



Trabalho 2

1 Descrição Geral

A equipe ou aluno deve implementar o controle e supervisão do sistema de tempo real descrito no PDF 'controle-caldeira-2'. Um Simulador desse sistema será disponibilizado para interagir com o software em tempo real desenvolvido pelo aluno neste trabalho. 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 14545.

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.

O Trabalho será dividido em três partes: Implementar requisitos e realizar teste comportamental; Visão estatística das medições de tempo Real; Elaboração de documento no formato de artigo para demonstrar todo o desenvolvimento do trabalho. A seguir será descrito como deve proceder em cada parte do trabalho.

2 Parte I - Implementar requisitos e realizar teste comportamental

2.1 Requisitos do sistema

Implementar em C no Linux o programa CONTROLADOR, 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. Tarefa para verificação da temperatura a cada 10ms para disparo de tarefa de alarme caso esteja acima de 30 graus.
6. Entrada através do teclado dos valores de referência para nível e temperatura;
7. Armazenagem periódica dos valores lidos dos sensores T, No e H e armazenagem desses dados em arquivo, através de um buffer duplo (produtor/consumidor).
8. Armazenagem periódica dos tempos de respostas das tarefas periódicas de controle de temperatura e nível, e armazenagem desses dados 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 e disparo de alarme 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 uma tarefa periódica veja o exemplo “tarefaperiodica3.c”.
- A tarefa para mostrar informações na tela pode ser implementada com *sleep* simples de 1s.
- Use um monitor para proteger o recurso de *Socket* usado na comunicação.
- Use monitor(es) para proteger os dados dos sensores. Ver exemplo “sensor.c” e “sensor.h”
- O alarme de temperatura pode ser implementado usando variável de condição em monitor. Veja o exemplo “sensor.c”.
- Use um monitor buffer duplo para armazenagem de dados em arquivo sobre os sensores. Ver exemplo “bufduplo.c”
- Use um monitor buffer duplo para armazenagem de dados em arquivo sobre os tempos de respostas das tarefas periódicas. Ver exemplo “bufduplo.c”.

2.2 Realizar teste comportamental

Busque simular o comportamento do sistema em tempo real e examine seu comportamento em consequência de eventos externos. Faça os seguintes testes:

1. Teste 1
 - Simulador setado em Ti=10, No=0 e Ta=10
 - Controle de temperatura: 25 °C

- Controle de altura: 2m
- Colha os dados do comportamento do sistema em relação ao tempo até o controle desejado. Grave num arquivo usando o monitor buffer duplo os valores dos sensores T, No e H.
- Crie os gráficos de (Tempo x Temperatura) e (Tempo x altura) dos dados coletados

2. Teste 2

- Simulador setado em $T_i=20$, $T_a=25$, $N_o=10$ nos primeiros 30s, depois $N_o=70$ nos próximos 30s, depois $N_o=100$ nos próximos 40s e finalmente $N_o=0$ nos próximos 80s.
- Controle de temperatura: $29\text{ }^{\circ}\text{C}$
- Controle de altura: 1.5m
- Colha os dados do comportamento do sistema em relação ao tempo até o controle desejado. Grave num arquivo usando o monitor buffer duplo os valores dos sensores T, No e H.
- Crie os gráficos de (Tempo x Temperatura x No) e (Tempo x altura x No) dos dados coletados.

3 Parte II - Visão estatística das 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 (Página 283).

Ref faça o Teste 2 do ponto (2.2) estendendo o tempo do teste para coletar um número de amostras ≥ 10.000 dos tempos de respostas das tarefas de controle de temperatura e nível. No teste 2 o No foi variado três vezes, nesse caso fica a cargo da equipe definir os intervalos de tempo dessas três variações. Para realização dessa parte do trabalho o requisito 8 do ponto (2.1) precisa ser implementado. Com os dados coletados faça as seguintes análises estatísticas.

1. Obtenha o tempo mínimo e médio do tempo de resposta das tarefas periódicas de controle de temperatura e nível.
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 os deadlines 50ms e 70ms determinados nos requisitos 1 e 2 do ponto (2.1) são suficientes? Se não foram, quais fatores podem ter contribuído para não

serem suficientes? Sistema Operacional? Hardware? Lógica da programação? Plote um gráfico para cada tarefa, mostrando as medidas realizadas, na ordem dos casos de teste. Veja o exemplo de gráfico a ser plotado.

