

Para os seguintes problemas, monte soluções em CSP, demonstre as propriedades sugeridas, a seguir implemente os modelos na linguagem Go seguindo a estrutura de canais e processos do seu modelo CSP. Discuta aspectos principais do modelo e da implementação.

1. O Problema dos Leitores e Escritores: Este problema pode se instanciar com qualquer estrutura de dados, banco de dados ou sistema de arquivos que seja lido e modificado por threads concorrentes. Uma estrutura (qualquer) de dados pode ser lida ou escrita (modificada). Assuma entretanto que esta estrutura pode ser complexa e uma operação de leitura ou de escrita não é atômica e necessita sincronização. As threads são divididas em categorias: leitoras e escritoras. Enquanto uma thread está lendo, outras threads leitoras podem ler simultaneamente. Uma thread escritora tem que ter acesso exclusivo à seção crítica. Ou seja, uma thread escritora não pode entrar na seção crítica enquanto qualquer outra thread (leitora ou escritora) está acessando. Tarefa: monte um modelo CSP para os leitores e escritores, e uma implementação em GO seguindo este modelo que pode suportar número flexível de threads.
 - a partir do modelo CSP, demonstre exclusão mútua para thread escritora, demonstre não bloqueio;
 - esta especificação permite starvation (postergação) de escritores. Demonstre se isto ocorre no seu modelo. Discuta se voce observa isso em alguma configuração do seu programa GO;
 - discuta o modelo e a implementação;
 - apresente o código de ambos devidamente comentados;
2. O Problema dos Leitores e Escritores - sem Starvation: Tarefa: Estenda a solução anterior para que os escritores não sejam indefinidamente postergados.
 - a partir do modelo CSP, demonstre exclusão mútua para thread escritora, não bloqueio, não postergação;
 - discuta o modelo e a implementação;
 - apresente o código de ambos devidamente comentados.
3. O Problema dos Leitores e Escritores - sem Starvation - prioridade para Escritores: Tarefa: Estenda a solução anterior para que os escritores tenham prioridade. Ou seja, se um escritor está acessando a estrutura, enquanto houverem escritores esperando a seção crítica será passada para outro escritor, independente de haverem leitores esperando. Quando a fila de escritores estiver vazia, os leitores podem proceder todos concorrentemente suas leituras. Discuta as mesmas propriedades
 - a partir do modelo CSP, demonstre exclusão mútua para thread escritora, não bloqueio, a prioridade de escritoras e a possibilidade de postergação de leitoras;
 - discuta o modelo e a implementação;
 - apresente o código de ambos devidamente comentados.

Entrega: solicita-se a entrega de texto em formato de artigo, contendo: título, aluno(a)(s), resumo, introdução apresentando o problema, e as demais seções conforme necessidade do enunciado. Solicita-se ainda que modelo e código sejam entregues no moodle na data definida. Sugere-se uso do LaTeX. Seguir o formato disponível no moodle para isso.