Universidade Estadual Paulista "Júlio de Mesquita Filho" (UNESP)  
Departamento de Estatística, Matemática Aplicada e Computação (DEMAC)  
Curso: Ciência da Computação – Disciplina: Sistemas Operacionais II – Integral  
Prof.: Fabricio Breve – Projeto nº 2 – Data de Entrega: 11/06/2024

Implemente um simulador de um sistema de arquivos, utilizando a linguagem C, que possua as seguintes características:

1. [1,0] Simular um disco com 256 setores de 512 bytes. Considere que os 10 primeiros setores são ocupados pelo setor de boot, sistema de arquivos e o diretório raiz, ficando os demais setores disponíveis para dados do usuário.
2. [1,0] O simulador deve permitir a alocação de arquivos de forma não contígua. Sugere-se usar uma tabela de alocação de blocos e um mapa de bits para gerenciar o espaço livre (a mesma estrutura pode ou não ser aproveitada para ambos os propósitos). Outras formas de alocação mais avançadas também podem ser utilizadas. Cada setor do disco pode estar livre ou ocupado exclusivamente por um único arquivo ou diretório.
3. [0,8] O simulador deve permitir a criação de diretórios e subdiretórios com um comando **criad** até o mínimo de 8 níveis. Cada diretório pode conter vários subdiretórios. Cada diretório/subdiretório deve ocupar um setor do disco. Nomes de subdiretórios precisam ser únicos dentro de um diretório.

**Sintaxe:** criad caminho\nome\_do\_diretorio

1. [0,8] O simulador deve permitir a criação de arquivos dentro dos diretórios e subdiretórios com um comando **criaa**. Cada diretório pode conter diversos arquivos. Ao criar o arquivo, o tamanho do mesmo deve ser especificado e devem ser alocados setores no disco de acordo com seu tamanho. Nomes de arquivos precisam ser únicos dentro de um diretório, e não podem ter o mesmo nome que um subdiretório que esteja no mesmo diretório.

**Sintaxe:** criaa caminho\nome\_do\_arquivo tamanho

1. [0,8] O simulador deve permitir a exclusão de diretórios utilizando um comando **removed**. O simulador não deve permitir a exclusão de diretórios que não estejam vazios.

**Sintaxe:** removed caminho\nome\_do\_diretorio

1. [0,8] O simulador deve permitir a exclusão de arquivos utilizando um comando **removea**.

**Sintaxe:** removea caminho\nome\_do\_arquivo

1. [1,0] O simulador deve oferecer um comando **verd** que exibe os arquivos e subdiretórios do diretório indicado como parâmetro. Na listagem, os diretórios devem conter uma indicação de que são diretórios e os arquivos devem conter tamanho, além de data e hora de criação. No final da listagem deve aparecer o total de arquivos e diretórios, o total do tamanho dos arquivos e a quantidade de espaço livre no disco. Lembre-se que o espaço vazio dos setores utilizados não pode ser utilizado por outros arquivos ou diretórios, portanto não devem entrar na conta do espaço livre. Também deve ser indicado o tamanho total alocado para arquivos e diretórios, incluindo aí o espaço perdido com fragmentação (espaço não usado em setores ocupados).

**Sintaxe:** verd caminho

1. [1,0] O simulador deve oferecer um comando **verset** que exibe graficamente quais são os setores ocupados por determinado arquivo. A representação pode ser feita apenas com caracteres ASCII.

**Sintaxe:** verset caminho\nome\_do\_arquivo

1. [1,0] O simulador deve oferecer um comando **mapa** que mostre graficamente quais são os setores ocupados e quais são os setores livres.
2. [1,0] O simulador deve oferecer um comando **arvore** que mostre graficamente a árvore de diretórios do sistema de arquivos.
3. [0,4] O simulador deve oferecer um comando **ajuda** que quando utilizado mostra quais são os comandos disponíveis no sistema.
4. [0,4] O simulador deve fornecer um comando **sair** que pode ser usado para encerrar o simulador.

Observações:

* O trabalho pode ser feito individualmente, em dupla ou em trio.
* O grupo deverá eleger um integrante para postar o código-fonte no *Google Classroom*, indicando o nome de todos os integrantes. Os demais integrantes devem apenas indicar na tarefa quem são os componentes de seu grupo.
* Todos os membros do grupo deverão ter domínio suficiente para responder qualquer questão sobre o código-fonte ou o comportamento do programa.
* Utilize estruturas para representar os setores, armazenando nelas os metadados do diretório ou arquivo correspondente, além dos ponteiros que forem necessários.
* Para simplificar a alocação de arquivos, você pode considerar que todo o espaço de um setor utilizado por um arquivo será usado por dados do usuário, ou seja, o espaço ocupado por metadados e ponteiros não precisa ser computado.
* Para um exemplo de funcionamento de simulador, veja o programa **faex.exe** que está no *Classroom*. Este é um executável de 16 bits que roda sobre o MS-DOS, compilado no Borland Turbo C. Você pode executá-lo no Linux ou Windows utilizando o emulador DOSBox (<http://www.dosbox.com/>). Instale-o no Ubuntu com “**sudo apt-get install dosbox”**. Execute o faex.exe com “**dosbox faex.exe”**. No simulador, digite **ajuda** para ver os comandos. O Fedora sugere a instalação dos pacotes necessários na primeira execução do DOSBox. Note que este simulador não é exatamente igual ao que está sendo pedido.
* Não é necessário ficar limitado aos requisitos ou ao modelo, use sua criatividade para criar um simulador ainda melhor.