Sistemas Operacionais - Trabalho Prático II

Sistema de arquivos FAT e um simples *shell*

O segundo trabalho pático da disciplina de Sistemas Operacionais consiste na implementação de um simulador de um sistema de arquivos simples baseado em tabela de alocação de 32 bits (FAT) e um *shell* usado para realizar operações sobre este sistema de arquivos.

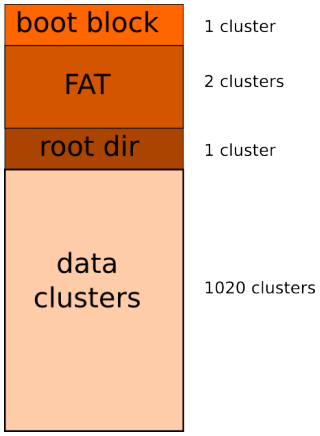
O sistema de arquivos virtual

O sistema de arquivos deverá ser armazenado em uma partição virtual e suas estruturas de dados mantidas em um único arquivo nomeado *fat.part*. A partição virtual terá um tamanho total determinado por:

- 512 bytes por setor;
- cluster de 2048 bytes (4 setores por cluster);
- 1024 clusters;

Dessa forma, seu tamanho pode ser calculado por 512 bytes por setor * 4 setores por cluster * 1024 clusters = 2097152 bytes (2MB). Os dados devem ser alocados sempre em *clusters*, ou seja, um arquivo ocupará fisicamente no mínimo um cluster (2048 bytes) no sistema de arquivos virtual.

O primeiro cluster é definido como *boot block*, e conterá informações referentes ao volume (partição). Por motivos de simplificação, o *boot block* terá o tamanho de 1 cluster (2048 bytes) e não 1 setor (512 bytes) como seria o usual, e deve ser preenchido com o valor 0xbb. A FAT terá um tamanho determinado por 1024 clusters de dados * 4 bytes por entrada (32 bits) = 1024 entradas de 32 bits = 4096 bytes (2 clusters), inicializada com 0x00. O diretório *root* estará localizado logo após a FAT e também terá um tamanho de 1 cluster. O diretório *root* possui um conjunto de entradas de diretório que



podem apontar para outros diretórios ou arquivos. Inicialmente, as entradas de diretório devem estar livres, inicializando-se todas as estruturas com 0x00.

Após a FAT e o diretório *root*, encontra-se a seção de dados contendo o restante dos clusters. Outros diretórios (e sub-diretórios) são definidos como clusters que possuem diversas entradas de diretório (assim como o diretório *root*), possuindo uma estrutura apresentada adiante.

Detalhes sobre o sistema de arquivos

O sistema de arquivos possui uma série de limitações, que foram determinadas com o intuito de simplificar a implementação do trabalho. A primeira limitação refere-se ao tamanho da FAT, onde é possível armazenar apenas 1024 entradas para blocos, o que limita o tamanho da partição virtual em 2MB. Se mais entradas fossem necessárias (para um disco maior), seriam necessários blocos adicionais para a FAT. A segunda limitação refere-se ao número de entradas de diretório em cada nível da árvore. Cada entrada ocupa 32 bytes, o que limita o número de entradas de diretório em 64, tanto no diretório raiz quanto em sub-diretórios.

Não será permitido o uso de trapaças para a manipulação das estruturas de dados (como ler todo o sistema de arquivos para a memória para manipular as estruturas. Deve-se ler e escrever sempre utilizando a unidade *cluster*, independente de ser a FAT, diretório ou bloco de dados de arquivo. Sugere-se manter três blocos de 2048 bytes em memória - FAT (1 dos 2 blocos), bloco com entrada de diretório e um bloco de dados. Não esqueça de após manipular a FAT ou dados de atualizar o sistema de arquivos virtual nas entradas de diretório. Lembre-se que o sistema precisa manter-se consistente, ao ponto de poder ser recuperado em qualquer instante.

- Informações sobre o valor das entradas na FAT de 32 bits:

```
0x00000000                -> cluster livre
0x00000002 - 0xfffffffe   -> arquivo (ponteiro p/ proximo cluster)
0xffffffff                -> fim do arquivo
```

- Informações sobre a estrutura das entradas de diretório:

```
16 bytes          -> nome do arquivo
1 byte            -> atributo do arquivo
7 bytes           -> reservado
4 bytes           -> numero do primeiro cluster ocupado
4 bytes           -> tamanho do arquivo
```

Byte de atributo do arquivo - valor: 0 - arquivo, 1 - diretório.

- Estruturas pré-definidas (usadas como referência)

```
/* 1 cluster da tabela FAT, 512 entradas de 32 bits */

uint32_t fat[512];

/* entrada de diretorio, 32 bytes cada */

typedef struct dir_entry{
    uint8_t filename[16];
    uint8_t attributes;
    uint8_t reserved[7];
    uint32_t first_block;
    uint32_t size;
};

/* diretorios (incluindo ROOT), 64 entradas de diretorio
com 32 bytes cada = 2048 bytes */

dir_entry dir[64];

/* bloco de dados, 2048 bytes */

uint8_t data_block[2048];
```

Depuração

Sugere-se utilizar a ferramenta de sistema *hexdump* para realizar a depuração do sistema de arquivos virtual.

Detalhes sobre o shell

Um pequeno *shell* deve ser implementado para a manipulação do sistema de arquivos. Este shell deve oferecer recursos para a carga do sistema de arquivos e manipulação de diretórios e arquivos. Os seguintes comandos devem ser implementados no shell:

- init - inicializar o sistema de arquivos com as estruturas de dados, semelhante a formatar o sistema de arquivos virtual
- load - carregar o sistema de arquivos do disco
- ls [/caminho/diretorio] - listar diretório
- mkdir [/caminho/diretorio] - criar diretório
- rmdir [/caminho/diretorio] - remover diretório
- create [/caminho/arquivo] - criar arquivo
- rm [/caminho/arquivo] - excluir arquivo
- write "string" [/caminho/arquivo] - anexar dados em um arquivo
- cat [/caminho/arquivo] - ler o conteúdo de um arquivo

Entrega e apresentação

O trabalho deverá ser realizado em duplas. Qualquer linguagem de programação pode ser utilizada (preferencialmente C) para o desenvolvimento do trabalho, desde que as estruturas de dados que implementam o sistema de arquivos sejam manipuladas e armazenadas de acordo com a especificação. A entrega do trabalho deverá ser realizada pelo moodle em um arquivo *.tar.gz* contendo a implementação, instruções de uso e nome dos integrantes. A apresentação do trabalho será realizada em aula.