

Controle e supervisão de um sistema de caldeira simulado

Pedro Cézar Rodrigues Baltazar (402085) Sergio Franco Sales Filho (414596)

Resumo—Utilizando um simulador feito em java de uma caldeira controlada via software, foi possível fazer um programa em C que controlasse o sistema dessa caldeira, utilizando, principalmente, a biblioteca *pthread*, que com ela é possível criar *threads* do zero, fazer controle de acesso de variáveis com *mutex* (implementado por meio de monitores), fazer medições de tempo de execução etc. Basicamente, na caldeira, há sensores e atuadores que podem ser lidos e modificados utilizando o protocolo de comunicação exigido pelo sistema.

Index Terms—*Thread*, caldeira, *mutex*, C, atuadores, sensores.

I. INTRODUÇÃO

SISTEMAS de Tempo Real (STR) são sistemas que são elaborados com o propósito secundário, além do propósito principal do projeto, de que suas tarefas ou funções sejam executadas dentro de certas requisições de tempo, como de período, de *deadline* e de tempo de execução. Assim, torna-se necessário que o sistema atenda a todas essas exigências, onde, muitas vezes um mesmo processador/memórias são utilizados para a execução de todo o projeto, que, por sua vez, também podem concorrer por esses recursos.

Portanto, é implementada a idéia do *mutex*, que tem a função de bloquear posições da memória especificadas no programa, para que outras *threads* que concorrem com esta variável não possam acessá-las e causar possíveis problemas de cálculos que dependem do valor desta variável. E, para a implementação do *mutex*, foram utilizados monitores, que, basicamente, são mecanismos de sincronização de *threads* que são implementados no programa na forma de módulos, onde todos os problemas relacionados com o acesso de variáveis compartilhadas acontecem apenas dentro dos monitores, que garantem a segurança no momento de acesso dessas variáveis.

Com isso, foi possível elaborar um programa em C que colhesse informações a respeito do estado de funcionamento de uma caldeira (em simulação), bem como, enviasse valores de vazão (kg/s) e calor (J/s), dentre outros, para o controle deste estado de funcionamento. Assim, este trabalho pretende demonstrar a utilização da biblioteca *pthread* (linguagem C) na elaboração de *threads* e *mutex* para controlar determinados parâmetros de funcionamento de um sistema de aquecimento (caldeira), onde todos os processos acontecem dentro dos monitores, contando com a comunicação (coleta e envio) entre o software e o sistema de atuadores e sensores da caldeira, com os cálculos feitos relativos ao controle de temperatura, nível e vazões de água, com a disponibilização das informações na

tela e coleta da entrada pelo usuário do software e, por fim, com o registro das informações obtidas durante a execução do programa em um arquivo de *log*.

II. METODOLOGIA

A. Sistema da Caldeira

Para o sistema da caldeira, utilizando o software de simulação de aquecimento de uma caldeira, que possibilita uma comunicação com o programa via LAN, utilizando o protocolo de comunicação entre caldeira e programa, onde o programa pode enviar um pedido de valor de medição de um sensor ou enviar um valor para um atuador de acordo com o protocolo correto. Para realizar a comunicação entre o simulador e o programa, iniciou-se o programa em linha de comando como `java -jar Aquecedor2021.jar <porta >`. Já para o início do programa criado, após a compilação, foi executado o comando `./controle-caldeira <ip-local-do-simulador> <porta>`.

Da caldeira, é possível ler os sensores T_a (temperatura ambiente), T (temperatura na caldeira), T_i (temperatura da água não controlada de entrada do atuador), N_o (temperatura da água de saída) e H (nível de água na caldeira). E, para o controle da caldeira, é possível configurar valores para os atuadores N_i (vazão de água de entrada de temperatura não controlada), Q (calor gerado pelo aquecedor), N_a (vazão de entrada de água a $80^\circ C$) e N_f (fluxo de água de saída para o esgoto).

Assim, com esses valores obtidos pelos sensores e a possibilidade de alteração dos atuadores, foi possível configurar a caldeira para alcançar valores específicos de temperatura (T) e de nível de água (H), utilizando equações que relacionam essas grandezas relativas ao sistema da caldeira.

Para o controle da temperatura na caldeira (T) e do nível de água (H), foi pedido como entrada, no início da execução do programa (uma única vez), valores de temperatura de referência e de nível de referência, respectivamente, inseridos pelo usuário.

B. Medições de Tempo Real

Já para a parte de medições de tempo real, que variam devido ao design do software (programação concorrente), outras atividades do sistema e devido ao *hardware*, foi medido o tempo de resposta, com mais de 10.000 amostras, da tarefa de controle de temperatura da caldeira, onde o N_o foi variada mais de cinco vezes pela interface do software do simulador. Para a medição do tempo de resposta, foi utilizado o relógio interno do sistema operacional antes e depois do início da

A. 1, Universidade Federal do Ceará, Sobral, Brasil

A. 2, Universidade Federal do Ceará, Sobral, Brasil

execução da *thread* e, para armazenar os resultados, foi utilizado um *buffer* duplo (produtor/consumidor) que guarda os seus valores armazenados em um arquivo *.txt*, que posteriormente, foi utilizado no software *MatLab* para um estudo estatístico detalhado do tempo de execução desta tarefa, como será visto na próxima seção.



Autores Universidade Federal do Ceará (UFC) -
Campus Mucambinho

III. RESULTADOS E DISCUSSÕES

Do sistema da caldeira, foi observado que existe um certo intervalo de tempo até os valores de temperatura e de nível chegarem aos seus valores de referência, isso pode ser explicado dado que a temperatura da caldeira e o nível de água estão relacionados de uma forma que quando qualquer uma das *threads* (controle de temperatura ou controle de nível) entram em ação, podem prejudicar ou ajudar a atuação da outra.

Já para a parte das medições de tempo real, foi possível obter o resultado de mais de 10.000 medições. Esse resultado foi utilizado no *MatLab* e foi feito dois gráficos: um gráfico de dispersão das amostras (na ordem dos casos de teste) e um gráfico de histograma das medições realizadas, como é possível observar na figura abaixo.

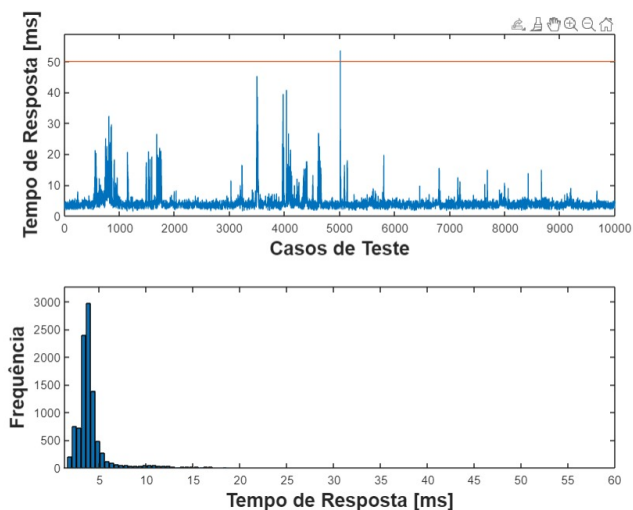


Figura 1. Gráfico com as medições realizadas, na ordem dos casos de teste.

De acordo com a figura, é possível visualizar o valor médio do tempo de resposta de todos os casos de teste (por volta de 50ms) e, também, no histograma, que a maior parte dos tempos de medição não passaram de 10ms , onde, a maior parte dos casos estão próximos dos 4ms de tempo de execução.

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.