

Controle e supervisão de um sistema de caldeira simulado

Daniel Meneses da Rocha¹ Francisco Silvan Felipe do Carmo² Joaquim Walisson Portela de Sousa³

Resumo—Com os conhecimentos adquiridos sobre como funciona um sistema em tempo real, foi proposto o uso e implementação de algoritmos que solucionassem os problemas apresentados. A simulação de uma caldeira foi feita com a disponibilidade de uma interface gráfica desenvolvida em Java que recebe certos parâmetros. Em seguida usando a linguagem C foi desenvolvido um controle e supervisão de um sistema de caldeira simulado através de sensores e atuadores sujeitos a requisitos temporais, entre eles o controle de temperatura no interior da caldeira sendo limitado a uma faixa de temperatura de referência, e o controle de nível de líquido sendo limitado a uma altura pré-definida, definida pelo usuário usando terminal.

Index Terms—tarefas, escalonador, sensores, temperatura, requisitos.

I. INTRODUÇÃO

SOFTWARE de tempo real pode ser entendido como sistemas de computador com requisitos de tempo real, que são uma categoria especial de sistemas operacionais. É um sistema onde é essencial a confiabilidade e a execução de tarefas que acontece em virtude de determinados eventos, com um tempo de resposta predefinido, ou seja, ele está ligado ao meio físico tendo que responder a estímulos do ambiente. Sua principal característica é que suas rotinas de processamento geral do sistema devem ser bastante especializadas e curtas, pois as tarefas devem ser feitas no menor tempo possível, podendo ter prejuízos caso o sistema não seja confiável e aconteça alguma falha.

No sistema em tempo real, a sincronização e a comunicação entre tarefas podem causar inconsistência de dados e mudança de prioridade entre tarefas mal implementadas. Para fragmentos de código em que uma tarefa acesse algum recurso compartilhado entre várias tarefas, uma seção crítica deve ser definida. A fim de evitar esses conflitos, é utilizado o mecanismo Mutex, que é um tipo abstrato de dados cujos atributos são um valor lógico e uma fila de threads bloqueadas. O valor lógico registra o fato de que a cada momento, o mutex pode estar em estado livre ou ocupado, já a fila de threads contém todas as threads bloqueadas esperando a liberação deste mutex, tendo assim uma técnica de sincronização que garante acesso exclusivo a uma tarefa.

A ideia básica por trás do monitor é que, na maioria dos programas, a maior parte do código é sequencial, referindo-se ao comportamento independente e paralelo de cada tarefa.

Porém, em algumas poucas situações, as tarefas precisam interagir entre elas, dando origem a situações de exclusão mútua e outras mais complexas [1].

Por fim o trabalho apresenta uma simulação de um sistema de tempo real para o controle e supervisão de uma Caldeira, utilizando as técnicas citadas acima como o mutex e o monitor, visando organizar as tarefas na utilização de threads, onde dois ou mais processos não poderão usar uma mesma variável e assim ocorrer problemas. Nessa Caldeira os sensores analisam a temperatura buscando sempre verificar o seu limite crítico para evitar superaquecimento, já os atuadores verificam o fluxo e o nível da água além de sua temperatura, podendo assim receber dados do sistema. Ao final da análise é gerado um arquivo com todos os dados obtidos durante a execução do sistema.

II. METODOLOGIA

A organização desse trabalho consiste em solucionar o problema apresentando no Sistema de Controle e Supervisão no Linux presente no documento de descrição, no qual ocorre a implementação e supervisão de um sistema em tempo real de uma Caldeira.

A. Sistema de Controle e Supervisão no Linux

O sistema busca sincronizar o controle de temperatura e o nível da caldeira através de uma interação entre eles, sendo necessário algumas lógicas de programação para esse controle ser em sincronia. Esse método de sincronização e comunicação entre as tarefas é feito pelo uso de variáveis compartilhadas, assim a base do sistema é em tarefas que realizam a comunicação através de variáveis globais compartilhadas. Essas variáveis globais são acessadas por várias threads, sendo possível a transação dos dados, assim as threads dividem as variáveis globais enquanto suas variáveis locais do programa são devidamente alocadas em sua determinada função atendendo os requisitos. Implementando a linguagem C no Linux, o sistema aplica alguns requisitos que utilizam o conceito de threads buscando a interação entre as tarefas e compartilhando dados entre elas. Os requisitos foram feitos da seguinte forma:

1) *Requisito*: Foi criada uma tarefa periódica (thread) para o controle de temperatura, com um período de 50ms usando as funções “clock_gettime e clock_nanosleep”. O algoritmo irá ler as variáveis e depois analisar a temperatura, aumentando ou diminuindo se necessário em conjunto com o atraso de tempo, além de ser uma tarefa com prioridade máxima diante do escalonador do sistema.

