Manual do Programador

Mini-projeto PPP

Pedro Miguel Oliveira

4 de junho de 2022

Foi desenvolvida uma aplicação para gerir as contas do bar dos alunos de uma escola secundária. A aplicação mantém informação atualizada sobre todos os alunos (nome, data de nascimento, ano, turma, número, saldo) e as despesas por eles efetuadas ao longo de um ano civil (valor, descrição, data) em listas ligadas. A aplicação foi completada com produtos e transações sobre esses. A aplicação permite realizar, de forma interativa, as seguintes operações:

1. Introduzir dados de um novo aluno.
2. Eliminar um aluno existente.
3. Listar todos os alunos.
4. Listar os alunos com saldo abaixo de um determinado valor.
5. Apresentar toda a informação de um determinado aluno.
6. Efetuar uma despesa por um determinado aluno.
7. Carregar a conta de um aluno com um valor.

O item 3 e 4 não foram completados na íntegra, pois não ordenam pelas ordens requiridas no enunciado.

- Toda a informação é armazenada em ficheiros de texto, os quais são carregados quando o programa é iniciado e atualizados sempre que necessário de modo que não haja perda de informação.
- A aplicação tem uma proteção contra o utilizador bastante sólida.

Conteúdo

1 Introdução

Este projeto consiste na implementação de uma aplicação para gerir contas de alunos num bar de uma escola secundária. A aplicação permite criar alunos para que estes posteriormente possam fazer compras e em seguida, possam ser feitas compras. Da mesma forma que podem ser criados alunos estes também podem ser apagados.

O bar tem:

- um nome
- uma morada
- um dia atual (hoje)

2 Organização geral da aplicação

2.1 lib.h

Este é o ficheiro *header* do projeto, contém os *includes*, constantes necessárias, e as várias estruturas e o cabeçalho de todas as funções.

2.1.1 MAX_SIZE_LINE

Tamanho máximo de uma linha de um ficheiro de texto.

2.1.2 MAX_SIZE_INPUT

Tamanho máximo para input do utilizador.

2.1.3 _FILENAME

Nome ou caminho para os ficheiros de texto com os dados da aplicação:

- **INIT_FILENAME** - com as informações gerais do bar
- **STUDENTS_FILENAME** - com as informações dos estudantes
- **PRODUCTS_FILENAME** - com os produtos para venda
- **BILLS_FILENAME** - com as vendas.

2.1.4 YEAR

É definido o ano mínimo **MIN_YEAR** e ano máximo **MAX_YEAR** a aceitar pela aplicação.

2.2 main.c

Este é o ficheiro que contém todas as funções do projeto.

3 Estrutura da aplicação

3.1 date

Para utilização de datas uniformizadas.

3.2 bar

O bar contém um nome, uma morada e um dia atual. É possível alterar o dia atual através do menu principal da aplicação.

3.3 student

Cada estudante tem um número, um nome, uma data de aniversário, um ano e turma, e o saldo.

3.3.1 student_list

Esta é uma estrutura auxiliar a fazer listas de estudantes, por exemplo na procura de estudantes abaixo de x valor.

3.4 product

Decidi implementar uma estrutura para produtos, para completar o meu projeto. Cada produto contém um código de barras, um nome, e um preço. O stock não é considerado neste trabalho.

3.5 bill

É sob esta estrutura que são registadas as vendas decorridas na aplicação.

3.5.1 bill_list

Esta é uma estrutura auxiliar a fazer listas de vendas, por exemplo aquelas que pertencem a um certo utilizador.

3.5.2 bill_descriptive

Esta é a estrutura que guarda as diferentes quantidades de produtos que são vendidas, cada linha do talão, em determinada venda. Cesto de compras.

3.5.3 bill_descriptive_list

Esta é uma estrutura auxiliar a fazer o cesto.

4 Ficheiros de texto

4.1 Nome/caminho iniciais

Na pasta do executável, incialmente definidos para: 'init.txt', 'students.txt', 'products.txt' e 'bills.txt' respetivamente.

