



UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

Figure 1: Logo Ufrj

Lista 2 computacao concorrente

Lucas Tatsuya Tanaka

Prof:Silvana Rosseto

Dre:118058149

May 24, 2021

1 Leitores e escritores com prioridade para escrita

Os requisitos necessários para a elaboração do problema seria um programa concorrente ao qual obedece o padrão leitor escritor com prioridade para escritores baseando se principalmente em um formato de atuadores e sensores na qual a principal função dos sensores seria a captação dos dados de temperatura que variam de 25 a 40 graus, já os atuadores sendo responsável por emanar sinais vermelhos caso a temperatura captada nas últimas 5 medidas for maior que 35, amarelo caso nas últimas 15 medidas tenha se medido 5 vezes temperatura maior do que 35, e verde caso nenhuma das possibilidades seja correspondida. Com relação à prioridade para a escrita o principal funcionamento seria o acesso sendo garantido para leitores a qualquer momento ao recurso compartilhado porem com o excesso de tempos quando o escritor utiliza o espaço compartilhado os leitores são bloqueados e dão prioridade para os escritores estes que escrevem um de cada vez no espaço compartilhado tendo prioridade para os escritores na qual quando durante fila de execução aparecer um escritor este ganha

preferência para a execução da escrita na variável compartilhada este sendo feito através do formato FIFO(file in file out). O padrão adaptado no programa apresenta possibilidade de starvation para as Threads leitoras na qual estas tem grande dificuldade para ganhar execução durante o processamento do programa possibilitando assim um constante bloqueio das Threads e impossibilidade do acesso da mesma. As principais decisões tomadas foram a divisão do projeto em dois principais arquivos de TAD sendo eles Queue.h e Thread.h na qual Queue.h fica responsável pelas funções utilizadas para atender o padrão FIFO requisitado pelo problema, e Thread.h contendo as funções responsáveis por controlar os fluxos de execução do programa e as próprias Threads Leitora e Esritora. Para ser possível a implementação de Threads junto a TAD é necessário a utilização de funções inicializadoras para que as variáveis globais pudessem ser acessadas por todas as Threads de maneira a respeitar as particularidades da implementação de TAD além disso As principais duas estruturas de dados utilizadas são Queue e Estrutura na qual Queue é composto por 4 inteiros e um ponteiro de inteiro na qual os 4 inteiros servem como identificadores de posição, da frente, e posição de trás e o ponteiro representando um array a estrutura Estrutura é composto por 3 Queue cada uma representando os valores de temperatura captado os ids e a quantidade de vezes em que o valor foi lido. Se apresenta funções responsáveis pelo controle da estrutura Queue como isFull que mede quando uma fila se encontra cheia, isEmpty para quando uma fila se encontra vazia, front que mostra a posição do valor que esta a frente da fila, rear que mostra a posição do valor que esta no final da fila, dequeue que retira o primeiro valor da fila, enqueue que coloca o valor no final da fila e createQueue que inicializa a estrutura de fila. Em seguida as principais funções responsáveis pelo controle de Threads no padrão leitor escritor com prioridade para escritores são apresentadas como entLeitor, saiLeitor, entEscritor, saiEscritor que garantem a execução das Threads na ordem especificada pelo padrão, além disso, se encontra as funções Leitor e Escritor que realizam as Threads leitoras e escritoras respectivamente. São apresentadas 2 principais funções responsáveis pela inicialização de estruturas e variáveis de controle de processo, createEstrutura é responsável pela inicialização da estrutura Estrutura e thread_lock_init sendo responsável por inicializar as variáveis de semáforo responsáveis pelos bloqueios e exclusão mutua das Threads. A principal maneira de averiguar a solução realizada através do programa foram a execução de um programa semelhante à solução leitor escritor porem com prioridade balanceada para ambos as Threads o que fez com que erros pudessem ser entendidos e melhor tratados averiguando uma melhor correteude para a solução do problema.

2 Aplicação de monitoramento de temperatura

Podemos descrever como sendo as principais estruturas de dados utilizados sendo Estrutura e Queue, ambos estando em escopo global. São passados os Ids através do pthread_create como argumento estes sendo as únicas variáveis passadas da main diretamente para as Threads , tendo outro fator importante

no programa sendo o não retorno de valores pelo join da main porém printado valores diretamente das Threads no terminal. Os algoritmos utilizados pelas Threads Leitor podem ser descrito pela execução de entLeitor que fica com saiLeitor responsável pelas ordens de execução das Threads leitoras, em seguida e armazenado os 15 primeiros valores em um array caso este valor lido seja igual ao id da Thread e para cada valor que seja do mesmo id também se aumenta o número de leituras na Queue responsável pela leitura, em seguida os 15 valores são relidos e caso os valores sejam maiores que 35 se aumenta o valor das variáveis responsáveis e caso seja menor que 35 o valor responsável pelo disparo do alerta vermelho zera, em seguida é verificado se é atendido os valores maiores que 5 e caso sim, é disparado o alerta vermelho e amarelo caso nenhuma das condições sejam atendidas será disparado o alerta verde em seguida é calculado a média dos números presentes dentro do array principal de valores. As Threads Escritoras podem ser descritas pela utilização de entEscritor que junto a saiEscritor são responsáveis pelo controle da ordem de execução em seguida se utiliza a função sleep para simular o tempo de captação de temperatura por sensores em seguida se calcula um número aleatório que varia entre 25 a 40 caso o número seja maior que 30 ele será armazenado a Queue de valores, em seguida é captado e armazenado na mesma posição na fila de id o próprio identificador da Thread. A face apresentada demonstrada e simples e acaba por demonstrar os requisitos do programa e algumas outras informações pertinentes com relação ao programa sendo elas os valores que são captados pela Thread escritora além da quantidade de vezes em que a Thread captou valores já nas Threads leitoras e exibido o valor de alerta sendo ele Vermelho, amarelo ou verde além da média dos valores armazenados na fila principal. Os principais testes realizados foram através da compilação de um segundo programa ao qual o padrão adotado foi o leitor escritor no qual não tinha prioridade para nenhuma das partes.

3 Discussão

Os resultados apresentados condizem com o definido pelo trabalho. As principais dificuldades apresentadas foram principalmente a interpretação dos requisitos dos problemas e assim sua posterior desenvolvimento de solução.

4 Referências bibliográficas

Listar as referências bibliográficas utilizadas.