Experimento #4

Sistemas Operacionais A

**Introdução:**

O experimento consiste na elaboração e implementação do “Barbeiro Dorminhoco” na linguagem C com duas diferentes formas de exclusão mútua do recurso “waiting”, que representa a quantidade de clientes sentados na sala de espera aguardando o corte.

Na primeira parte deve-se utilizar fila de mensagens e semáforos como garantia da exclusão mútua e troca de informações entre o cliente e barbeiro

Na segunda parte, o mecanismo aplicado devem ser os threads, utilizando mutex, e semáforos para a proteção de “waiting”, e outros métodos para o acesso e alteração das informações do corte.

Em ambas as partes, o cliente gera um vetor de caracteres de tamanho aleatório entre 2 e 1023, com os valores também nesse intervalo, e a função do barbeiro é tratar o recebimento desse String, ordenando-o (decrescente com fila e crescente com threads) e devolvendo-o de volta ao cliente. Após receber os dados o cliente deve exibir seu número e o do barbeiro que o atendeu, o tempo de espera decorrido entre o começo da espera e fim do corte, e os vetores antes e depois da ordenação. E caso o cliente não seja atendido também deve-se apresentar qual seu número.

**Respostas às perguntas:**

* **Contidas no texto do Experimento:**
* **Qual é o recurso comum que necessita de exclusão mútua?**

A variável waiting é o recurso que necessita de exclusão mútua.

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

Para não ocorrer problemas com a escrita/leitura desordenada será implementado, por meio de fila de mensagens, a exclusão mútua ao acesso e atualização desta variável.

* **Como os números foram colocados na String?**

O número gerado aleatoriamente é transcrito, com a função sprintf(), em um vetor de caracteres auxiliar com tamanho máximo de 5. Após guardado, o número transcrito é copiado para o vetor principal índice por índice, e ao chegar ao fim do vetor auxiliar é adicionado um espaço antes que o próximo número seja inserido.

Ao fim da inserção de todos os números no vetor principal, é inserido o caractere de finalização de String, ‘\0’. O código do procedimento e sua chamada estão apresentados abaixo:

Texto

Descrição gerada automaticamente

Figura 1. Função criada para adicionar os itens ao vetor principal

Texto

Descrição gerada automaticamente

Figura 2. Chamada da função

* **Como o barbeiro vai ter acesso aos valores a serem ordenados? Como o cliente vai ter acesso aos resultados?**

Por serem gerados como threads, os processos barbeiros e clientes compartilham a mesma memória do processo total, e assim variáveis globais são vistas por todos os processos leves gerados, permitindo acesso às cadeiras presentes no salão, e suas respectivas variáveis.

**Análise dos Resultados:**

Desconsiderando o tipo de ordenação realizada (crescente ou decrescente), o resultado foi idêntico em ambas as partes do experimento

**Conclusão:**

Apesar dos resultados serem os mesmos, a implementação de diferentes mecanismos de exclusão mútua gera preocupações diferentes ao decorrer do código. Ao se utilizar a fila de mensagens deve-se preocupar com quem recebe a mensagem correta, pois um cliente pode receber o “cabelo” de outro caso o tipo de mensagem não seja determinado com êxito. Contudo ao utilizar threads, saber de quem é o cabelo se torna mais complicado, para que seja possível organizar os atendimentos foram utilizadas estruturas de Cadeiras que contém as diversas informações sobre o corte permitindo sua impressão.