

Trabalho 1 de Estrutura de dados: Simulação de Porto

Trabalho realizado por Gustavo Teixeira Magalhães. Esse documento se refere à como usar o makefile e compilar o arquivo de forma mais eficiente.

Tutorial Make:

Para compilar e rodar o programa com o makefile, entre na linha de comando (Bash do linux ou o interpretador do Cygwin) e digite os seguintes comandos:

- `make` para apenas compilar o trabalho.
- `make run` para compilar e rodar o trabalho.
- `make clean` para limpar arquivos previamente compilados.
- `make cleanmake` para recompilar o arquivo, removendo a compilação anterior.
- `make cleanrun` para recompilar, limpar e rodar a nova compilação.

Estrutura dos Arquivos:

Os arquivos código fonte estão na pasta `src`, o arquivo compilado fica na pasta `obj` e as documentações (fora este README) ficam na pasta `DOCS`.

Sobre a implementação

O trabalho visa a implementação de um simulador portuário por meio de estruturas clássicas de dados (pilha e fila), visando o entendimento e prática no uso dessas estruturas.

Estruturas Escolhidas

Para controlar as operações do porto, resolvi utilizar a implementação dinâmica das estruturas pilha e fila, devido à facilidade da implementação, eficiência de memória e limpeza de código. As estruturas são, especificamente:

- Uma fila de atracamento (que contém os navios);
- Quatro pilhas de containers por navio;
- Cinco pilhas de containeres pra travessa (que contém os containers descarregados pelos navios);

As imagens abaixo esclarecem melhor as estruturas:

□

Anexo1: Estruturas da pilha e fila

□

Anexo2: Estrutura da travessa e suas pilhas de container

Contadores de tempo

O trabalho é executado por meio de uma variável de contagem de tempo chamada `ut`. Essa variável é responsável por iterar o programa, bem como computar cálculos de desempenho. Existem dois tipos de `ut`:

- `ut geral`: Serve pra iterar o programa e mostrar quantas iterações já ocorreram.

- ut do navio: Serve pra contar quanto tempo o navio fica na fila de espera.

Além desses ut's, existe também o controle dos carros de transporte, que é traduzido em uma variável (carro1, carro2, etc). Essa variável pode ter 3 valores possíveis: 0, 1 ou 2.

- 0: Significa que o carro está pronto para armazenar a pilha da travessa.
- 1: Significa que o carro acabou de ficar ocupado e está indo armazenar uma pilha.
- 2: Significa que na próxima iteração, o carro vai ficar livre.

A simulação

O usuário entra no programa e lhe é perguntado quantos dias do porto ele quer simular, ou seja, quantas iterações da simulação ele quer ver. Após a decisão do usuário, o programa procede com a execução e segue um algoritmo padrão, reportando um relatório no final de cada dia. Esses processos são, segundo a ordem de execução:

Atracar

No começo de cada dia, de 0 a 3 navios são enfileirados em cada fila de atracamento. O numero que é enfileirado é escolhido aleatoriamente.

Ao enfileirar um navio, ele é criado, recebe um id (baseado na quantidade de navios na pilha), suas pilhas de containers são empilhadas (número aleatório entre 4 e 16 containeres, mínimo de um por pilha) e por fim, o navio é colocado na fila.

Desembarcar

Após atracar todos os navios da iteração, começa o processo de desembarque, no qual um navio da fila desembarca um container em uma pilha da travessa de sua fila de atracamento. Esse é o processo:

O primeiro navio na fila verifica qual container deve ser desempilhado, desempilha esse container e coloca em uma das pilhas da sua travessa, sendo essa a primeira pilha disponível. Após esse processo, verifica-se se todas as pilhas do navio estão vazias, se estiverem, ele é desenfileirado.

Armazenar

Os containers numa pilha da travessa são levados por guias e encaminhados para armazenamento (abstração) por um carro, se o carro da fila estiver disponível. Se estiver, ele demora 2 ut pra concluir o processo.

Relatório

No final da iteração é feito um relatório, que mostra as filas de navios, as travessas e o tempo médio de espera por fila.

Reset

Se uma fila estiver vazia, ela é resetada, para evitar id's muito grandes dos navios.

Referências e Tecnologias

- Todas as estruturas de dado foram feitas usando como base os slides do professor Nivio Ziviani.
- As imagens foram feitas usando o microsoft paint.
- O programa foi codificado na linguagem C, usando o compilador gcc por meio do WSL (Windows Subsystems for Linux).
- Foi também criado, um *makefile* para compilação mais simplificada.
- Foram usados git e github para o controle de versão e repositório do projeto.