

RELATÓRIO SEGUNDO TRABALHO COMPUTAÇÃO CONCORRENTE 2019.2

ALUNOS: Matheus Araujo Borges, Willian Raphael Rosa Gomes.

DRE: 118057290, 118058482.

Projeto de Solução:

Para resolver o problema dos leitores/escritores, os dois principais conceitos de concorrência utilizados foram: variáveis de condição e exclusão mútua utilizando a linguagem C.

Tanto as threads escritoras quanto as leitoras então fundamentalmente divididas em três “blocos” de execução, cada um tem um papel específico durante a execução do programa.

O primeiro bloco tem como função determinar se a thread pode ou não executar, ou seja, threads leitoras não executam ao mesmo tempo que threads escritoras, mais de uma thread escritora não pode executar ao mesmo tempo e threads leitoras podem executar ao mesmo tempo. Isso é determinado utilizando duas variáveis de condição e uma de exclusão mútua, trabalhando da seguinte forma: caso seja uma thread escritora ela verifica o número de threads lendo, o número de threads escrevendo e uma variável de sinal (que será explicada mais adiante), caso as duas primeiras sejam zero ela poderá executar, caso contrário ela será interrompida. Caso a thread em questão seja uma thread leitora ela verifica somente o número de threads escrevendo e uma outra variável de sinal (que também será explicada mais adiante), e caso a primeira seja zero ela prosseguirá com sua execução e caso contrário esperará. Caso uma thread espere, ela acrescentará em uma unidade uma das variáveis que guarda quantas threads escritoras ou leitoras estão bloqueadas. Já caso

uma thread prossegua, ela adicionará um ao número de threads escrevendo ou lendo, de acordo com seu tipo.

O segundo bloco, nas threads leitoras, tem como objetivo efetuar a leitura de fato, ou seja, gravar no arquivo .txt o valor atual da variável compartilhada. Já nas escritoras, o seu papel é de gravar na variável compartilhada o valor do id da thread.

Já o terceiro bloco, em ambos os tipos de thread, tem o papel de finalizar a leitura/escrita e sinalizar para uma das duas filas(variáveis de condição) que elas podem prosseguir. Para determinar para qual variável de condição será enviado o sinal de desbloqueio, caso seja uma thread escritora, ela verificará se há threads leitoras bloqueadas, se houver, ela as desbloqueará, caso contrário se houverem threads escritoras bloqueadas elas receberam sinal e caso nenhum tipo de thread esteja bloqueada, somente um unlock será executado.

Para tratar o caso de starvation do programa, duas variáveis foram utilizadas, sinal e sinal2. Essas, por sua vez, armazenam o tipo da thread que recebeu um sinal para prosseguir sua execução, caso o sinal foi recebido para liberar a fila das threads escritoras, é garantido que a próxima thread a executar seu segundo bloco será escritora e o mesmo se aplica para as leitoras. Dessa forma, com essa alternância entre a execução das threads, nenhuma “passará fome”.

Arquivo Log:

O arquivo de log gerado será composto de diversas chamadas de função que, por sua vez, serão lidas e verificadas por um programa logger feito na linguagem python 3.7. O programa principal registrará cada alteração realizada em variáveis globais em um arquivo de log que terá seu nome informado na execução do programa, além disso tal arquivo também receberá o valor inicial das mesmas. O programa logger, terá as mesmas variáveis do programa original e as iniciará com o valor contido no log, em seguida ele processará as alterações realizadas nestas e as comparará com o valor de suas próprias variáveis, caso algum problema seja encontrado, o mesmo será impresso na tela e o programa finalizará, caso contrário, o programa mostra uma mensagem dizendo que não houveram erros na execução e fecha.

Testes Realizados:

Realizamos testes com valores diversos de entrada do programa e para confirmar se os resultados eram os desejados, utilizamos o arquivo de log obtido. Checamos o arquivo tanto utilizando o programa auxiliar, tanto verificando-o manualmente e os resultados obtidos foram os esperados.

Utilizamos os seguintes casos de teste no programa logger:

6 threads leitoras, 3 escritoras, 7 leituras e 4 escritas;

3 threads leitoras, 6 escritoras, 4 leituras e 7 escritas;

6 threads leitoras, 6 escritoras, 4 leituras e 7 escritas;

6 threads leitoras, 6 escritoras, 7 leituras e 4 escritas;

6 threads leitoras, 6 escritoras, 7 leituras e 7 escritas;

1 thread leitora, 6 escritoras, 7 leituras e 7 escritas;

6 threads leitoras, 1 escritora, 7 leituras e 7 escritas;

1 thread leitora, 1 escritora, 7 leituras e 7 escritas;

Em todos os casos acima o resultado obtido foi:

Programa executou sem problemas