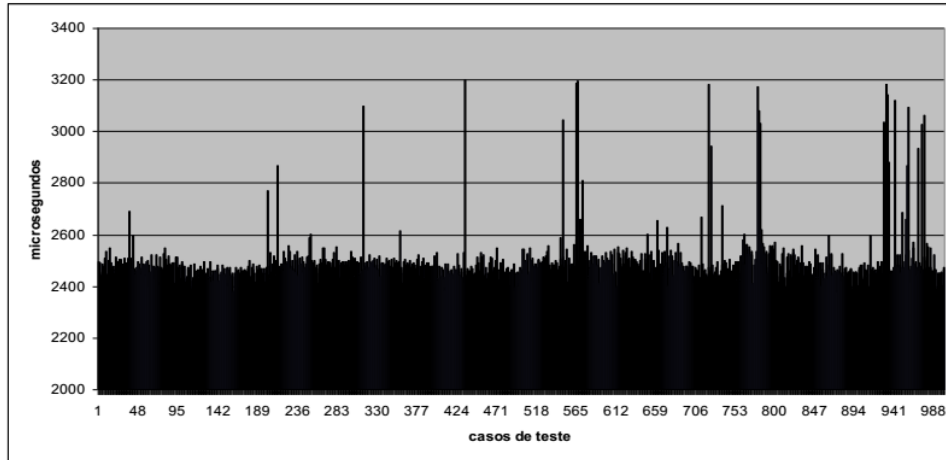


Figura 1: Gráfico com as medições realizadas, na ordem dos casos de teste.

3. Plote um gráfico histograma e verifique a porcentagem de amostras que cumpriram o deadline definidos nos requisitos. Se considerar apenas 98% das amostras qual seria o HWM? Qual seria o HWM(95%)? Plote o histograma para HWM(98%) e HWM(95%) também. Veja a seguir um exemplo de histograma a ser plotado.

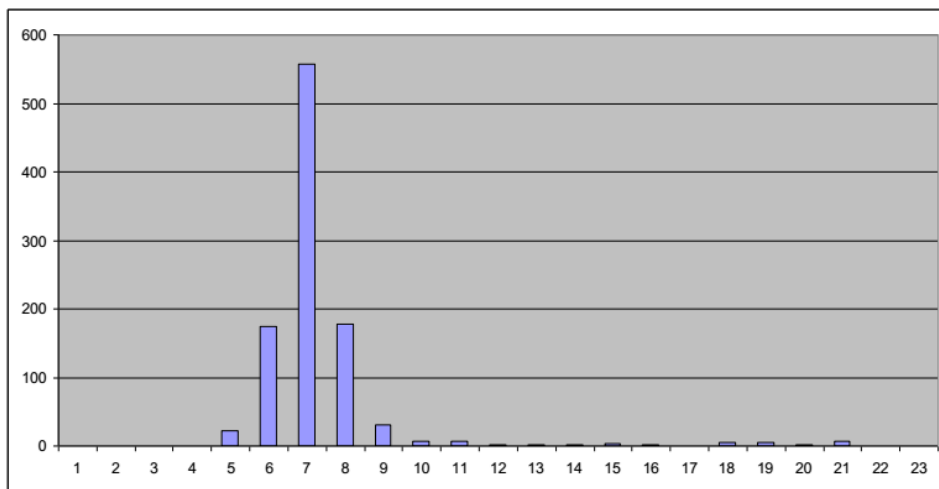


Figura 2: Histograma das medições realizadas.

