



## Trabalho 1

# 1 Descrição Geral

A equipe deve implementar o controle e supervisão do sistema de tempo real descrito no PDF 'controle-caldeira-2'. Ver a Figura 1.

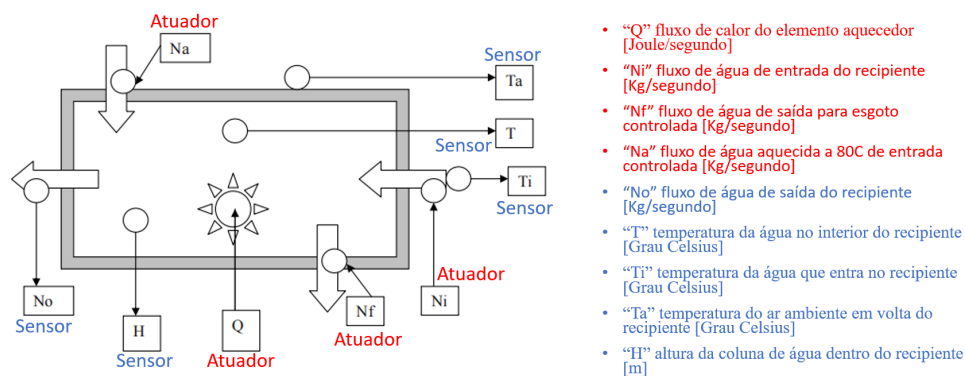


Figura 1: Sistema de Caldeira.

Um Simulador desse sistema será disponibilizado para interagir com o software em tempo real desenvolvido pelo aluno neste trabalho.

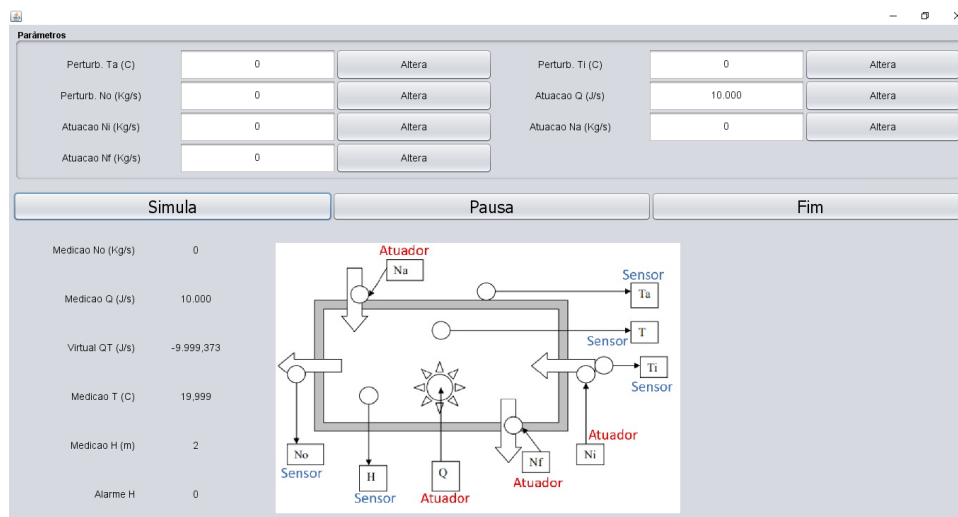


Figura 2: Simulador em Java.

Para executar o simulador, basta executar o seguinte comando:

```
java -jar aquecedor2008_1.jar <número-porta-escutada>
```

Caso o usuário execute o simulador dando dois cliques no arquivo, a porta padrão é 4545.

A caldeira possui instrumentação embutida e aceita os seguintes comandos:

```
"sta0" lê valor de Ta
"st-0" lê valor de T
"sti0" lê valor de Ti
"sno0" lê valor de No
"sh-0" lê valor de H
"ani123.4" define valor de Ni como 123.4
"aq-567.8" define valor de Q como 567.8
"ana123.4" define valor de Na como 123.4
"anf123.4" define valor de Nf como 123.4
```

Cuidado com a formatação dos valores em ponto flutuante.

