

Nome: André Martins Ferreira

Cartão: 159098

Os programas são compostos por 3 arquivos, `readwrite.c` e `rwmonitor.h` que são comuns a ambas as versões, e `rwmonitor1.c` e `rwmonitor2.c` que são exclusivos da versão 1 e 2 respectivamente.

O arquivo `readwrite.c` contém o código de inicialização do programa, com a leitura da entrada, chamada da função de inicialização da estrutura utilizada para sincronização (`monitor_init`) e criação das threads, além do código dos escritores e leitores.

Os leitores e escritores possuem cada um sua própria thread, que fica em um loop infinito, dentro qual a uma seção crítica guardada por uma estrutura denominada `rwmonitor`. Para entrar e sair da seção, os leitores utilizam a função `enter_reader_monitor` e `exit_reader_monitor` respectivamente, já os escritores utilizam `enter_writer_monitor` e `exit_writer_monitor`.

No arquivo `rwmonitor.h` está descrito a interface dessas funções, e a definição do tipo `rwmonitor_t`, que é declarado como um ponteiro void, que serve como proxy para `rwmonitor_t_struct`, cuja implementação depende da versão.

O arquivo `rwmonitor1.c` possui a implementação das funções de sincronização utilizando semáforos. Nele, a estrutura `rwmonitor_t_struct` é implementada com 2 semáforos e um número inteiro `nr`. Um dos semáforos é denominado `r`, o outro `w`.

O semáforo `r` protege o inteiro `nr`, já o semáforo `w` protege a seção crítica em si, e é sua posse alterna entre livre, dos leitores, ou de um escritor específico. O inteiro `nr` contém o número de leitores atualmente na seção crítica.

O protocolo de entrada para os leitores é o seguinte: Primeiro é feito um `wait` no semáforo `r`. em seguida é incrementado `nr`, caso `nr` seja 1 após o incremento significa que não há outro leitor na zona crítica, mas que pode haver um escritor atualmente, e também é necessário se proibir a entrada de escritores enquanto leituras estão ocorrendo, nesse caso é feito um `wait` no semáforo `w`. Então é feito um `post` no semáforo `r`, permitindo que outros leitores também entrem na seção crítica, já que múltiplos leitores podem entrar nela ao mesmo tempo.

O protocolo de saída dos leitores é parecido, primeiro é feito um `wait` no semáforo `r`, então se decrementa `nr`. Se `nr` é 0, significa que não há mais leitores, então o acesso dos escritores deve ser permitido através de um `post` no semáforo `w`. Em seguida é feito um `post` no semáforo `r`.

O protocolo dos escritores é mais simples, sendo feito um `wait` no semáforo `w` para a entrada, e um `post` nele para a saída.

O arquivo `rwmonitor2.c` possui a implementação das funções de sincronização utilizando variáveis de condição. A estrutura `rwmonitor_t_struct` é implementada com 2 variáveis de condição, `condr` e `condw`, um mutex denominado `lock`, e 2 inteiros, `nr` e `nw`.

`Nr` e `nw` contém respectivamente o número de leitores e escritores dentro da seção crítica.

`Condr` é a variável de condição onde leitores esperam caso um escritor esteja dentro da seção, e `condw` é a variável na qual escritores esperam caso leitores ou um escritor esteja dentro da seção. `Lock` serve como o monitor dessas variáveis de condição, além de prover exclusão mútua ao acesso de `nr` e `nw`.

O protocolo de entrada dos leitores é o seguinte: primeiro se adquire a `lock`, então enquanto há escritores na seção crítica, denotado por `nw` ser maior do que 0, se espera em `condr`. Então `nr` é incrementado, e se libera a `lock`.

No protocolo de saída dos leitores, é adquirida a lock, então é decrementado nr, caso nr se torne 0 significa que não há mais leitores dentro da seção, e um signal é feito em condw para acordar um escritor que esteja esperando, e então a lock é liberada.

O protocolo de entrada dos escritores é parecido com o de entrada dos leitores. Se adquire a lock, enquanto há escritores ou leitores na seção crítica ($nw > 0$ ou $nr > 0$) se espera em condw, nw é incrementado, e se libera a lock.

O protocolo de saída dos escritores: Adquire-se a lock, decrementa-se nw, é feito um signal em condw e um broadcast em condv acordando até um escritor e todos os leitores que estejam esperando, e então é liberada a lock.

Um Makefile é provido junto com o código fonte, então a compilação pode ser feita através do comando make. Para executar a primeira versão, com semáforos, executa-se `./readwrite1 <numero de leitores> <numero de escritores>`, para executar a segunda versão, com variáveis de condição, o comando é `./readwrite2 <numero de leitores> <numero de escritores>`.