

Q. 1a) O programa funciona com dois semáforos, um chamado Mutex e um chamado roomEmpty. Quando um leitor vai escrever, ele tranca o roomEmpty, impedindo outros processos de acessar a memória, escreve e então destranca. Quando um leitor quer imprimir a memória, ele irá usar o Mutex para trancar o acesso ao semáforo roomEmpty e verificar se é o primeiro a entrar, caso que requer trancar a memória com roomEmpty. Os leitores podem acessar a memória concorrentemente, mas isso irá bloquear escritores tentando escrever. Uma vez que o último leitor leia, ele irá destrancar o roomEmpty e permitir a escritores escreverem na memória. **Por consequência, ocorre starvation.**

Q. 1b) Funciona muito similar à versão com starvation, mas com a adição de um semáforo chamado catraca. Este semáforo é bloqueado pelo primeiro escritor na fila de acesso à memória, impedindo outros leitores de passarem na frente. Todos os leitores são presos na catraca até que o último escritor destranque a catraca, e então trancam o acesso a memória até que outro escritor entre na fila e todos os leitores acessando encerrem seu acesso e tudo se repita. **Nesta versão portanto, não ocorre starvation.**

Q. 1c) Nesta versão, os leitores irão tentar ler a memória e impedir um escritor de escrever, porém o primeiro escritor a chegar irá bloquear quaisquer leitores seguintes de ler a memória. O processo escritor bloqueia outros escritores e todos os leitores, mas similar à versão com starvation, os escritores só permitirão leitura assim que todos tiverem escrito. **Dessa forma, os escritores tem prioridade e os leitores estão sujeitos à starvation.**

Q. 2a) O programa funciona idem a versão da questão 1a, porém utilizando canais bufferizados ao invés de semáforos. Os processos irão tentar remover uma struct vazia dos canais mutex e roomEmpty como fariam para esperar um semáforo, e uma vez que encerrem sua sessão crítica, irão colocar uma struct vazia no canal, simulando o signal de um semáforo. **Como o código segue a mesma estrutura da questão 1a, pode ocorrer starvation pois leitores tem prioridade**

Q. 2b) Idem a versão com semáforos, esta versão adiciona um canal bufferizado chamado catraca que serve para trancar todos os leitores em sua fila enquanto o escritor atual realiza sua escrita, liberando os próximos leitores assim que terminar de escrever, **preservando a propriedade de não ocorrer starvation de escritores**

Q. 2c) Nesta última versão, os semáforos também são substituídos por canais. Seguindo a mesma estrutura da versão com semáforos, os processos escritores, sempre que chegarem para escrever, irão bloquear a fila de leitores de forma que enquanto houverem processos escritores chegando, os leitores irão ser bloqueados. Uma vez que não hajam mais escritores, os leitores são liberados para escrever. Os escritores tem prioridade e os leitores ficam suscetíveis à starvation.