

Implementação de um Sistema de Arquivos

A princípio, analisamos diversas estruturas de dados que poderiam armazenar os diretórios visando uma boa eficiência na busca de arquivos e com um baixo custo para estruturá-lo. Assim, optamos pela implementação da árvore binária, já que na sua inserção ela organiza as entradas de diretório ordenadas pelos nomes dos arquivos, de modo que não é necessário outro algoritmo de ordenação. Desta forma, estruturado de maneira alfabética, no qual o nó esquerdo contém caracteres precedentes na ordem alfabética consequentemente o nó direito possui caracteres subsequentes. A partir disso, facilitando no momento de realizar a inserção, busca e remoção, pois para achar a posição do elemento basta comparar se o nome do arquivo desejado é subsequente ou precedente que o nó atual provendo uma melhor eficiência já que não é necessário percorrer todos os elementos da árvore para achar a posição desejada. Desta maneira, seguindo o princípio da busca binária a cada comparação, descarta-se metade dos possíveis caminhos.

Ademais, outra estrutura de dados implementada foi a lista de inodes, no qual proporcionou um maior desempenho ao criar um arquivo ou um diretório, já que para sua alocação basta pegar a primeira célula da lista, o qual se encontra no superblock e atualizá-lo. Enquanto, na sua remoção é necessário apenas resetar as informações e o inserir no início da lista.

A estrutura de gerenciamento de espaços livres utilizada foi o mapa de bits, que é estruturado armazenando um bit para cada bloco, sendo assim 0 para livre e 1 para ocupado, de modo que são representados os blocos de forma sequencial e ordenada. Outrossim, essa característica torna o mapa de bits vantajoso em sistemas com grande quantidade de blocos livres, uma vez que seu tamanho é fixo. Em contrapartida, a lista de livres cresce proporcionalmente ao número de blocos livres, dado que a lista deve armazenar o endereço do bloco em disco e o ponteiro para o próximo. No entanto, quando a memória

secundária estiver cheia ou próximo a isso, a lista se torna menor que o mapa já que haverá poucos blocos livres.

Sobretudo, o mapa de bits apresenta maior desempenho e eficiência principalmente na alocação de blocos, pois normalmente consegue armazená-lo inteiramente em memória. Já a lista carrega para memória apenas um bloco de registros, o qual dificulta sua manipulação e consequentemente gerando perda de eficiência.

REFERÊNCIAS:

LINTZMAYER, Carla Negri; MOTA, Guilherme Oliveira. **Análise de algoritmos e estruturas de dados**. Centro de Matemática, Computação e Cognição, 2020.

TANENBAUM, Andrew Stuart; BOS, Herbert. **Sistemas operacionais modernos**. 5. ed. Porto Alegre, RS: Grupo A, 2024. 1 recurso online. ISBN 9788582606186.