Paradigmas - Seminário 2: Parte 2

Frank Montes
Gabriel Francisco
Patrick Duarte
Roberto Ramos

Cenário Concorrência - Relatório

1. QUAL MECANISMO FOI USADO PARA PROTEGER A REGIÃO CRÍTICA?

O mecanismo utilizado foi o **SEMÁFORO SysV** (System V Semaphore), implementado através das funções PHP:

- sem_get(): Cria ou obtém um semáforo com capacidade para 1 processo
- sem_acquire(): Trava o semáforo (operação P/wait)
- sem_release(): Libera o semáforo (operação V/signal)

REGIÃO CRÍTICA PROTEGIDA:

- A região crítica no código é o acesso à fila de pedidos compartilhada, que inclui:
- Verificação se a fila existe (shm_has_var)
- Leitura da fila (shm_get_var)
- Remoção de um pedido da fila (array_shift)
- Atualização da fila na memória compartilhada (shm_put_var)
- Atualização do array de pedidos por cozinheiro

COMO FUNCIONA:

1. Antes de acessar a fila, cada cozinheiro executa sem_acquire($sem_id)
2. Isso garante que apenas UM cozinheiro por vez possa acessar a fila
3. Após terminar as operações na fila, o cozinheiro executa sem_release($sem_id)
4. Isso permite que outro cozinheiro entre na região crítica

JUSTIFICATIVA:

O semáforo é necessário porque múltiplos processos (cozinheiros) acessam o mesmo recurso compartilhado (fila de pedidos). Sem o semáforo, ocorreriam race conditions, onde dois cozinheiros poderiam:

- Ler o mesmo pedido
- Tentar remover o mesmo elemento da fila
- Corromper a estrutura da fila
- Perder pedidos ou processá-los em duplicidade

2. COMO VOCÊ VALIDOU QUE AGORA FUNCIONA CORRETAMENTE?

MÉTODOS DE VALIDAÇÃO IMPLEMENTADOS:

A. **CONTAGEM DE PEDIDOS:**
   a. Inicialização: 100 pedidos (0 a 99)
   b. Verificação: Cada pedido deve ser processado exatamente uma vez
   c. Implementação: O código rastreia quais pedidos cada cozinheiro processou

B. **ACOMPANHAMENTO POR COZINHEIRO:**
   a. Cada cozinheiro mantém uma lista dos pedidos que processou
   b. Ao final, o sistema exibe quantos pedidos cada cozinheiro atendeu
   c. Isso permite verificar se há duplicatas ou pedidos perdidos

C. **MENSAGENS DE DEBUG:**
   a. "Cozinheiro X preparando pedido=Y"
   b. "Cozinheiro X terminou pedido=Y"
   c. Essas mensagens permitem rastrear o fluxo de execução

D. **VERIFICAÇÃO DE INTEGRIDADE:**
   a. Se a fila ficar vazia, ela é removida da memória compartilhada
   b. Cozinheiros param de executar quando não há mais pedidos
   c. O sistema aguarda todos os processos terminarem antes de finalizar

CONCLUSÃO:

O uso do semáforo SysV garante que apenas um processo por vez acesse a região crítica, eliminando race conditions e garantindo a integridade dos dados compartilhados. A validação pode ser feita através da verificação da completude e unicidade dos pedidos processados.

RESULTADOS DOS TESTES EXECUTADOS:

Para validar o funcionamento correto do sistema, foram executados 10 testes sequenciais com os seguintes resultados:

ESTATÍSTICAS DOS TESTES:

- **Total de execuções: 10**
- **Sucessos: 10**
- **Falhas: 0**
- **Taxa de sucesso: 100%**

VALIDAÇÕES REALIZADAS EM CADA TESTE:

✓ Todos os pedidos foram processados (100 pedidos únicos)

✓ Nenhum pedido foi duplicado

✓ Nenhum pedido foi perdido (números 0-99 todos presentes)

✓ A fila de pedidos terminou vazia

✓ Soma matemática correta (0+1+2+...+99 = 4950)

DETALHAMENTO DOS RESULTADOS:

```
=== RESUMO DOS TESTES ===
Total de execuções: 10
Sucessos: 10
Falhas: 0
Taxa de sucesso: 100%

=== RESULTADOS DETALHADOS ===
Exec | Todos | Sem   | Sem   | Fila  | Soma  | Total | Status
     | Proc. | Dupl. | Perd. | Vazia | OK    | OK/5  |
-----+-------+-------+-------+-------+-------+-------+--------
   1 |   ✓   |   ✓   |   ✓   |   ✓   |   ✓   |  5/5  | SUCESSO
   2 |   ✓   |   ✓   |   ✓   |   ✓   |   ✓   |  5/5  | SUCESSO
   3 |   ✓   |   ✓   |   ✓   |   ✓   |   ✓   |  5/5  | SUCESSO
   4 |   ✓   |   ✓   |   ✓   |   ✓   |   ✓   |  5/5  | SUCESSO
   5 |   ✓   |   ✓   |   ✓   |   ✓   |   ✓   |  5/5  | SUCESSO
   6 |   ✓   |   ✓   |   ✓   |   ✓   |   ✓   |  5/5  | SUCESSO
   7 |   ✓   |   ✓   |   ✓   |   ✓   |   ✓   |  5/5  | SUCESSO
   8 |   ✓   |   ✓   |   ✓   |   ✓   |   ✓   |  5/5  | SUCESSO
   9 |   ✓   |   ✓   |   ✓   |   ✓   |   ✓   |  5/5  | SUCESSO
  10 |   ✓   |   ✓   |   ✓   |   ✓   |   ✓   |  5/5  | SUCESSO

🎉 RESULTADO: Todos os testes passaram! O sistema funciona corretamente.
O semáforo está protegendo adequadamente a região crítica.
```

CONCLUSÃO:

O uso do semáforo SysV garante que apenas um processo por vez acesse a região crítica, eliminando race conditions e garantindo a integridade dos dados compartilhados.