4. Observe se as tarefas de controle de temperatura e controle de nível 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 para cada tarefa 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:
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

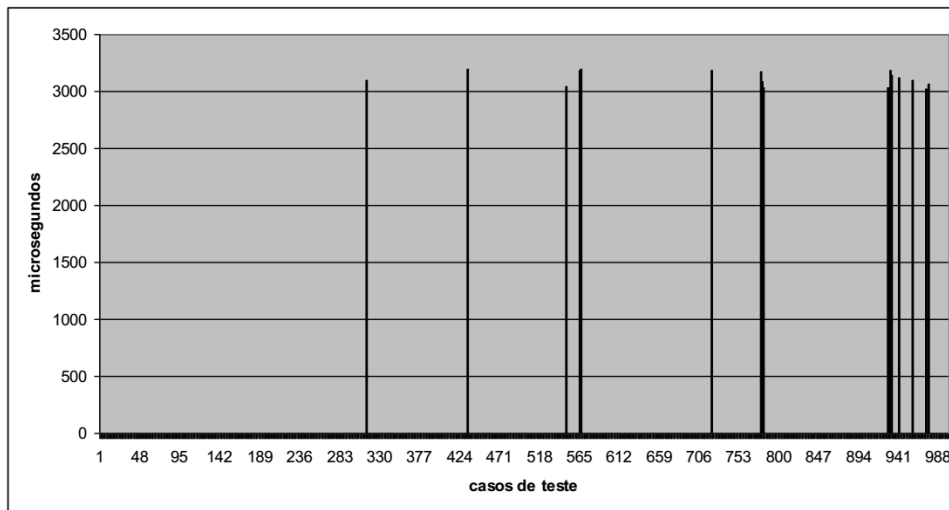


Figura 3: Gráfico mostrando apenas as medições onde o deadline foi perdido.

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.

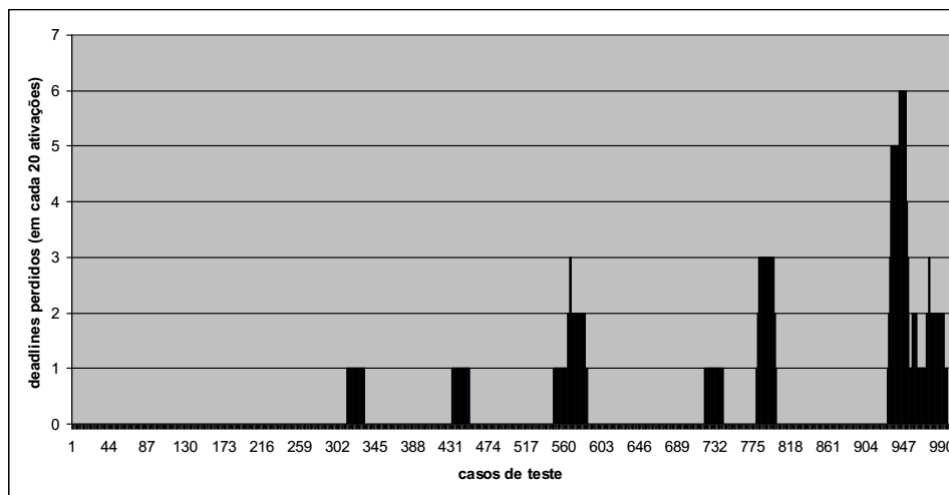


Figura 4: Gráfico mostrando número de deadlines perdidos em janela de 20 ativações.

4 Parte III - Elaboração do documento em formato de artigo

A equipe deverá elaborar um documento em formato de artigo descrevendo todo o processo de desenvolvimento do trabalho. O documento terá que ser feito em latex usando o modelo disponibilizado pelo professor. O artigo deve ter no mínimo 3 páginas e no máximo 6 páginas.

5 Informações Gerais

- Trabalho individual ou em dupla;

- Data de entrega: 01/09/2020;
- Nota: 6 pontos na 2ª Avaliação parcial.
- A avaliação do trabalho se dará mediante apresentação do código implementado e do documento elaborado pela equipe. A composição da nota seguirá a seguinte tabela:

Tabela 1: Tabela de avaliação

Item	Nota
Requisitos do Sistema (8 requisitos)	3.0
Realizar teste comportamental (2 testes)	3.0
Visão estatística das medições de tempo real (5 análise estatística)	3.0
Organização do artigo	1.0
Total	10.0

OBSERVAÇÃO: Os requisitos 7 e 8 são essenciais para fazer a parte dos testes comportamentais e a visão estatística das medições de tempo real. Portanto, não fazê-los implicará em uma nota baixa.

Bom trabalho a todos!