

Nome dos integrantes:

Tiago de Jesus Chehab – 2211194

Lucas Santana de Souza – 2211131

Gabriella Stefany Silva Pereira - 2211288

Relatório - Trabalho Prático Hovercooked

Introdução

Hovercooked é uma simulação de um jogo de restaurante onde os jogadores devem preparar e servir pratos em um ambiente de cozinha. Este documento detalha a implementação do jogo utilizando a concorrência dentro da linguagem de programação C. O jogo simula uma cozinha onde pedidos são gerenciados e preparados por cozinheiros. A implementação utiliza threads, seções críticas e sincronização para garantir o funcionamento correto do jogo.

Mecânica do Jogo

O jogo se desenrola em um loop, onde novos pedidos são adicionados de tempos em tempos a um mural de pedidos que devem ser concluídos antes do tempo limite acabar. Os jogadores controlam os cozinheiros, atribuindo-lhes pedidos específicos. Cada cozinheiro, representado por uma thread individual, processa um pedido de cada vez, seguindo as etapas de preparação e cozimento para depois concluir o mesmo.

A cozinha possui bancadas e a cozinha como recursos na qual os cozinheiros precisam compartilhar e não podem ser usados simultaneamente por dois ou mais cozinheiros. O jogo possui diferentes níveis de dificuldade, na qual irá desafiar o jogador de forma progressiva.

Sistema de Pontuação

O jogo possui um sistema de pontuação detalhado que avalia o desempenho do jogador. A pontuação é calculada com base nos seguintes fatores:

- **Pedidos entregues:** Cada pedido entregue corretamente adiciona pontos à pontuação.
- **Entregas rápidas:** Pedidos entregues em menos de 5 segundos concedem um bônus de pontos.
- **Pedidos errados:** Pedidos preparados incorretamente resultam em perda de pontos.

- **Satisfação do cliente:** O tempo médio de entrega dos pedidos influencia a satisfação do cliente, que, por sua vez, afeta a pontuação final. A satisfação do cliente é representada por uma porcentagem que varia de acordo com o tempo médio de entrega. Se o tempo médio for menor ou igual a 5 segundos, a satisfação é de 100%. Se for menor ou igual a 10 segundos, cai para 80%, e se for maior que 10 segundos, a satisfação é de 60%. Além disso, a cada pedido errado, a satisfação diminui em 10%.
- **Bônus por alta satisfação:** Uma alta satisfação do cliente (90% ou mais) garante um bônus adicional na pontuação.

Interface do Usuário (NCurses)

A interface do jogo é implementada utilizando a biblioteca ncurses, que permite a criação de interfaces de usuário em modo texto no terminal. A tela é dividida em três seções principais:

- **Pedidos:** Exibe a lista de pedidos a serem preparados, indicando quais já foram concluídos.
- **Cozinheiros:** Mostra o estado de cada cozinheiro (disponível, preparando ingredientes ou preparando prato), o tempo restante para concluir o pedido atual e a bancada/cozinha que está utilizando.
- **Recursos:** Apresenta a disponibilidade de bancadas e cozinhas.

Threads e Concorrência

Thread de mural de pedidos (*mural_pedidos*): Essa thread é responsável por gerar novos pedidos e adicioná-los ao mural de pedidos num intervalo de 1 a 5 segundos. A geração de pedidos é aleatória.

Thread de gerente (*gerente*): Essa thread lê comandos do usuário para atribuir pratos aos cozinheiros.

Thread de cozinheiro (*cozinheiro*): Cada cozinheiro é uma thread que espera um pedido, prepara os ingredientes, prepara o prato e libera os recursos.

Estrutura do Projeto

O projeto está dividido nos seguintes arquivos:

- **main:** Função principal que inicializa o jogo, cria as threads dos cozinheiros, gerente e mural de pedidos, e executa o loop principal do jogo.
- **cozinheiro:** Função executada por cada thread de cozinheiro, responsável por preparar os pratos atribuídos.

