



Laboratório de Programação II

Trabalho II - 2017/I

Data de entrega: 26/06/2017

Prof. Márcio Sarroglia Pinho

Descrição Geral

Este trabalho deve ser desenvolvido, entregue e apresentado em ambiente LINUX.

Este trabalho, que deverá ser feito **individualmente ou em duplas**, consiste em modelar e implementar uma simulação de um restaurante que vende pastéis e pizzas.

A simulação funciona da seguinte forma:

A cada instante de tempo, pode chegar ou não um novo cliente com um pedido de pastéis ou pizzas. Para a entrega de qualquer deste alimentos, é necessário que sejam executadas três tarefas: montagem, cozimento e empacotamento.

O restaurante possui 3 montadores, 3 fornos e 3 empacotadores. Cada um destes só pode atender a um pedido de cada vez.

Ao chegar, o pedido de cada cliente é colocado numa fila de montagem, e atendido quando possível. Portanto, um pedido só sai da fila de montagem quando houver um montador disponível.

Ao terminar o processo de montagem, o pedido entra na fila de cozimento, e só vai para um dos 3 fornos, quando houver disponibilidade.

Finalmente, ao sair do cozimento, o pedido entra na fila de empacotamento, e similarmente, só vai para um dos 3 empacotadores quando este estiver livre. O processo termina quando o pedido sai do empacotamento.

Aspectos Técnicos

O simulador funciona através da passagem de tempo, realizada pelo pressionamento de uma tecla. Não é necessário simular o relógio real. Devem ser exibidas na tela as seguintes informações:

- Fila de Montagem
- Estado e número do pedido em cada montador
- Fila de Cozimento
- Estado e número do pedido em cada forno
- Fila de Empacotamento
- Estado e número do pedido em cada empacotador
- Tempo médio de atendimento dos pedidos

O tempo médio deve levar em conta o tempo que os pedidos levam para ficarem prontos, ou seja, desde o momento que chegam até o momento em que saem do sistema.

É importante lembrar que em um instante de simulação devem ocorrer TODAS as seguintes ações, como se estivessem acontecendo ao mesmo tempo:

Chegada de um novo pedido

A cada instante, deve ser gerado um número aleatório entre 0 e 99 (através da função *rand()*, ver documentação): se for maior que 80 (20% de chance), deve-se simular a chegada de um pedido. Quando o pedido chega, ele deve ser colocado na fila de montagem.

Um pedido é representado pelas seguintes informações: um **número seqüencial** (para controle), uma certa **quantidade de pizzas** (sorteada entre 1 e 3), uma certa **quantidade de pasteis** (sorteada entre 1 e 5) e o **instante de tempo** em que o pedido entrou no sistema (para o cálculo do tempo médio de atendimento).

Processo de Montagem

O tempo de montagem de um pedido corresponde ao tempo que ele deve ficar com um dos montadores.

Se houver um montador livre, o primeiro pedido da fila de montagem é atribuído a ele. Para cada montador que estiver montando uma pizza/pastel, o **tempo de montagem** restante deste é decrementado de uma unidade. Quando o tempo chegar a zero, o pedido sai da montagem e é colocado na fila de Cozimento (ver abaixo).

O tempo de montagem de cada pedido é calculado de acordo com a quantidade de cada tipo de alimento: 3 instantes de tempo para cada pizza, e 1 instante de tempo para cada pastel. Por exemplo, se um pedido tem 2 pizzas e 5 pastéis, então o tempo total será $3 \cdot 2 + 5 = 11$ instantes.

Processo de Cozimento

O tempo de cozimento de um pedido corresponde ao tempo que ele deve ficar no forno/fogão.

Havendo disponibilidade, o primeiro pedido da fila de Cozimento é colocado no forno/fogão. Para cada forno/fogão contendo um pedido em andamento, o **tempo de cozimento** restante deste é decrementado de uma unidade. Quando o tempo chegar a zero, o pedido sai do forno e é colocado na fila de pedidos a embalar (ver abaixo).

O tempo de cozimento é calculado de acordo com a quantidade de cada tipo de alimento: 4 instantes de tempo para cada pizza, e 2 instantes de tempo para cada pastel. Por exemplo, se um pedido tem 2 pizzas e 5 pastéis, então o tempo total será $4 \cdot 2 + 5 \cdot 2 = 18$ instantes.

Processo de Empacotamento

O tempo de empacotamento de um pedido corresponde ao tempo que ele deve ficar com um dos empacotadores. Se algum estiver livre, o primeiro pedido da fila de empacotamento é atribuído a ele. Para cada empacotador que estiver processando um pedido, o **tempo de empacotar** restante deste é decrementado de uma unidade. Quando o tempo chegar a zero, o pedido sai do empacotamento e é entregue ao cliente (ou seja, sai do sistema).

O tempo de empacotador é calculado de acordo com a quantidade de cada tipo de alimento: 1 instante de tempo para cada pizza e 1 instante para cada pastel. Por exemplo, se um pedido tem 2 pizzas e 5 pastéis, então o tempo total será 7 instantes.

Avaliação

Deve-se criar uma classe *Fila* que represente uma fila de pedidos. A criação de classes adicionais fica dependente da sua implementação (por exemplo, classes *Pedido*, *Forno*, *Empacotador*, *Montador*, etc). Um dos objetivos deste trabalho é a modelagem adequada dos atributos e métodos dessas classes.

Isso significa que você deve explorar os conceitos de programação orientada a objetos, e não simplesmente criar um longo programa usando apenas a classe *Fila*. Trabalhos que implementem a solução dessa forma, mesmo que 100% correta, sofrerão descontos na nota final.

O código-fonte deve estar adequadamente comentado (nas suas partes/algoritmos principais, pelo menos) e no seu início deve haver um comentário com o(s) nome(s) do(s) aluno(s).

Apresentação

O trabalho será apresentado NA SALA DE AULA, na máquina do professor. Por esse motivo, TODOS os trabalhos já devem ter sido enviados pelo Moodle antes do horário da aula. Durante a apresentação, **todos os alunos** devem estar presentes e aptos a responder quaisquer perguntas. Respostas insatisfatórias ou a ausência do aluno acarretarão em **anulação** da nota final.

Até ANTES do horário de início da aula do dia da apresentação, cada aluno ou dupla já deverá ter submetido um arquivo **.zip** (não RAR, nem 7z ou qualquer outro formato) pelo sistema *Moodle*, contendo o projeto completo do programa. O programa deverá ser nomeado da seguinte forma:

nome_sobrenome.zip No caso de uma dupla, usar a forma:

nome1sobrenome1_nome2sobrenome2.zip.

O programa deverá ser executado no sistema operacional Linux. Não serão aceitos trabalhos que não compilem corretamente no g++, sob hipótese nenhuma.

Todos os trabalhos deverão ser entregues até o dia da apresentação, e apresentados na data marcada - **não serão aceitos trabalhos fora do prazo.**

Trabalhos copiados resultarão em nota zero para todos os alunos envolvidos.

FIM