FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

PROJETO

Princípios de Programação Procedimental

LICENCIATURA EM ENGENHARIA INFORMÁTICA

2021/2022

Beatriz Costa Pereira Monteiro - 2021234863

Luís Gonçalo Teixeira Pereira – 2022230244

Índice

Introdução

No âmbito da unidade curricular de Princípios de Programação Procedimental foi-nos proposto a realização de um trabalho com o objetivo de desenvolver o código de uma aplicação cuja finalidade é manter atualizada toda a informação sobre os alunos, como por exemplo, data de nascimento, ano, turma etc.

O relatório tem como função demonstrar sucintamente o código e os erros obtidos durante a elaboração do mesmo bem como possíveis melhorias.

Manual do Programador

Este programa tenciona elaborar uma “aplicação” que funciona como um sistema de uma escola, isto é, que guarda dados de cada aluno agregado de várias funcionalidades que são necessárias a este serviço. O programa em questão serve tanto para o aluno como para o administrador da escola, podendo este alterar dados de alunos, carregar contas, etc.

O programa contém diversas funções e estruturas que estão contidos nos ficheiros ".h”, ".c" e o ficheiro de texto ".txt".

Assim que o programa é iniciado deparamo-nos com um menu que contém as seguintes opções:
- Adicionar Aluno;
- Eliminar um aluno existente;
- Consultas de saldo;
- Efetuar uma compra;
- Informação de um aluno;
- Transferências;
- Carregar a conta;
- Consultar histórico;
- Entre outras;

De seguida serão apresentadas as funções que se mostraram mais trabalhosas, bem como a solução mais prática encontrada,

<u>Adicionar aluno</u>

É pedido ao utilizador para escrever os dados de um aluno: nome, turma, ano, número de estudante e data de nascimento. Foram inseridos verificadores para avaliar se a data inserida é valida ou não e de igual modo para verificar se o aluno que pretendemos adicionar já existe ou não. Há medida que o utilizador escreve os dados, estes vão ser inseridos numa estrutura, que são inseridos numa lista ligada.

É necessário ter todos os dados do aluno numa lista ligada pois esta irá ser necessária para todas as outras opções do código, tais como eliminar um aluno.

De notar que caso a data inserida seja inválida ou o aluno já exista, o utilizador irá ser aviso desse erro e todos os dados inseridos pelo utilizador não irão ser inseridos na lista.

Por fim, será adicionado ao ficheiro todos os dados dos alunos para estes serem guardados e posteriormente usados em outras instâncias.

Eliminar um aluno existente

Para esta ação funcionar foi necessário encontrar a posição na lista ligada do número de estudante que se pretende eliminar. De seguida ó ponteiro que aponta para o numero de estudante atrás do que queremos eliminar é apontado para o próximo desse mesmo numero de estudante, passando a frente o numero que queremos eliminar.
Com a função free() toda a informação desse aluno será eliminada da estrutura.

Consultar histórico

Para o histórico foi necessária a criação de um ficheiro que guarda todas as ações efetuadas por todos os alunos. Sempre que é efetuada uma ação, tal como, uma compra, uma transferência e o carregamento da conta, os dados serão enviados para o ficheiro para que este seja consultado sempre que o utilizador quiser.

Efetuar compras

O utilizador ao escolher a opção 4 no menu principal, irá seguir-se ate outro menu (menu compra()) onde este terá escolher que tipo de compra quer efetuar.
É lida a opção de compra pretendida através de um scanf e irá outra vez aparecer no terminal um menu com todos os produtos disponíveis consoante o tipo de compra escolhido.
Finalmente, após escolher o produto desejado, é verificado se o aluno tem ou não saldo suficiente. Caso tenha, irá ser descontado ao saldo o valor da compra. Caso contrário, o aluno irá ser avisado da insuficiência de saldo e será "mandado" de volta para o primeiro menu.

Alterar dados do utilizador devido a possíveis erros

O utilizador ao escolher a opção no menu principal, irá seguir-se ate outro menu onde este terá escolher qual o dado que pretende alterar.
Para esta função foi necessária a procura do número de estudante na lista ligada. De seguida, é substituído o valor/string que queremos eliminar pela nova pedida ao utilizador.

Falhas e possíveis soluções

Após varias utilizações do código, reparamos que ao não receber números inteiros como strings, ou seja, ao usar “scanf(“%d”, &x)”, se o utilizador escrever no terminal um caracter diferente de um número, o programa iria bloquear.

Também e não menos importante, caso o utilizador por alguma razão escrevesse um número elevado de caracteres, o programa também iria bloquear.

Como solução, foi pensado em criar uma string para receber cada inteiro, scanf(“%3s”, &x), essa string iria depois ser convertida em inteiro usando a função atoi().

Com esta solução todos os nossos problemas iriam ser resolvidos, pois através de condicionadores, o programa não iria aceitar caracteres diferentes de números (o utilizador iria ser alertado da ocorrência) e também não iria bloquear caso o utilizador escrevesse um número elevado de caracteres pois no scanf efetuado o “3” antes do s faz com que só leia os 3 primeiros caracteres.

Conclusão

Este trabalho serviu ajudou-nos a compreender melhor toda a funcionalidade de C, uma vez que nos permitiu detetar e consequentemente corrigir os erros que foram aparecendo durante a elaboração do código. Durante a elaboração do programa deparamo-nos com dificuldades na implementação de várias funções, no entanto devido às diversas tentativas para obter o correto funcionamento do mesmo, essas dificuldades foram ultrapassadas.

.