ESTGOH
Escola Superior de Tecnologia e Gestão de Oliveira do Hospital
Instituto Politécnico de Coimbra

Licenciatura Tecnologias e Sistemas de Informação

Programação II

2019/20

Relatório

Ficha 5

Jorge Miguel dos Santos Martins Nº a2019100813

01/06/2020

Índice

1 Introdução

Esta foi uma ficha realizada no sentido de compreensão no uso de estruturas com listas ligadas.

No decorrer de toda esta ficha poderemos dizer que o exercício mais complexo é o exercício 2, não pela sua quantidade de questões, mas sim pela navegação em águas até então desconhecidas que são as listas ligadas em C, obviamente foi necessário fazer alguma pesquisa para compreensão da logica, mas principalmente para o uso correto de uma sintaxe da linguagem de forma correta e funcional.

Este relatório será apenas um guia ao código, pois no decorrer da escrita de código foram deixados comentários a explicar as funções.

2 Trabalho realizado

2.1 Pergunta 1

Nesta questão é pedido para que através do exercício 4 da ficha 4, usemos um ponteiro para uma memoria alocada dinamicamente vez de um vetor de estruturas com memoria estática. Usando o já conhecido procedimento através da função "malloc" é feito o alocamento do espaço requerido. Inicialmente tentei fazer com que fosse sendo aumentado o espaço a ocupar na memoria enquanto iam sendo inseridos os funcionários, mas seguidamente através de conselho da Docente, mudei para um uso estático de quantos funcionários o utilizador irá inserir.
A grande mudança neste exercício é o facto de deixar de existir um vetor estático de estruturas para passar a usar um ponteiro ao qual aloco uma memoria estática enquanto todo o acesso é feito através de ponteiro em vez de índice.

2.2 Pergunta 2

Este é o exercício chave nesta ficha, ao qual é através dos subprogramas aqui desenvolvidos que adaptados são usados nas outras questões. Iniciando este é um subprograma com mais comentários pois nos outros devido apenas usar estes subprogramas adaptados não achei necessidade de comentar algo já desenvolvido anteriormente.
Sendo este o exercício que mais tempo foi gasto na sua realização, sendo que o grau de dificuldade é mais elevado devido ao facto de inicialmente não compreender a sintaxe usada no manuseamento das listas ligadas, não enchendo o Docente com perguntas através da minha pesquisa individual encontrei vários exemplos de como usar as listas ligadas.
Este é um exercício completo no que toca à manipulação de listas ligadas.
Apenas de realçar o facto de na troca de valores ter usado a troca de ponteiros quando não era necessário, tendo me levado a alguns cálculos gráficos de como iria realizar tais tarefas, mas seguidamente uso um outro subprograma "troca" que realiza essa troca de valores e não de ponteiros de entre listas.

No subprograma de inserir uma "n" quantidade de valores que os utilizadores digam, decidi que através deste subprograma que é criada a lista 2, esta que depois é usada para ser copiada no subprograma pedido pela professora de copiar uma lista para outra.
Tanto no subprograma de escrever, e copiar é pedido ao utilizador qual das listas quer escolher.

2.3 Pergunta 3

Adicionando uma nova opção no acesso às listas ligadas, neste exercício é nos implementado o uso do ponteiro "tail" ou cauda ao qual na medida que são inseridos valores (inseridas listas com valores) este ponteiro fica sempre apontar para a lista que se encontra no final das listas ligadas.
Neste exercício implementei uma estrutura de ponteiros para estrutura, dando o nome de "Head" para o ponteiro que irá apontar para o início e "tail" para o que vai apontar para o final.
Neste exercício adaptei o subprograma "junta valor" para o qual caso a lista esteja vazia uso uma condição para que seja usado o código usado no subprograma de inserir no inicio de uma lista no exercício anterior.
Também é possível observar que apenas nos subprogramas de "junta_valor", "removeUltimo", "removerOcorrencia", "mostrarUtimo", achei necessário o envio do ponteiro "tail" pois em alguns dos subprogramas facilita com o seu uso e em outros o caso de termos de mudar esse ponteiro (inserindo novo valor no final ou removendo).

