



UNIVERSIDADE FEDERAL DE SÃO CARLOS

Bacharelado em Ciência da Computação

Laboratório de Sistemas Operacionais

Professor Gustavo Maciel Dias Vieira

Campus Sorocaba

Projeto Final

Sistema de Arquivos

Daniel Ramos Miola 438340

Giuliano Raphael Sbrugnera 408093

Igor Felipe Ferreira Ceridório 408611

Matheus Casarin Paez 438308

Sorocaba

2013

1. Introdução

O Projeto Final têm por objetivo criar um sistema de arquivos simples, aprofundando o entendimento de como funciona a abstração de arquivos em dispositivos de discos.

É utilizado um interpretador de comandos para criar a conexão entre determinados comandos e seus efeitos no sistema de arquivos. A partir desse interpretador, o sistema de arquivos pode ser manipulado e testado, de acordo com as suas funcionalidades implementadas. Os comandos realizam funções básicas de um sistema de arquivos elementar, como formatar o sistema, criar e remover um arquivo, assim como copiar um arquivo do sistema virtual para o sistema real e vice-versa.

2. Discussão e Resultados

Todas as implementações das funções foram feitas levando em consideração a estrutura do disco de blocos de 512 bytes e a tabela com agrupamento de 8 blocos, onde a FAT de 32 agrupamentos que pode referenciar no máximo 256MiB e o diretório de nível único pode conter no máximo 128 arquivos. Estruturas essas que ocupam os primeiros 33 agrupamentos do disco (32 ou 256 blocos para FAT e 1 ou 8 blocos para o diretório) em qualquer disco idealizado para o projeto. As funções de interface do sistema de arquivos elementar implementado neste projeto foram divididas em dois grupos principais que são descritos a seguir.

2.1. Funções de Gerência de Arquivos

As funções de gerência dos arquivos no projeto são responsáveis por gerenciar o sistema quanto a suas aplicações mais básicas, em especial iniciar, formatar o sistema de arquivos e criar e remover arquivos do sistema. Segue a descrição de cada função implementada:

- **fs_init:** função de inicialização do sistema de arquivos a qual carrega a tabela de alocações do disco e verifica se o mesmo está formatado. São recuperados do disco os primeiros 32 agrupamentos (256 blocos) contíguos onde se localiza a tabela de alocação do sistema e o próximo agrupamento (8 blocos) onde fica o único nível de diretórios presente. Após carregá-los é feita a verificação dos valores para constatar se o sistema está formatado.

- **fs_format:** função que varre o vetor da tabela de alocação em memória atribuindo os valores corretos às partições do vetor que endereçam à própria FAT e diretórios, assim como marca como livres todos os outros endereços da tabela e marca como não usados (F) todos os arquivos do diretório indicando que o disco está totalmente livre, e por último gravando as alterações em disco.

- **fs_free:** função que retorna o tamanho do espaço livre do disco em bytes. De forma simples varre a tabela de alocações contando o número de agrupamentos livres até que o máximo de agrupamentos possível para o disco em uso seja alcançado e retorna o valor em bytes

multiplicando os agrupamentos por 4kiB.

- **fs_list:** lista os arquivos do diretório seguidos de seu tamanho varrendo o vetor de estruturas do diretório e concatenando em um buffer os atributos dos arquivos que estão em uso no disco.

- **fs_create:** função que cria um arquivo de tamanho zero no sistema. Primeiramente é verificado se já existe um arquivo em uso com o mesmo nome, e então são atualizados seus atributos na primeira estrutura livre do diretório e é reservado para o mesmo o primeiro agrupamento livre da tabela de alocação, apesar do arquivo ter tamanho 0 é reservado para o mesmo um agrupamento na FAT evitando assim que haja arquivos na estrutura de diretórios sem opção nenhuma de agrupamentos livres em disco. E então é atualizado o sistema de arquivos no disco para evitar perda de arquivos por falha.

- **fs_remove:** responsável por remover um arquivo do sistema pelo nome, a função procura o arquivo no diretório e então o marca como não usado(F) e então chama uma função recursiva que varre a tabela de alocação a partir do primeiro índice de primeiro bloco do arquivo até o último sinalizado com 2, e retorna marcando-os como livres na estrutura. Os dados do arquivo em disco são mantidos, porém ao marcar suas posições como livres apenas se evita o trabalho de apagar os dados de forma real.

2.2. Funções de Leitura e Escrita de Arquivos

Essas funções do sistema de arquivos são responsáveis pelo gerenciamento do estado dos arquivos dentro do sistema bem como a escrita e leitura dos dados armazenados nos blocos de cada arquivo. Segue a descrição da função concluída:

- **fs_open:** função responsável por abrir um arquivo para leitura ou escrita de acordo com os parâmetros obtidos. Primeiramente caso o arquivo deva ser aberto para escrita a função faz a busca na estrutura do diretório verificando se o arquivo desejado existe e então o apaga e cria um de mesmo nome usando as funções `fs_create` e `fs_remove` implementadas anteriormente, caso o arquivo não exista é criado com o nome buscado, é marcado então na estrutura do arquivo no diretório que o mesmo está em uso de escrita(W). Analogamente, para um arquivo ser aberto para leitura, é feito a busca e gerado um novo arquivo caso este não exista ou sua estrutura é simplesmente marcada como em uso de leitura(R). Em ambos os casos, escrita ou leitura, a tabela de alocação e os diretórios são atualizados em disco após a alteração e é retornado o índice da estrutura do mesmo no diretório.

Com as abstrações oferecidas previamente pelo professor para a realização do trabalho, a realização da tarefa se fez de forma constante e sem dificuldades para gerar as funções requisitadas para entrega, de modo que podemos afirmar que houve o pleno entendimento do funcionamento de um sistema de arquivos elementar baseado em tabela de alocações, salvo os imprevistos já listados.

3. Dificuldades encontradas na realização

A principal dificuldade na realização da tarefa foi em relação ao entendimento do código da abstração fornecida pelo professor acarretando em muito tempo desperdiçado somente para essa tarefa. Tempo esse que mal gerenciado ocasionou na não conclusão do projeto final como um todo, faltando as funções `fs_close`, `fs_write` e `fs_read`.

4. Conclusão

Concluimos que com as abstrações oferecidas previamente pelo professor para a realização do trabalho, a tarefa se desenrolou de forma constante e sem dificuldades para gerar as funções requisitadas para entrega, de modo que podemos afirmar que houve o pleno entendimento do funcionamento de um sistema de arquivos elementar baseado em tabela de alocações, salvo os imprevistos e não conclusão já listados.

5. Bibliografia

[1] Slides do curso Laboratório de Sistemas Operacionais – UFSCar Sorocaba

[2] Slides do curso Sistemas Operacionais – UFSCar Sorocaba