modernos 2ed. - autor: Andrew S. Tanenbaum).

Data de entrega: 22/03

**1.** *Descreva a estrutura (struct real) de alocação de arquivos do NTFS, conhecida como MFT (Master File Table) e a* *estrutura do registro da entrada de diretório do NTFS. Quero a struct em linguagem C.*

O MFT é uma tabela que registra atributos de cada arquivo armazenado. Esses atributos consistem em uma série de informações, entre elas: nome, data da última modificação, permissões (conceito explicado no tópico anterior) e, principalmente, localização na unidade de armazenamento. Como necessita guardar várias informações de praticamente todos os arquivos no disco, o NTFS reserva um espaço para o MFT - Zona MFT -, geralmente de 12,5% do tamanho da partição. Cada arquivo pode necessitar de pelo menos 1 KB para o registro de seus atributos no MFT, daí a necessidade de um espaço considerável para este.

Em relação a estrutura do registro da entrada de diretório:

* Arquivo especial que mantém registro sobre arquivos e diretórios.
* Um registro é composto por: Nome do arquivo, Atributos de base, Referência para um elemento MFT.
* Sua estrutura depende do seu tamanho.
* Diretórios pequenos são representados como lista linear de arquivos.
* Diretórios grandes usam arvore B+.
* Ordenação alfabética.
* Diretório raiz possui uma referência no MFT.
* Utiliza uma estrutura de árvore para todos os diretórios do disco com a finalidade de reduzir o número de acessos ao disco para localizar arquivos.
* Este sistema é o melhor para diretórios grandes.

struct reg\_atributos\_mft

{

char info\_padrao[];

char nome\_arquivo[];

char desc\_seguranca[];

char lista\_atributos[][];

long int id\_objeto;

char ponto\_analise[];

char nome\_volume[];

char info\_volume[];

char indice\_raiz[];x

char indice\_alocacao[];

char mapa\_bits;

char registro\_fluxo\_utilidade;

char dados[];

};

struct reg\_mft

{

char cab\_registro[]

char cab\_info\_padrao[];

char nome\_arq[];

char cab\_dados[];

int blocos\_disco[][];

};

**2.** *Descreva a estrutura (struct real) de alocação do I-node e a estrutura do registro da entrada de diretório do Linux. Quero a struct em linguagem C.*

struct inode

{

short int mode;

short int nlinks;

short int uid;

short int gid;

int size;

int addr[39];

char gen;

char atime[4];

char mtime[4];

char ctime[4];

};

struct in\_directory

{

short int num\_inode;

char file\_name[14];

};

**3.** *Compare  um i-node a um registro da MFT em relação aos seguintes aspectos:***a)** *Onde são armazenados os atributos em cada uma dessas estruturas de alocação?*

No caso do windows, como cada registro da MFT (master file table) descreve somente um arquivo ou diretório, neste registro há os atributos com todos atributos (como já descrito na questão a cima).

No caso do linux, todos atributos de arquivo ficam dentro de um i-node.

**b)** *Em sistemas baseados em i-nodes, quando um arquivo cresce além do limite da estrutura de alocação, são utilizados blocos indiretos para acomodar os endereços do arquivo. Explique o que precisa ser feito quando um arquivo cresce além do limite de espaço de um registro da MFT.*

Se um arquivo aumentar de tamanho, será necessário, algumas vezes, usar dois ou mais registros da MFT para brigar a lista de todos os blocos. Nesse caso, o primeiro registro MFT, chamado de registro-base, aponta para os outros registros da MFT. Um mapa de bits faz o acompanhamento de quais entradas da MFT estão livres.

**3.** *Descreva o conceito de grupo de blocos do EXT3. Qual a vantagem do EXT3 utilizar grupos de blocos ao invés da organização tradicional dos sistemas de arquivos (metadados no inicio do disco e dados de arquivos e diretórios no restante do disco)?*

O disco formatado por ext3, é formado por blocos, onde o primeiro bloco (bloco 0) não é utilizado pelo Linux (contem código de inicializar o computador). Após o bloco 0, a partição do disco é dividida em grupos de blocos, desconsiderando onde vão cair os limites do cilindro.

Descrito em ordem, cada bloco contém:

\* Superbloco: espaço responsável por guardar a informações sobre o layout do sistema de arquivos incluindo o número de i-nodes, o número de blocos de disco e o começo da lista de blocos de disco livres (em geral algumas centenas de entradas).

\* Descritor do grupo: contém informações sobre a localização dos mapas de bits, o número de blocos livres e i-nodes no grupo, e o número de diretórios no grupo.

\* Mapa de bits e Mapa de bits do i-node: Mapas de bits são usados para controlar os blocos livres e i-nodos livres.

\* I-nodes: Eles são numerados de 1 até algum máximo. Cada i-nodo tem 128 bytes de comprimento e descreve exatamente um arquivo. Um i-nodo contém informações de contabilidade, assim como informações suficientes para localizar todos os blocos de disco que contêm os dados do arquivo.

\* Blocos de dados: Todos os arquivos e diretórios estão armazenados aqui. Se um arquivo ou diretório consiste em mais do que um bloco, os blocos não precisam ser contíguos no disco. Na realidade, os blocos de um arquivo grande têm uma maior chance de disseminar-se por todo o disco.

A Grande vantagem de se utilizar esse tipo de organização em grupo de blocos, é que quando os arquivos crescem, são colocados junto ao grupo de blocos, preferivelmente logo após o último bloco anterior. Além disso, quando um novo arquivo é inserido em um diretório, o ext3 tenta colocar dentro do mesmo grupo de blocos em que está seu diretório. Quando novos blocos de arquivos são alocados, ext2 também pré-aloca um número (oito) de blocos adicionais para aquele arquivo, de maneira a minimizar a fragmentação de arquivos devido a futuras operações de escrita. Esse esquema equilibra a carga do sistema de arquivos através do disco inteiro. Ele também tem um bom desempenho devido a suas tendências para colocação e fragmentação reduzida.

Referências:

Para escrever certinho: SOBRENOME, Nome Abreviado. Título: subtítulo (se houver). Edição (se houver). Local de publicação: Editora, data de publicação da obra.

* Tanenbaum, Andrew S. Sistemas Operacionais Modernos. 2ed. São Paulo: Prentice Hall, 2003.
* Tanenbaum, Andrew S. Sistemas Operacionais Modernos 3ed. São Paulo: Pearson Prentice Hall, 2009.