2.4 Pergunta 4

Sendo uma pilha aquele que vai empilhando e o seu ultimo a entrar é o primeiro a sair, então na teórica indica que o ultimo está no topo dessa pilha (o ultimo a ser carregado para essa pilha fica "em cima" de todos os outros), então para este programa adaptei os subprogramas "junta_no_ini_lista", "removerprimeiro", "mostrarPrimeiro", para o que é pedido neste exercício. De referenciar que o subprograma "junta_no_ini_lista" insere a lista sempre no inicio do encadeamento assim fazendo o empilhamento das mesmas.

2.5 Pergunta 5

Neste exercício é pedido uma fila sendo esta o oposto do uso de pilha, então para este exercício usei os subprogramas “junta_no_ini_lista”, “junta_valor”, “removerprimeiro”, “mostrarPrimeiro”. Aqui usei o “junta_no_ini_lista” para que apenas inicialmente caso a lista esteja vazia insira a primeira lista, seguidamente após condição de lista estar com elementos, uso o subprograma “junta_valor” para que seja inserida a lista sempre no final do encadeamento.

2.6 Pergunta 6

Este é o exercício em que nos é pedido para adaptar o exercício 4 da ficha anterior, mas já com listas ligadas e não apenas com ponteiros como feito no exercício 1 desta ficha 5. Onde existiram maiores adaptações foi na inserção dos dados dos funcionários, ao qual novamente adaptei o que já tinha sido usado no exercício 2 da ficha 5. Uso uma adaptação dos subprogramas “junta_valor” e “junta_no_ini_lista” ao qual a verificação de escolha entre se a lista está vazia para o “junta_no_ini_lista” ou se já tem elementos para o “junta_valor” é feita no main a quando o utilizador escolhe a opção de inserir dados.

Nos programas de listagem e listagem com filtro (caso dos 500 euros) aqui apenas foi adaptado o subprograma a nova realidade de uso de listas ligadas, que no decorrer desta lista são observados várias vezes, na realização de um ciclo em que enquanto o ponteiro não indicar “NULL” após uma escrita o ponteiro iguala ao elemento “prox” da lista que aponta para a seguinte lista encadeada.

De observar que realizei um subprograma de troca de apenas de valores para o uso de ordem Alfabética, utilizei o algoritmo Bubble Sort sendo este o qual estou mais familiarizado.

Visto a logica de a construção deste exercício já estar arquitetada anteriormente decidi não mudar e apenas adaptar com o uso de listas ligadas.

3 Conclusão

Na realização desta ficha de trabalho foram obtidos conhecimentos importantes no manuseamento de listas ligadas.

Este relatório foi realizado apenas como guia de contextualização do código estando em sintonia com o que foi dado em forma de comentários para uma melhor perceção das decisões e logicas por detrás do que foi escrito. É de avisar que existem muito mais comentários no exercício dois desta ficha pois os outros exercícios partem em parte de adaptações dos subprogramas ali utilizados ao qual achei desnecessário estar numa repetição e enfadamento de informação o que poderia levar a confusão.

Concluindo que o exercício 2 desta ficha é o exercício chave para a compreensão e uso das listas ligadas e que por isso também não perdendo tempo na mudança de logica nas adaptações dos exercícios, apenas realizei as mudanças para o uso de listas ligadas.

Sei que existem certos pontos a limar no que toca na segurança da implementação de dados, quando o utilizador tentar inserir algo que não um inteiro num local devido para isso, mas devido ao tempo apertado foquei na realização do núcleo do exercício deixando um pouco a sua redundância sem muito foco.

4 Referências

Listar consultas usadas

- Luis Damas Liguagem C
- Matéria teórica Universidade de Coimbra
- Matéria teórica UBI

5 Anexos

Exer1.c

Exer2.c

Exer3.c

Exer4.c

Exer5.c

Exer6.c