

Relatório Trabalho 1, Sistemas Operacionais 2

a) Detalhes das Estruturas utilizadas para armazenamento do Sistema de Arquivos;

O disco rígido consiste em um vetor de booleanos contendo $128 * 1024 * 1024 * 8$ posições para simular uma memória de 128 megabytes, cada arquivo ou diretório ocupará 512 bytes (1 bloco) da memória sendo eles reservados para, nesta ordem:

1 para estado (d para diretório, a para arquivo)

86 para o nome

10 para registrar o endereço do bloco do pai

400 para conteúdo caso seja arquivo, se diretório 200 para armazenar diretório filhos (10 para cada filho) e 200 para arquivos filhos(10 para cada filho)

12 para data e hora

3 para permissão

Este esquema foi discutido em sala de distribuição dos bytes do bloco foi discutido em sala e convencionou-se que seria implementado assim nos trabalhos, mas é necessário ressaltar que este esquema causa fragmentação interna por causa que é necessário apenas 7 bytes para armazenar o endereço de cada filho ou pai ao em vez de 10.

Todas as operações são feitas em cima de uma string com as características do bloco descrito acima que é transformado em um vetor booleano para ser gravado no no vetor de booleanos que representa o disco rígido na posição adequada. Para ler ou manipular o bloco é lido e transformado em um vetor de string.

b) Dificuldade no desenvolvimento do trabalho. Teoria e/ou prática;

O trabalho foi dividido em 2 partes inseparáveis, sendo a primeira as funções e a segunda a implementação do disco rígido, como são inseparáveis na lógica usada na implementação deste trabalho, tudo que foi feito na primeira parte teve de ser jogado fora para fazer a segunda

c) Complexidade das operações das chamadas de sistema implementadas (utilizando notação

O);

ls: $O(n^2)$

mkdir: $O(n^2)$

cd: $O(n^2)$

rmdir: $O(n^2)$

cp: $O(n^2)$

mv: $O(n^2)$

rm: $O(n^2)$

chmod: $O(n^2)$

createfile: $O(n^2)$

cat: $O(n^2)$

batch: $O(n^2)$

dump: $O(n^2)$

info: $O(1)$

Boa parte das funções são $O(n^2)$ pois chamam a função encontraDiretorio que possui esta complexidade e ela não é ultrapassado depois por outra parte a função.

d) Sugestões, considerando as estruturas utilizadas, para a atualização que especifica um sistema de arquivos multiusuário;

Seria necessário reservar um bloco para guardar N os usuários, e em cada bloco de arquivo ou diretório seria necessário armazenar quem criou aquele arquivo e seria necessário verificar a permissão sempre que um usuário fosse acessar um arquivo ou diretório.

e) Sugestões, considerando as estruturas utilizadas, para a atualização que especifica um sistema de arquivos com acesso concorrente;

Seria necessário apenas que o estado fosse alterado imediatamente para um terceiro estado tal como M que representaria modificação sempre que um processo fosse alterar ou criar um novo bloco, e a implementação de um Algoritmo dos Semáforos para resolver um possível e provável problema de concorrência.

f) Comentários para as perguntas:

i. O sistema de arquivos implementado é eficiente?

Sim, ele faz tudo que é proposto de forma direta lendo e escrevendo diretamente do disco rígido simulado.

.

ii. O sistema de arquivos implementado é confiável?

Com exceção da função dump quando usada em sequência que gera um bug onde o segundo dump quando vai gerar a linha responsável por reescrever o arquivo usado como primeiro dump pode ser confundido com o comando do segundo dump, esse problema pode ser resolvido com o dump não podendo escrever comandos que criam arquivos ou apenas criar comandos de arquivos sem seu conteúdo, outro problema é que como um arquivo pode armazenar 400 caracteres o sistema tiver muitos arquivos, ou um que seja, com um conteúdo muito grande ou próximos ao 400 caracteres o dump não poderá armazenar os comandos para recriação do sistema de arquivos atual por falta de espaço, o problema pode ser corrigido com a mesma solução do anterior.

iii. Quais testes foram executados para 'garantir' a confiabilidade do software?

Para cada função foram realizados testes para garantir que cada erro possível foi coberto e avisado ao usuário do sistema quando causado.

iv. Você enxerga diferentes implementações de sistemas de arquivos para diferentes hardwares? Exemplifique.

A implementação feita é boa o suficiente e elimina a fragmentação externa, tendo que tomar cuidado com a fragmentação interna principalmente a citada no item a), fazendo as devidas alterações é possível implementar em diferentes hardware se estes precisarem de um sistemas de arquivos.

