

Enunciado de Trabalho

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

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. Suponha uma estrutura (qualquer) de dados que 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 processos (Go routines).

- a partir do modelo CSP, demonstre: exclusão mútua para thread escritora; não bloqueio;
 - a especificação acima permite starvation (postergação) de escritores. Demonstre se isto ocorre no seu modelo. Discuta se você observa isso em alguma execução do seu programa GO.
 - apresente o código de ambos devidamente comentados. Isto inclui:
 - Nome do autor em todos os fontes;
 - Referência devida caso exista reuso de código;
 - Comentários extensivos sobre cada definição do seu programa;
 - Responda às solicitações acima com comentários no modelo CSP ou no código Go;
 - Construa seu programa para, além de funcionar, ser lido e entendido por outra pessoa. Programar para funcionar é diferente, e apenas uma parte, de programar escrevendo uma idéia para que outra pessoa possa entender. Isto se faz com construções adequadas e com comentários.
2. 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 quer escrever, novos leitores não devem acessar a estrutura. Ainda, se um escritor está acessando a estrutura, enquanto houverem escritores esperando estes terão direito de escrita antes de qualquer processo que deseja fazer leitura. Quando a fila de escritores estiver vazia, os leitores podem proceder 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;
 - responda às solicitações acima com comentários no modelo CSP ou no código Go;
 - apresente o código de ambos devidamente comentados (vide observação no item 1).

Entrega: solicita-se a entrega dos fontes deixando claros os pares de:

⟨ modelo CSP ; respectivo código fonte Go ⟩.

Como são duas situações, serão entregues dois pares. A avaliação será somente com base nos modelos/códigos fonte e seus comentários inseridos nos mesmos.