

## Trabalho do 4º bimestre: Implementação de sistema de arquivos

Peso na média: 5

Grupos de no máximo 2 pessoas

Entrega: 20 de novembro

Apresentações: dias 22/12

1. Implementar as funções básicas de um sistema de arquivos para armazenar dados sobre os arquivos e diretórios segundo a teoria do sistema de arquivos baseados em **FAT** e **i-nodes**. Defina uma forma de implementar a tabela FAT e como vai implementar a estrutura de i-nodes. Ao executar o seu programa, a primeira atividade será escolher qual sistema de arquivos será utilizado, qual o tamanho da partição e o tamanho dos blocos (se preferir, estes dados podem ser obtidos através de um arquivo de configuração).
2. No seu sistema de arquivos, considere que cada arquivo tem:
  - nome.ext (nome: limitado até 8 caracteres e ext limitado até 3 caracteres);
  - tamanho;
  - data de criação;
  - primeiro bloco ocupados no disco (para FAT) ou lista de blocos ocupados no disco (para i-nodes).
3. A forma de implementação dos diretórios é livre e deve ter as seguintes características:
  - para FAT: diretórios contêm informações sobre os atributos do arquivo e a indicação de qual o primeiro bloco do arquivo no disco;
  - para i-nodes: o diretório deve conter somente o nome do arquivo e um ponteiro para o i-node (os atributos tamanho, data e blocos ocupados no disco ficam gravados no i-node). Lembre-se que um i-node pode conter também um diretório (neste caso, tamanho não é gravado). Em relação a i-node, projete uma estrutura de dados para armazenar os blocos ocupados pelo arquivo no disco no i-node. Pode ser necessário pensar em uma estrutura de lista de blocos de mais do que 1 nível. Outro detalhe sobre os i-nodes é que se deve prever que os mesmos devem ocupar um espaço no disco (vários i-nodes em um bloco? Pense na melhor solução)

### Considerações para o trabalho:

1. Para testar, podes simular uma partição de um disco com 100 MB de tamanho, dividida em blocos de 512 KB:  
Tamanho total da partição = 100 MB e Tamanho do bloco = 512 KB => Total de blocos é  $100\text{MB}/512\text{KB} = 200$  blocos
2. Para o gerenciamento dos blocos livres e ocupados deve ser utilizado um mapa de bits (como este mapa fica armazenado é decisão do seu projeto).
3. Seu sistema de arquivos (FAT e i-nodes) deve interpretar os seguintes comandos:
  - **mkdir NOME**: Cria um diretório com o nome *NOME* (lembre de armazenar no disco)
  - **cd NOME**: Entra no diretório *NOME*
  - **cd ..**: Volta ao diretório anterior
  - **ls**: Lista o conteúdo do diretório atual, mostrando:
    - Se for arquivo: *A NOME Tamanho do arquivo BLOCOS\_OCUPADOS*
    - Se for diretório: *D NOME*
  - **touch ARQUIVO.EXT TAMANHO CONTEÚDO\_INICIAL** (considere que o tamanho informado é um número inteiro em bytes, o conteúdo é opcional e pode ser usado para incluir dados inicialmente no arquivo)  
Seu sistema de arquivos cria uma entrada para ARQUIVO.EXT no diretório atual, ocupando o número de blocos suficiente para TAMANHO (utilizar o mapa de bits para verificar os blocos livres e atualizar o mesmo ao criar ARQUIVO.EXT)
  - **rm ARQUIVO.EXT**: Excluir essa entrada do diretório atual e liberar esses blocos do disco (atualizando o mapa de bits)
  - **cat ARQUIVO.EXT**: Exibe o conteúdo do arquivo
  - **listmap**: Esse comando mostra o estado atual do mapa de bits, mostrando os blocos ocupados e livres do disco

Observações: *Comece o sistema com o diretório raiz criado*