

Tarefa - Listas em contiguidade física

Nesta tarefa, vamos implementar uma lista de contatos telefônicos. Cada contato terá as seguintes informações: nome (string de 140 caracteres), telefone (string de 20 caracteres), dia, mês e ano de nascimento (inteiros). Para começar, baixe o `kit_lista_final.zip` e extraia seu conteúdo. Se estiver trabalhando com o ambiente de programação online, faça o upload dos arquivos lá.

O kit contém os seguintes arquivos:

- `contato.h`: contém a struct com o tipo contato
- `lista_contatos.h`: interface da lista de contatos
- `io_arquivo.h`: interface do módulo de manipulação de arquivos
- `io_arquivo.c`: implementação do módulo de manipulação de arquivos
- `main.c`: programa principal, usuário da lista
- `contatos_50.txt`: arquivo com dados a serem inseridos na lista

A interface da lista de contatos apresenta várias funções para a implementação e o uso da lista. O `main.c` desse kit inicializa a lista, insere 4 contatos e mostra a lista.

Parte 1) Inicializar, inserir, exibir e destruir

A compilação do projeto falha porque as funções definidas na interface não estão implementadas em lugar algum. É como se o carro estivesse sem motor... Vamos, portanto, fornecer essa funcionalidade básica da lista, implementada em contiguidade.

Complete os atributos da estrutura `lista_contatos` no arquivo `lista_contatos.h`. Crie um arquivo `lista_contatos.c` e implemente as funções `inicializar`, `inserir`, `print_lista` e `destruir`, sem modificar a interface (as funções definidas em `lista_contatos.h`).

Parte 2) Aumentar capacidade conforme necessário

Com a implementação das funções na parte 1, a lista já funciona e 3 contatos aparecem ao executar o programa. Porém, a inserção do quarto contato falha porque a lista foi criada com capacidade inicial de 3 contatos. Vamos remover essa limitação da lista.

Modifique a função `inserir` para que, ao invés de falhar quando não tiver espaço, proceda da seguinte forma:

- Alocar um vetor de itens com capacidade de 150% da atual;
- Atualizar a nova capacidade;
- Copiar os itens do vetor antigo para o novo vetor e usar o novo a partir daí;
- Liberar o vetor antigo;
- Inserir o novo item normalmente.

Fique à vontade para adicionar novas funções se achar necessário, porém, não mude os arquivos `.h`.

Nota: listas em contiguidade física com realocação de tamanho também são chamadas de arranjos dinâmicos, arranjos redimensionáveis ou array list, entre outras denominações.

Parte 3) Preenchendo a lista a partir de um arquivo

Agora, a lista pode conter uma quantidade arbitrária de contatos. Mantenha em seu módulo principal a inicialização da lista com capacidade inicial de 3 contatos. Porém, ao invés de inserir os contatos manualmente, você irá inseri-los a partir dos dados no arquivo `contatos_50.txt`. Cada linha desse arquivo contém um contato, com os dados na seguinte ordem (todos os dados separados por vírgulas; o nome e o telefone podem conter espaços):

`nome,telefone,ano,mes,dia`

Exemplo com as 3 primeiras linhas:

```
Mary Loomis,+55 (85) 902265573,1969,3,10
Susan Smith,+55 (66) 927757359,1918,8,7
Steven Hollingsworth,+55 (34) 961936739,1944,7,12
```

Modifique seu módulo principal para chamar a função `preenche_lista` declarada no arquivo `io_arquivo.h`. Ela preenche a lista telefônica com os contatos dados em um arquivo texto (`contatos_50.txt`, por exemplo). Após o preenchimento, mande exibir a lista para verificar o funcionamento.

Extra: estude a implementação da função `preenche_lista`, no arquivo `io_arquivo.c`.

Parte 4) Remover, buscar e consultar

Implemente as funções `remover`, `buscar` e `consultar`.

Modifique seu programa principal para testar essas funções após o carregamento do arquivo.

O que entregar?

Entregue 3 arquivos:

- lista_contatos.h (com a struct preenchida)
- lista_contatos.c
- main.c

Bom trabalho!