## 2 Requisitos do Sistema

Implementar em C no Linux um programa para controlar a o sistema de caldeira, o qual deve incluir as seguintes funcionalidades de controle:

1. Criar uma tarefa periódica para o controle de temperatura. O período é de 50ms.
2. Criar uma tarefa periódica para o controle de nível da água. O período é de 70ms.
3. Usar os atuadores Ni, Q, Na, Nf nas tarefas de controle;
4. Criar uma tarefa para mostrar informações corrente na tela (Terminal) sobre os sensores Ta, T, Ti, No e H;
5. Sinal de alarme caso o sistema esteja acima de 30 graus.
6. Usar a entrada através do teclado para definir os valores de referência do nível e temperatura;
7. Armazenar os tempos de respostas da tarefa periódica do item 1 em arquivo, através de um buffer duplo (produtor/consumidor).

### Observações:

- Use o arquivo “controlemanual.c” disponibilizado pelo professor como base do programa .
- As tarefas devem ser criadas usando Threads.
- As tarefas periódicas de controle de temperatura, nível devem ser implementadas com precisão usando as funções *clock\_gettime* e *clock\_nanosleep*. Ver arquivo exemplo “tarefaperiodica1.c”
- Para o cálculo do tempo de resposta de um tarefa periódica veja o exemplo “tarefaperiodica3.c”.
- A tarefa para mostrar informações na tela pode ser implementada com *sleep* simples de 1s.

- Use monitor(es) para proteger os dados dos sensores. Ver exemplo “sensor.c” e “sensor.h”
- O alarme de temperatura pode ser implemetado usando variável de condição em monitor. Veja o exemplo “sensor.c”.
- Use um monitor buffer duplo para armazenagem de dados em arquivo sobre os tempos de respostas da tarefa periódica do controle de temperatura. Ver exemplo “bufduplo.c”.

### 3 Medições de tempo Real

O tempo de resposta de um sistema de tempo real varia devido o design do software e outras atividades do sistema. Desta forma, fazer uma única medição do tempo de resposta não mostra a gama de valores possíveis. São necessárias várias medições, e uma visão estatística destas medições. Para mais informações sobre uma Visão Estatística das Medições de tempo real, leia a Seção 14.13 do livro Fundamentos dos Sistemas de Tempo Real (1ª Edição) (Página 283), ou a Seção 16.2 do mesmo livro na 2ª Edição (Página 335).

Realize um teste do sistema projetado com um tempo suficiente para coletar um número de amostras  $\geq 10.000$  dos tempos de respostas da tarefa de controle de temperatura. No teste o *No* deve ser variado pelo menos 5 vezes, nesse caso fica a cargo da equipe definir os intervalos de tempo dessas variações até a coleta total das amostras. Para realização dessa parte do trabalho o requisito 7 precisa ser implementado. Com os dados coletados faça as seguintes análises estatísticas.

1. Obtenha o tempo mínimo e médio do tempo de resposta da tarefa periódica de controle de temperatura.
2. Obtenha o tempo de resposta máximo observado, também conhecido como **HWM** (do inglês **High Water Mark**, altura máxima da água). Em muitos casos o HWM acrescido de uma margem de segurança é usado como estimador do tempo de resposta no pior caso (WCRT – Worst-Case Response Time) e é comparado com o deadline da tarefa, para fins de verificação do cumprimento dos requisitos temporais. A margem de segurança pode ser determinada pela diferença entre o valor máximo e mínimo do tempo de resposta. Diante dos dados coletados é possível afirmar que o deadline da tarefa de controle de temperatura foi suficiente? Se não foi, quais fatores podem ter contribuído para não ser suficiente? Sistema Operacional? Hardware? Lógica da programação? Plote um gráfico, mostrando as medidas realizadas, na ordem dos casos de teste. Veja o exemplo de gráfico a ser plotado.