A. 1, Universidade Federal do Ceará, Sobral, Brasil e-mail: danielmenes@alu.ufc.br.

A. 2, Universidade Federal do Ceará, Sobral, Brasil, e-mail: silvanfelipe@alu.ufc.br.

A. 3, Universidade Federal do Ceará, Sobral, Brasil, e-mail: joaquimwalisson@alu.ufc.br.

2) *Requisito:* Foi criado uma tarefa periódica (thread) para o controle de nível da água, com período de 70ms usando as funções “clock_gettime e clock_nanosleep”. Será verificado se o nível da caldeira está alto ou baixo e também a temperatura, disso ele irá regular o fluxo de água, além de ser uma tarefa com prioridade máxima diante do escalonador do sistema.

3) *Requisito:* Usou-se os atuadores Ni, Q, Na, Nf nas tarefas de controle, para verificar o fluxo através do monitoramento da temperatura.

4) *Requisito:* Foi criada uma tarefa para mostrar informações correntes na tela (Terminal) sobre os sensores Ta, T, Ti, No e H, através da função “thread_mostra_status”.

5) *Requisito:* Realizou-se a verificação da temperatura a cada 10ms através da leitura dos sensores e foi criada a tarefa de alarme para disparar caso esteja acima de 30 graus.

6) *Requisito:* Criou-se a tarefa “thread_ler_usuario” para alterar através do teclado os valores de referência do nível e temperatura, recebendo esses dados do usuário.

7) *Requisito:* Foi criada a tarefa para armazenar em arquivo .txt os tempos de respostas da tarefa periódica do item 1, através de um buffer duplo (produtor/consumidor).

8) *Requisito:* Criou-se uma tarefa para armazenar a cada segundo os valores dos sensores de nível (H) e temperatura (T) e depois de algum tempo foi gravado em um arquivo.

B. Medições em Tempo Real

Logo a seguir ficam as medições em tempo real obtidas da simulação da caldeira, para realizar análises, fazer possíveis mudanças para melhorar o experimento e tornar a caldeira mais eficiente.

1) *Tópico:* Obter o tempo mínimo e médio da tarefa periódica do controle de temperatura em razão do seu tempo de resposta.

2) *Tópico:* Plotar um gráfico para mostrar as medidas realizadas na ordem dos casos de teste.

3) *Tópico:* Plotar um gráfico histograma para identificar o pior caso do tempo de resposta observado, conhecido como (HWM - High Water Mark).

III. RESULTADOS E DISCUSSÕES

Com a realização dos testes na simulação da caldeira foram obtidos os seguintes gráficos que representam o comportamento do sistema em tempo real no experimento. Durante o teste o limite de temperatura foi definido para 29°, o limite de altura foi 1,5 m, a temperatura ambiente de 25° e a temperatura inicial da simulação em 20°. O código foi executado em um desktop com sistema operacional Windows por meio do WSL e gcc.

Na medição do tópico 1, após suscetíveis testes no experimento, foi obtido como resposta para o tempo mínimo o valor de 1861 ms e para o tempo médio foi o valor de 1.9027e+04 ms.

Para medição do tópico 2 foi gerado um gráfico que mostra as medidas dos tempos de resposta nos casos de teste dado pela figura 1:

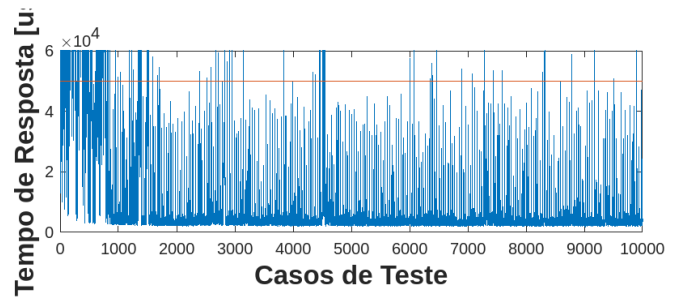


Figura 1. Gráfico com as medições realizadas, na ordem dos casos de teste.

Para a medição do tópico 3 foi gerado um gráfico que mostra o pior caso do tempo de resposta dado pela figura 2:

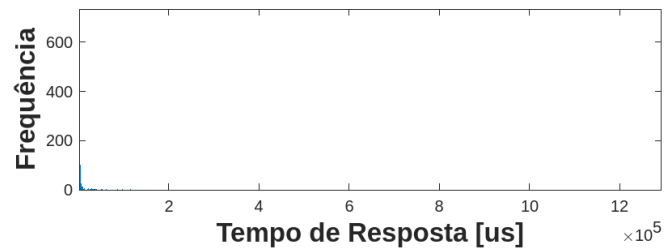


Figura 2. Gráfico que mostra o pior caso do tempo de resposta.

REFERÊNCIAS

- [1] Oliveira, R. S. Fundamentos dos Sistemas de Tempo Real. Original registrado na Biblioteca Nacional. Primeira edição, revisão 3, outubro de 2018. ISBN-13: 9781728694047.