

Estrutura de Dados 1 – Trabalho Final

Você foi contratado para o desenvolvimento de um sistema de cadastro de funcionários para a fábrica Acme®. Para tal, seu software deverá utilizar obrigatoriamente o padrão de estruturas de dados **Lista Ligada/Sequencial Dinâmica**.

A Lista Ligada implementada em seu programa deverá conter um ponteiro especial para o primeiro elemento (ponteiro para ponteiro). Os nós da lista deverão ser compostos por duas partes: a primeira parte armazenará os dados de um funcionário e será formada por uma **struct** com os dados pertencentes a este. A segunda parte armazenará um ponteiro com o endereço para o próximo nó da lista. O último nó da lista deverá apontar para NULL, indicando assim o final da mesma.

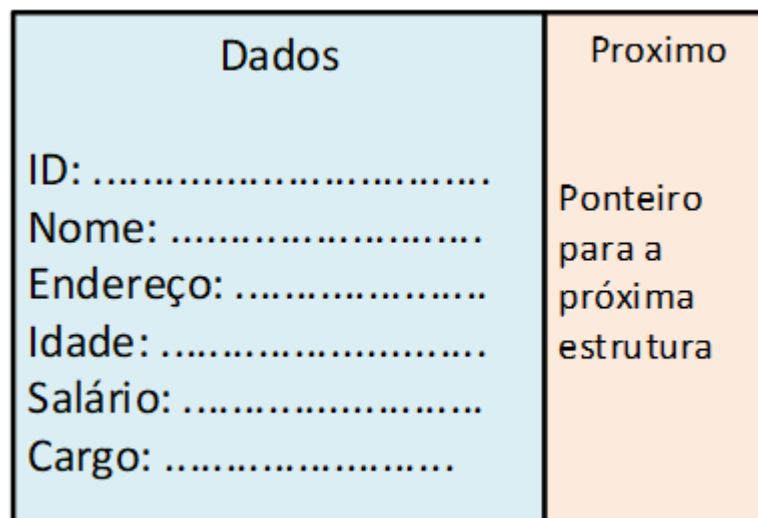


Figura 1 - Representação da estrutura (struct) nó, contendo todos os dados necessários para controle de uma entidade "Funcionário".

O projeto deverá ser montado utilizando o conceito de "Tipos Abstratos de Dados - TAD", e deverá contar com as **operações básicas necessárias** ao bom funcionamento do programa para controle de funcionários. Operações que não sejam necessárias ao funcionamento podem ser removidas. Seu programa deverá ser capaz de salvar e recuperar os dados armazenados entre seções de trabalho diferentes.

Os campos internos da **struct** destinados aos dados do funcionário, devem ter seus **tipos definidos de forma a permitirem o correto armazenamento** de todas as informações pertencentes a um funcionário. A Lista ligada que armazenará os dados dos funcionários da fábrica, deverá ser criada no momento da inicialização de seu programa, e logicamente destruída quando encerrada a aplicação, porém antes salvando todo o seu conteúdo em um arquivo no modo binário que facilitará a recuperação dos dados em uma próxima seção de trabalho.

Salvando Dados

Seu programa deve ser capaz de salvar e recuperar dados entre seções diferentes. Para isso, ele deve ser capaz de executar algumas verificações como por exemplo, verificar se existe o arquivo de dados gerado em uma seção anterior. Se é a primeira vez que se roda o programa, logicamente não existe ainda arquivo de dados. Isso tem que ser detectado para que caso exista o arquivo, o mesmo não seja sobrescrito ocasionando perda de dados. Aqui cabe uma sugestão: comece tentando abrir o arquivo em modo leitura, já prevendo sua existência. Uma vez que exista o arquivo, todo o seu conteúdo deve ser lido e transferido para a Lista Ligada de forma automática, antes de se iniciar o menu de usuário. Em seguida desfaz-se a ligação com o arquivo (fclose), pois o mesmo não é mais necessário.

Caso haja o erro de abertura em modo leitura pela não existência do arquivo, o programa prossegue normalmente então para o menu de usuário. Já que não existem dados ainda, e estes serão incluídos a partir deste momento.

Ao término da seção de trabalho, ainda com todo o conteúdo em memória, é momento então de salvar todo o arquivo sem o risco de perda de dados, sobrescrevendo ou criando o arquivo de dados. É fácil notar que a opção de abrir o arquivo em modo leitura acontece a maioria das vezes, exceto uma, a primeira. Em todas as outras seções com dados já pré-existentes, o arquivo será aberto em modo leitura, lido e depois fechado, sendo aberto novamente ao final do programa, agora em modo gravação binária, para então gravar todo o conteúdo da lista sobrescrevendo-o. Por outro lado, quando o programa rodar uma primeira vez, inicialmente não será necessária a abertura do arquivo imediatamente, somente ao final para salvar todo o conteúdo gerado durante a seção corrente.