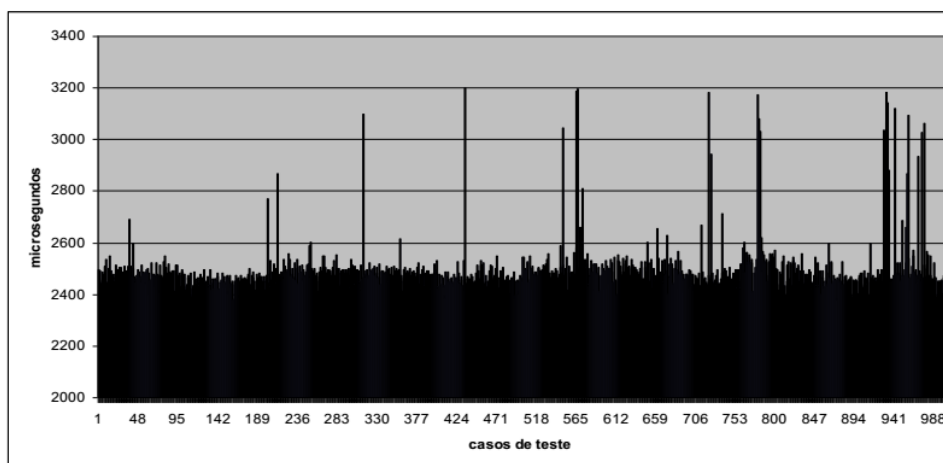


Figura 3: Gráfico com as medições realizadas, na ordem dos casos de teste.

3. Plote um gráfico histograma e verifique a porcentagem de amostras que cumpriram o deadline definido.

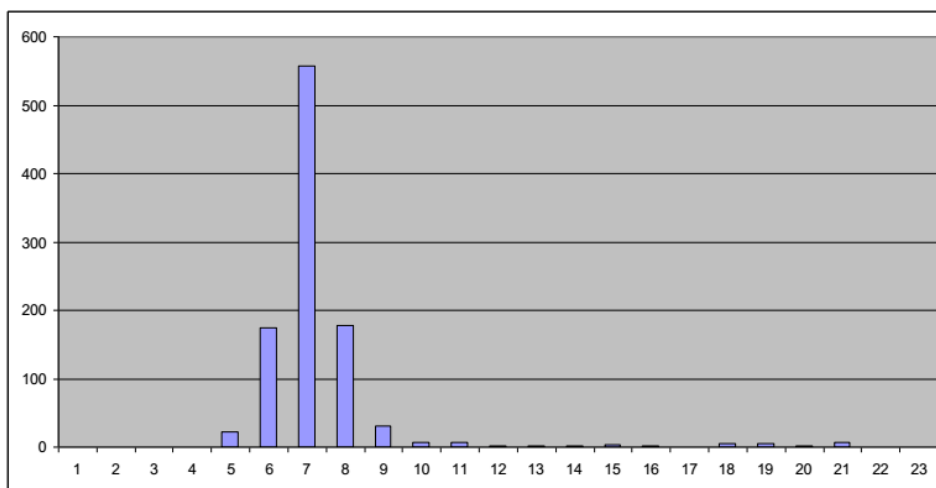


Figura 4: Histograma das medições realizadas.

4. Observe se a tarefa de controle de temperatura tem Fator Skip 20. Uma tarefa tem fator skip  $S$  quando a distância entre duas perdas de deadlines é de no mínimo  $S$  ativações. Plote um gráfico mostrando apenas as medições onde o deadline foi perdido. Se o sistema não tiver fator skip 20, qual fator ele tem? Veja o exemplo de gráfico a ser plotado a seguir:

