PRG22105 — Programação de Computadores II

Projeto 4 de junho de 2025

Professor: João Cláudio Elsen Barcellos, joao.barcellos@ifsc.edu.br
Estudante:

Exercício 1: Máquina de refrigerantes

Objetivo e visão geral: Neste projeto propõe-se o desenvolvimento de uma aplicação, executada via terminal, para simular uma máquina de refrigerantes. Esta aplicação fará uso de uma máquina de estados, estruturas de dados (e.g., pilhas, filas, listas simplesmente e duplamente encadeadas) e manipulação de arquivos.

A máquina de refrigerantes disponibilizará, inicialmente, dois refrigerantes: **Pureza** e **Guaraná Jesus**, ambos custando R$ 1.5. O usuário pode inserir moedas de R$ 1.0, R$ 0.5 ou R$ 0.25. O sistema deve rastrear apenas o **valor total** acumulado, sem necessidade de gerenciar, individualmente, cada moeda. Moedas inválidas são devolvidas automaticamente e imediatamente. O usuário pode usar a funcionalidade de *undo* (descrita abaixo) para cancelar a inserção de moedas ou reverter outras ações. O sistema calculará e informará o valor do troco (simplesmente como $Troco = ValorInserido$ - $PreçoProduto$) automaticamente quando o valor inserido for suficiente e um produto for selecionado.

Requisitos mínimos:

1. **Máquina de Estados:**
   - Implementação via `switch-case`;
   - Estados obrigatórios: menu inicial, seleção de produtos, pagamento e configuração pelo dono da máquina de refrigerantes (e.g., para adicionar propagandas e/ou mudar os produtos que estão disponíveis).
2. ***User interface* (UI):**
   - Entrada e saída de dados via terminal;
   - Atualização "dinâmica" da interface (e.g., por meio do comando *clean*).
3. **Gerenciamento de propagandas:**
   - Propagandas são gerenciadas por meio de fila(s);
   - Apresentadas ciclicamente no terminal, a cada 1 min, por exemplo.
4. **Estruturas de Dados:**
   - **Pilhas** para desfazer (*undo*) últimas ações, incluindo...;
     - Retornar ao estado anterior (e.g., voltar do estado de pagamento para o estado de seleção de produtos);
     - Desfazer a inserção da última moeda (funcionando como uma tecla de devolução);
   - **Filas** para gerenciar as propagandas;

- **Listas Encadeadas**...;
  - **Lista duplamente encadeada** para configuração de produtos: cada nó representará um produto com informações como ID, nome, preço e estoque. Esta estrutura permite ao administrador adicionar, remover e atualizar produtos facilmente;
  - **Lista simplesmente encadeada** para *logs* de vendas: cada nó conterá informações sobre uma venda (timestamp, produto, valor pago, troco). Os logs serão carregados ao iniciar o sistema e atualizados durante a operação, mantendo o histórico completo. Considere o uso de arquivos binários para armazenamento eficiente destes dados.

5. **Menu de Configuração:**
   - Arquivo de configuração protegido por senha;
   - Atualização de nome, preço e detalhes dos produtos acessível apenas pelo dono.
6. **Persistência e *Logs*:**
   - *Logs* das vendas armazenados em arquivo;
   - Recuperação desses arquivos toda vez que esta máquina de refrigerantes é religada.
7. **Documentação e versionamento:**
   - Uso obrigatório do GitHub;
   - Recomenda-se uso de `Doxygen` para documentação do código.
8. **Funcionalidade adicional:**
   - Deve-se implementar alguma funcionalidade extra.

Sugestão de entregas parciais:

Entrega 1: Levantamento de requisitos, diagramas da FSM e especificação das estruturas;

Entrega 2: Implementação da FSM básica, estruturas principais (pilha, fila, lista encadeada), arquivo de configuração inicial e *logs*;

Entrega 3: Integração completa, configuração segura, recuperação de estado via log, documentação e demonstração.

Critérios de avaliação:

- Eficiência das estruturas implementadas;
- Clareza e organização do código;
- Uso correto de máquinas de estados e manipulação de arquivos;
- Qualidade da documentação;
- Uso efetivo do GitHub.

Observações:

- Implementação obrigatória em C;
- O sistema deve rodar integralmente no terminal;
- Estruturas devem ser implementadas sem uso de bibliotecas externas (apenas `stdio.h`, `stdlib.h`, etc.);
- É esperado um comportamento coerente e estável em todas as operações;
- Este trabalho foi baseado em [1], o qual pode ser usado como referêcia para ter ideias;
- Pode-se avaliar também a organização do firmware apresentada em [2] ou em qualquer repositório do SpaceLab.

Referências

[1] Eduardo Augusto Bezerra. *Projeto Final – Vending Machine*. Available at: https://gse.ufsc.br/bezerra/?page_id=2175. Accessed: 2025. 2018.

[2] *eps2 (41a6761)*. Github. Web-base repository. Available at: https://github.com/spacelab-ufsc/eps2. Accessed on: 19 February 2025. SpaceLab, 2023.