Menu de opções

Seu software deve possuir ao menos um menu (podem ser criados quantos forem necessários), e contar com as seguintes funcionalidades:

- Adicionar funcionário;
- Excluir funcionário;
- Editar funcionário, inclusive com opção específica para aumento de salários por percentual de aumento (dissídio salarial informado pelo usuário);
- Buscar funcionário por ID;
- Exibir funcionário ordenados por ID, com todos os seus dados;
- Exibir uma lista de funcionários por faixa salarial. Esta faixa deverá ser informada pelo usuário, com mínimo e máximo. Exemplo: de R\$ 1.000,00 à R\$ 2.000,00.
- Sair da aplicação, salvando antes, todo o conteúdo da lista em arquivo.

Descrição das funcionalidades das opções de menu

- Adicionar funcionário:
 - O funcionário deve ser inserido na lista ordenadamente por ID;
 - Após a inclusão, a aplicação deve retornar ao menu principal.
- Excluir funcionário:
 - Antes de remover o funcionário, a aplicação irá exibir as informações do funcionário em questão, e em seguida perguntar se realmente o usuário deseja excluir o funcionário. Caso a resposta seja positiva, o funcionário deve ser excluído da lista.
 - Após a exclusão do funcionário, a aplicação deve retornar ao menu principal.
- Editar funcionário:
 - A aplicação primeiro localiza o funcionário por ID na lista, e em seguida serão exibidas as suas informações;
 - Neste ponto deve-se determinar se será uma **edição de informações** ou **reajuste salarial**;
 - Em seguida, no caso de edição, a aplicação pede a reinserção/correção dos dados e salva-os na lista. Atenção!! O ID não pode ser alterado, portanto não pode entrar na edição;
 - Caso a opção seja “reajuste”, o mesmo será aplicado;
 - Após a edição das informações do funcionário ou reajuste salarial, a aplicação deve voltar ao menu principal.
- Buscar funcionários:
 - A aplicação primeiro localiza o funcionário por ID na lista, e em seguida serão exibidas as informações do funcionário;
 - Após exibir as informações do funcionário, a aplicação deve voltar ao menu.
- Exibir funcionários ordenado por ID:
 - A aplicação irá exibir o ID, nome, cargo e salário dos funcionários;
 - Após exibir as informações dos funcionários, a aplicação deve retornar ao menu principal.
- Exibir funcionários por faixa salarial:
 - A aplicação irá exibir o ID, nome, cargo e salário dos funcionários, selecionados entre um intervalo mínimo e máximo de salários, que deverá ser passado pelo usuário;
 - Após exibir as informações dos funcionários, a aplicação deve retornar ao menu principal.
- Sair da aplicação;

- A aplicação será encerrada, liberando toda a memória utilizada para o processamento de seus dados.

Regras:

- Implementar o código na Linguagem C;
- Utilizar exclusivamente estruturas de dados **alocadas dinamicamente**, sendo portanto, obrigatório o conceito de lista ligada ou dinamicamente encadeada;
- No programa principal, o seu arquivo “*main()*”, você deverá somente “chamar as funções” do programa. Nenhum dado de funcionário deve ser preenchido, impresso ou editado no “*main()*”, **para isso modularize seu programa, crie funções específicas!**
- Clareza, indentação e comentários que sejam pertinentes são bem vindos;
- **Trabalhos copiados (e o Trabalho que serviu de fonte, inclusive) terão nota 0. Todos os códigos serão submetidos ao detetor de plágio, que além de verificar possíveis cópias em todos os projetos enviados e de outras turmas (semestres anteriores que estão arquivados), também verificará códigos oriundos da internet;**
- O trabalho será realizado no máximo em duplas;
- Use o bom senso, **não complique!** Implemente uma funcionalidade por vez, facilitando assim a correção de possíveis erros durante seu desenvolvimento.

O Trabalho final deverá ser entregue com:

- Código fonte (bem indentado e comentado);
- Arquivo executável, e todos os arquivos gerados pelo compilador (*.o*, *.depend*, *.layout*, *.cbp*);
- Relatório:
 - Explicar as decisões de implementação tomadas, uma visão geral do funcionamento do programa, comentários sobre os testes executados, etc.
 - Resultado dos testes, quais foram os problemas, como foram resolvidos, etc.

Todo o material deve ser entregue na plataforma Moodle, compactado (zipado) em data especificada na plataforma.

Não há a possibilidade de alteração da data de entrega.