

Projeto de Algoritmos e Estrutura de Dados I - 2º Semestre 2018

Gabriel Van Loon Bodê da Costa Dourado Fuentes Rojas

Prof. Robson Cordeiro

1. RESUMO DO PROJETO

O objetivo deste projeto é modelar as operações de um robô de uma fábrica de brinquedos, cujo afazeres são: armazenar brinquedos de uma esteira e organizá-los em caixas; inserir as caixas de brinquedos em uma esteira de caixas.

Para isso, foi criada uma biblioteca “fabrica.h” que encapsula o protótipo de todas as sub rotinas necessárias para modelar esse cenário. Além disso, a implementação dessas sub rotinas foi dividida de duas maneiras distintas, uma com abordagem sequencial estática e outra encadeada dinâmica.

Também foi criada um arquivo “main.c” que pode ser utilizado em ambas as abordagens, uma vez que as sub rotinas e seus parâmetros não se alteram.

2. ESTRUTURAS DE DADOS UTILIZADAS

Para a execução do projeto, foi necessária a criação dos seguintes tipos de dados:

- ❑ struct Brinquedo: tipo de dado que modela o brinquedo de uma fábrica, e tem as propriedades *Número de Série* e *Nome do Modelo*. Foi criado para facilitar a manipulação desses elementos ao longo do código.
- ❑ struct ListaCirc: TAD Lista Circular de Brinquedos, que modela a Lista de Brinquedos da fábrica. Na implementação estática sequencial aloca um vetor de tamanho fixo de brinquedos, enquanto na encadeada dinâmica é utilizada em conjunto com uma estrutura auxiliar struct NoL que representa um Nó/Bloco de Lista.
- ❑ struct Pilha: TAD Pilha de Brinquedos, que modela a Caixa de Brinquedos da fábrica. Na implementação estática sequencial aloca um valor fixado *P* definido pelo usuário na “main.c”, enquanto na versão encadeada dinâmica é utilizada em conjunto com uma estrutura auxiliar struct NoP que representa um Nó/Bloco de Pilha.
- ❑ struct Fila: TAD Fila de Pilhas, que modela a Esteira de Caixas da fábrica. Na implementação estática sequencial aloca um vetor de tamanho fixo de *Pilhas*,

enquanto na encadeada dinâmica é utilizada em conjunto com uma estrutura auxiliar *struct NoF* que representa um Nó/Bloco de Fila.

3. ABORDAGENS UTILIZADAS NAS IMPLEMENTAÇÕES

Geral:

- ❑ Manteve-se igual os nomes e parâmetros de chamada das subrotinas em ambas as implementações;
- ❑ Prezou-se a legibilidade das funções para facilitar reutilização ou manutenção do código;
- ❑ A Pilha, por ter um tamanho máximo pré-fixado, se demonstrou mais enxuta e adequada de ser implementada de modo sequencial estático;
- ❑ A Fila e a Lista, por serem manipuladas com mais frequência (e.g. no caso da Lista, que realiza remoções no meio do vetor), acabam por serem mais eficientes e fáceis de implementar no modo encadeado dinâmico;
- ❑ Aspectos mais específicos, como as etapas que o programa executa e como utilizar o programa via terminal estão documentados no próprio código da “main.c”.

Sequencial Estática:

- ❑ Como haveria a necessidade de implementar uma Fila de Pilhas, todos os vetores utilizados pelas estruturas foram alocados na Heap para diminuir o custo de manipular as variáveis das TAD's;
- ❑ Foi inserido na “fabrica.h” um macro que permite alterar o tamanho fixo a ser alocado na manipulação da Lista e da Fila;

Encadeada Dinâmica:

- ❑ Como citado anteriormente, foram criadas tipos auxiliares para representarem os blocos encadeados de cada estrutura.
- ❑ Todas as estruturas foram implementadas com blocos de encadeamento unidirecional.