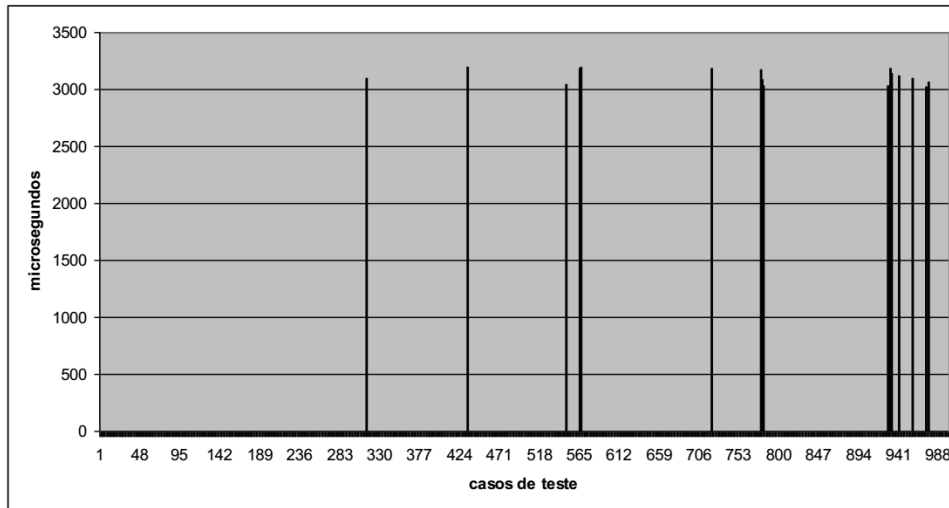


Figura 5: Gráfico mostrando apenas as medições onde o deadline foi perdido.

5. Uma tarefa é  $(m,k)$ -firme se ela sempre cumpre “ $m$ ” deadlines em qualquer janela de “ $k$ ” ativações consecutivas. Necessariamente temos  $m \leq k$ . Caso  $m = k$ , então todos os deadlines precisam ser cumpridos. Observe uma janela de 20 ativações consecutivas dos dados coletados das duas tarefas. Identifique o pior momento e com isso determine qual valor de  $m$  para  $(m,20)$ -firme. Plote um gráfico para cada tarefa mostrando número de deadlines perdidos em janela de 20 ativações. Veja o exemplo a seguir de gráfico a ser plotado.

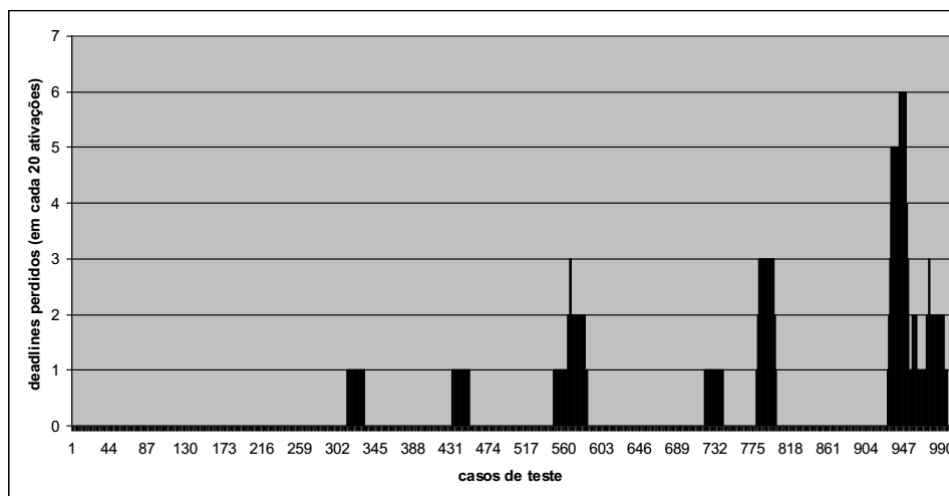


Figura 6: Gráfico mostrando número de deadlines perdidos em janela de 20 ativações.

6. Gerar relatório em PDF contendo os gráficos e a descrição das observações.

## 4 Informações Gerais

- Trabalho em grupo de até 4 pessoas;
- Data de entrega: 16/01/2022;
- Nota: 10 pontos (Parte Prática da AP2).

- A avaliação do trabalho se dará mediante apresentação do código implementado e do relatório elaborado pela equipe. A composição da nota seguirá a seguinte tabela:

Tabela 1: Tabela de avaliação

Item	Nota
Requisitos do Sistema (7 requisitos)	5.0
Medições de tempo real (5 análise estatística)	4.0
Organização do relatório	1.0
Total	10.0

Bom trabalho a todos!