- **gerente:** Função executada pela thread do gerente, que recebe comandos do jogador para atribuir pratos aos cozinheiros.
- **mural_pedidos:** Função executada pela thread do mural de pedidos, que gera novos pedidos aleatoriamente.
- **atualizar_tela:** Função responsável por atualizar a interface do jogo com as informações mais recentes.
- **atualizar_pontuacao:** Função que calcula e atualiza a pontuação do jogador.
- **mostrar_menu_inicial:** Função que exibe um menu interativo para o usuário escolher a dificuldade.
- **obter_opcao_menu:** Função que captura a opção escolhida pelo usuário no menu.
- **mostrar_tela_inicial:** Função que exibe a tela inicial do jogo.
- **mostrar_tela_tempo:** Função que exibe a tela para o jogador definir o tempo de jogo.
- **fim_de_jogo:** Função que exibe a tela de fim de jogo e encerra o programa.

Compilação e Execução Requisitos GCC Biblioteca ncurses Compilação Para compilar o projeto, utilize o comando make: sh Copiar código make Execução Para executar o projeto, utilize o comando: sh Copiar código ./hovercooked

Estruturas de Dados

O código utiliza as seguintes estruturas de dados principais:

- **Prato:** Armazena informações sobre um prato, como código, tempo de preparo dos ingredientes, tempo de preparo do prato, nome e informações adicionais sobre o pedido (tempo de início e se foi preparado corretamente).
- **Cozinheiro:** Representa um cozinheiro, armazenando seu ID, estado (disponível, preparando ingredientes ou preparando prato), prato que está preparando, tempo restante para concluir o pedido, bancada e cozinha que está utilizando, além de informações sobre o pedido (tempo de início e se foi preparado corretamente).

Compilação e Execução Requisitos

1. Pré-requisitos:

- Sistema operacional Ubuntu.
- Compilador GCC.
- Biblioteca ncurses.

2. Instalar as Bibliotecas Necessárias:

- “sudo apt-update”

- “sudo apt-get install libncurses5-dev libncursesw5-dev”

3. Compilar código:

○ Para compilar o projeto, utilize o comando make no terminal bash para compilar os arquivos.

Para executar digite ./hovercooked ou make run

4. Executar o Jogo:

- ./hovercooked

Dificuldade do Jogo

Ao iniciar o jogo você terá que escolher a dificuldade do mesmo, podendo selecionar entre Fácil, Médio e Difícil com a complexidade do jogo aumentando de forma gradativa.

- Fácil: Você terá apenas um pedido e sem tempo limite.
- Médio: Você terá todas as bancadas e todos os cozinheiros a sua disposição.
- Difícil: Você terá uma bancada a menos para concluir seus pedidos.

Ao escolher o nível Médio ou Difícil, insira o tempo de jogo em segundos.

Controles do Jogo

Na interface do jogo é necessário que se use os seguintes comandos para gerenciar os cozinheiros:

- Cozinheiro 1: 11 (Pizza), 12 (Hambúrguer), 13 (Suco)
- Cozinheiro 2: 21 (Pizza), 22 (Hambúrguer), 23 (Suco)
- Cozinheiro 3: 31 (Pizza), 32 (Hambúrguer), 33 (Suco)
- Cozinheiro 4: 41 (Pizza), 42 (Hambúrguer), 43 (Suco)

É importante destacar que cada cozinheiro irá para uma bancada disponível de forma automática. Caso não tenha uma bancada disponível, o cozinheiro vai a alguma bancada após ela ficar livre.

Cada prato exige um certo tempo de preparo para ser finalizado, no qual irá consumir tempo na bancada:

- Pizza: 5 segundos.
- Hambúrguer: 3 segundos.
- Suco: 2 segundos.

Os pedidos chegam ao gerente (jogador), através do mural de pedidos que fica localizado à esquerda da tela do jogo, os pedidos irão chegar de forma aleatória. Após um cozinheiro concluir um pedido, irá aparecer no local onde ficava a demanda uma mensagem de concluído.

É necessário concluir todos os pedidos que irão chegar, antes do tempo determinado pelo jogador acabar. Caso não tenha sucesso, o usuário será notificado que a partida foi perdida e que o objetivo não foi concluído.

Conclusão

Este projeto demonstra a aplicação prática de conceitos de programação concorrente para simular um ambiente complexo como uma cozinha de restaurante. A utilização de threads, mutexes, variáveis de condição e semáforos permite a coordenação eficiente de múltiplos cozinheiros e o compartilhamento de recursos limitados. A interface baseada em texto, criada com a biblioteca ncurses, oferece uma representação visual clara do estado do jogo, tornando a experiência mais interativa e compreensível.

Caso queira acessar o repositório criado pelos alunos que desenvolveram o projeto é só acessar o link: <https://github.com/Gabriella-gssp/OvercookedThreads>.