4.2 Formatação

4.2.1 init.txt

Informações gerais do bar.

```
BAR DE PPP          <nome>
PINHAL DE MARROCOS  <rua>
3030 COIMBRA        <codigo-postal>
3-6-2022            <hoje>
```

4.2.2 product.txt

Informações dos produtos para venda.

```
<codigo-barras>;<nome>;<preco>
10000;CAFE;0.80
```

4.2.3 students.txt

Informações dos estudantes.

```
<numero>;<nome>;<data-nascimento>;<ano>;<turma>;<saldo>
2012101;PEDRO MIGUEL OLIVEIRA;4-7-1998;10;CT4;10.00
```

4.2.4 bills.txt

Informações de carregamento ou vendas dos estudantes.

```
# carregamento - tipo 1
1;<data>;<numero-estudante>;<valor>
1;5-5-2022;2012101;10.00

# venda - tipo 0
0;<data>;<numero-estudante>;<[<qtd>,<codigobarras>:<...>]>;<total>
# um produto
0;3-6-2022;2012101;1,10000:0.80
# mutiplos produtos
0;3-6-2022;2016789;2,10002:2,10000:3.80
```

5 Funções principais

5.1 main()

Nesta função são inicializadas as filas de alunos, produtos e despesas. Faz-se o carregamento da informação dos ficheiros de texto para as filas previamente inicializadas. Esta função contém também o **menu principal** da aplicação. O menu principal permite:

1. adicionar um aluno
2. entrar na ficha de aluno
3. listar todos os alunos
4. listar alunos abaixo de x valor
5. carregar conta de aluno
6. efectuar venda
7. alterar data da aplicação

A função de apagar utilizador está dentro da ficha do aluno, no segundo menu da aplicação.

5.2 bar_init(...), bar_save_..(...)

Funções responsáveis pelo carregamento e escrita dos ficheiros de texto.

5.3 cli_asks_for_student_number(...)

Pede ao utilizador um número de aluno válido. Esta e qualquer outra instrução de entrada pode ser terminada com 'c!' com excepção das confirmações de compra ou remoção de elementos estas só podem ser concluidas com s ou n.

5.4 cli_show_user_bills(...)

Mostra carregamentos e compras de forma sumariada e permite a consulta detalhada dessas transações.

5.5 cli_student(...)

Mostra perfil do aluno, e dá possibilidade de ver as transações do mesmo: quer sejam carregamentos ou compras e descrição das mesmas. É neste menu, acessível na opção 2 do menu principal, que se apagam alunos.

5.6 get_students_below(...)

Cria uma lista não ordenada com os estudantes abaixo de um certo valor.

5.7 cli_asks_for_bill_descriptive(...)

Pede ao utilizador uma nova entrada para adicionar ao cesto. Deve ser introduzido no seguinte formato:

```
<qtd>x<codigo-barras>
2x10000
```

5.8 cli_bill(...)

Pede número de estudante e inicia uma venda ao utilizador. Este pode listar produtos, ver o cesto, adicionar ao carrinho, remover do carrinho, finalizar ou cancelar a venda. Estas últimas requerem uma confirmação adicional ao utilizador para efetivar a transação.

6 Funções complementares

6.1 Datas

São definidas funções para converter data para string e vice-versa, ou verificar se formatação é uma data válida.

6.2 Escolha de elementos

São definidas em funções a procura elementos nos diferentes tipos de listas.

7 Funcionamento

No arranque, a aplicação verifica a existênia dos ficheiros de texto utilizados. Se estes não exisitirem, a aplicação não arranca. Não foram testados cenários de manipulações por parte do utilizador nesses ficheiros.

Foi testado o bom funcionamento através do IDE Clion.

8 Execução da aplicação

No mesmo diretório deste manual do programador, segue a pasta src com o código fonte da aplicação, bem como os ficheiros de texto necessários à aplicação: init.txt, products.txt, bills.txt, students.txt. 4 O executável procura no seu diretório a existência destes ficheiros. É incluindo o ficheiro 'CMakeLists.txt' com este relatório.

9 Conclusão

Cumpro todos os requisitos do enunciado, construí alguma documentação e faço uso dos bons princípios de programação procedimental.

A aplicação tem uma proteção contra o utilizador bastante sólida.

Para terminar, ao serem removidos estudantes, as transações deste ficarão com estudantes inexistentes.

■