

Pontifícia Universidade Católica de Campinas Faculdade de Engenharia de Computação - FECOMP

Sistemas Operacionais A – Relatório Experimento 5

Beatriz Morelatto Lorente RA: 18071597

Cesar Marrote Manzano RA: 18051755

Fabricio Silva Cardoso RA: 18023481

Pedro Ignácio Trevisan RA: 18016568

**Sumário**

**1.Introdução.......................................................................................................3**

**2.Apresentação dos erros do programa exemplo e suas soluções...............4**

**3.Respostas das perguntas...............................................................................7**

**4.Resultados da execução do programa exemplo...........................................9**

**5.Resultados da execução do programa modificado....................................12**

**6.Análise dos Resultados................................................................................16**

**7.Conclusão.............................................................................................**

**Introdução**

O experimento realizado permitiu o entendimento do uso de threads na linguagem de programação C, como forma de se obter concorrência em um programa. O experimento foi divido em duas tarefas.

Na primeira tarefa, foi executado um programa exemplo, ilustrando o problema do produtor consumidor. Havia 10 threads produtoras e 10 consumidoras, que compartilham um buffer, percorrido de maneira circular. As threads produtoras armazenam o valor 10 no buffer, e as consumidoras retiram esse valor. O problema dessa tarefa está em como manipulamos os acessos a esse buffer, de modo que as condições de corrida ocorram o menor número de vezes possível.

Na segunda tarefa, é proposto a solução do problema do jantar das filósofas (adaptação do problema clássico do jantar dos filósofos). No problema, cada filósofa é uma thread, e cada uma irá comer no máximo 365 vezes. No lugar do estado de “pensando”, foi programado uma espera de 25 microssegundos. A tarefa foi importante para observarmos o uso do mutex, para garantir exclusão mútua.

**Respostas das perguntas**

**Perguntas do relatório**

***Pergunta 1: Qual é o recurso comum que necessita de exclusão mútua ?***

**Resposta:** O número de cadeiras disponíveis será um recurso comum do barbeiro e do cliente portanto será necessário a exclusão mútua nessa variável.

***Pergunta 2: De que maneira (leitura, escrita, ambos) barbeiros e clientes vão acessar o recurso comum ?***

**Resposta:**

***Pergunta 3: Como os números foram colocados na String ?***

**Resposta:**

***Pergunta 4: Como o barbeiro vai ter acesso aos valores a serem ordenados ?***

**Resposta:** O cliente passa uma mensagem com a String para ser ordenada, é necessário converter em inteiro e então ordenar.

***Pergunta 5: Como o cliente vai ter acesso aos resultados?***

**Resposta:** Por meio de uma mensagem enviada pelo barbeiro.

**Resultados da execução do programa 1**

**Resultados da execução do programa 2**

**Análise dos resultados**

**Programa 1**

**Programa 2**

**Conclusão**