

DOCUMENTO 4: Portfólio de avaliação

Aluno:	Ronyson da Silva Fonseca	Matrículas	32152120
Email:	ronyson.silva96@gmail.com		
Disciplina:	Jogos Para Consoles	Professor	José Ricardo da Silva Junior

Objeto: Verificação Suplementar

Habilidade a ser desenvolvida: Desenvolver uma solução para o problema do buffer limitado usando processos produtor e consumidor.

Produção do Aluno:

Link para o drive:

https://drive.google.com/drive/folders/1tFQUVc3_4dTIReQe_KHaigQclO0tF9yG?usp=sharing

Link para o repositório do git:

<https://github.com/ronysonSilva/VS-Consoles>

Link para o vídeo:

https://drive.google.com/file/d/1ZL_Ilr7D8BnFW3YPbeHmeQUmzD4bj7a3/view?usp=sharing

Relato do processo:

Desenvolvimento de uma solução para o problema Consumidor Produtor em Java no Eclipse utilizando um buffer.

Introdução

O trabalho atual irá apresentar uma solução para o clássico problema de sincronização de multiprocessadores "Consumidor/Produtor". O problema foi proposto inicialmente por Dijkstra. Para solucionar tal problema será utilizado diferentes semáforos que controlam o acesso ao que foi produzido e consumido dentro de um buffer circular. É esperado que com a solução apresentada todos os consumidores consigam consumir algo do que for produzido. Mesmo em uma situação onde existam diversos consumidores e apenas um produtor.

Materiais e métodos

A máquina utilizada nesta simulação foi um PC rodando Windows com um processador AMD FX-6300 de seis núcleos na frequência de 3.5 Gigahertz, 16 Gigabytes de Ram e uma Geforce 9500 GT.

As referências [1] , [2] e [3] foram usadas de base para a produção da solução.

Resultados

Seguindo ambas as referências e as direções apresentadas no documento de VS iniciou-se o processo de confecção do código, começando pelo Buffer, qual por um desconhecimento de como funciona, iria requerer uma quantidade maior de trabalho de minha parte para produzi-lo. Após rever o material disponibilizado para a tarefa da corrida de grilos (Objeto 3). Produzi um rascunho de código considerando que o buffer teria diversos semáforos, um para controlar o produtor tendo uma quantidade de 5 espaços, outro de 5 espaços para o consumidor e por fim um semáforo de apenas um espaço, sendo assim um mutex. Considerando a referência [3].

Foi produzido em conjunto um rascunho de um buffer circular, considerando os pontos apresentados na referência [1] foi produzido 2 métodos um para verificar se o próximo espaço está livre do buffer circular. E se o semáforo do produtor estava disponível. Por fim, foram criados no buffer os métodos de inserir e remover o que foi produzido pelo produtor e pelo consumidor, respectivamente. Sendo chamados de InsertItem & RemoveItem. Fora isso, o buffer circular também utiliza de algumas variáveis inteiras para verificar se o espaço atual está disponível ou não e através de algumas variáveis saber onde está sua "cabeça" e sua "cauda".

Com o rascunho de estrutura de forma funcional e feedback de outros alunos, foi iniciada a produção da segunda parte do trabalho, que seria a produção do consumidor e produtor. Ambos receberam a extensão para uso de threads, e dentro do seu método run herdado da thread foi utilizado chamado o "insertitem" do buffer no caso do produtor ou "removeitem" também presente no buffer no caso do consumidor.

Tanto o Buffer, quanto os produtores são objetos com construtores, considerando que seriam objetos que a classe main teria de instanciar.

Por fim, seguindo os requisitos do documento de VS foi desenvolvida a classe Main. Nela é iniciado o buffer e definido o tamanho dos semáforos que o mesmo irá utilizar. E, também é através dela que a quantidade de consumidores e produtores é definida. Assim como um tempo TOTAL de espera, qual é utilizado para informar quando devem ser paradas todas as "threads" de produção e consumo. Foi utilizado um laço do tipo "for" para "instanciar" todos os consumidores e produtores após o buffer ser iniciado. Após isso a classe main qual também estende uma thread utiliza o comando Thread.Sleep pelo tempo de espera definido pelo usuário. E após o final desse tempo de espera a simulação encerra informando o que foi produzido por cada

produtor e o que foi consumido por cada consumidor.

Como pontuado na introdução neste trabalho na produção atual mesmo em uma situação com apenas 1 produtor e 5 consumidores todos acabam conseguindo consumir. Resolvendo em si o problema.

Discussão e Conclusão

Com o código produzido é possível verificar que o problema foi solucionado, e que o uso dos semáforos resolveu o acesso simultâneo do produtor e consumidor ao buffer. Garantindo que tal ação não aconteça. E com isso que não aconteça nenhuma perda de dados mediante ao acesso simultâneo.

Referências

- [1] - https://en.wikipedia.org/wiki/Producer%E2%80%93consumer_problem - Página da wikipedia contendo um pseudo código e descrição do problema e outras soluções. Acesso em 19/06/2021
- [2] - <https://www.youtube.com/watch?v=eCjUuBeu1-8> - Aula sobre o problema “Jantar dos Filósofos”, a qual apresenta uma solução utilizando um mutex. Acesso em 19/06/2021
- [3] - https://en.wikipedia.org/wiki/Circular_buffer#:~:text=In%20computer%20science%2C%20a%20circular,asily%20to%20buffering%20data%20streams. - Referência de buffer circular.