Sérgio Venturi Pereira, Thiago Zipper Melato e Fabio Ivo

Trabalho de Sistemas em Tempo Real (STR) - Concorrência e multithread em STR

Este relatório contém a implementação de um algoritmo de simulação da rede CAN de um carro, utilizando threads.

Universidade do Vale do Itajaí - UNIVALI
Engenharia da Computação
Sistemas em Tempo Real

Orientador: Felipe Viel

Brasil – 2023

Sumario

Introdução

O seguinte trabalho tem por objetivo simular discretamente a rede CAN (Controller Area Network) de um carro, esta rede serve para interligar centrais de comando de diferentes funcionalidades, como motor, câmbio, conveniência, segurança etc.

Visto que alguns destes sistemas possuem grande importância e não podem sofrer com atrasos, como o sistema de airbags, é preciso que haja uma prioridade no processamento e exibição das informações.

Todos estes sistemas precisam se comunicar entre si via rede CAN, mas também precisam fornecer algumas dessas informações ao condutor, isso é feito por meio do computador de bordo ou pela central multimidia em alguns carros.

Essa exibição também exige um tempo de resposta em tempo real, visto que em alguns casos como de superaquecimento do motor o condutor precisa tomar medidas imediatas, ou quando o sensor frontal detecta um risco de colisão esta informação precisa ser repassada para o computador de bordo sem atraso.

Enunciado

A problemática deste trabalho consiste na simulação de um novo sistema de monitoramento do comportamento de várias áreas de um veículo. Sendo esses: Motor, Frenagem, Equipamentos de Suporte a Vida e Luzes, Vidros e Travas (LVT).

Cada sistema possui alguns sensores que devem ser monitorados, o sistema do motor deve controlar a injeção eletrônica e temperatura. Já na frenagem deve ser controlado apenas o ABS das duas rodas dianteiras. Para os equipamentos de suporte a vida é necessário controlar um airbag e cinto de segurança. E por fim o sistema de LVT controla as luzes dos faróis dianteiros, vidros elétricos dianteiros e as travas das portas dianteiras.

Todos estes sistemas devem se conectar ao computador de bordo que deve controlar estes sistemas e exibir suas informações. Para fins de simulação será atribuída uma tecla para executar as ações que deveriam ser obtidas por sensores, como uma colisão, pedal de freio, pedal do acelerador.

É necessário observar que a propagação das informações pelos fios leva 10us e deve-se manter dentro das deadlines necessárias para cada sistema, sendo elas:

- Injeção eletrônica: 500 us após alteração no pedal
- Temperatura do motor: 20 ms após detecção de temperatura acima do limite
- ABS: 100 ms após acionamento no pedal
- Airbag: 100 ms após detecção de choque
- Cinto de segurança: 1 segundo após carro em movimento
- LVT: 1 segundo após interação do usuário

Implementação

Para implementar o sistema solicitado foi optado por separar cada subsistema em um arquivo diferente e mais adiante criar uma thread para cada um. Sendo assim foram criadas 6 threads, sendo elas:

Lvtsystem
Brake_system
Motor_system
Life_support_equipment
Data_reading
Data_print

Uma thread para cada subsistema, e as outras duas fazem a função de ler e escrever as informações em tempo real.

Para veicular os dados foi criada uma string de valores binários onde cada byte representa o estado de cada sensor. Após o acionamento de um sensor o sistema responde escrevendo na string o novo estado do sistema e a cada 500 ms o computador de bordo atualiza e escreve na tela o estado de cada sensor.

No quesito da obtenção do tempo necessário para o sistema responder às ações, foi usado o método gettimeofday(), essa função é ideal para monitorar o tempo entre threads diferentes, visto que a função de clock() não funciona de modo ideal quando utilizado em mais de uma thread.

Desempenho

Os testes de desempenho foram baseados no tempo de resposta para cada subsistema reagir após a tecla que o aciona for clicada. Inicialmente foi testado o tempo de reação de cada subsistema individualmente, por sequência foi testado o tempo de reação quando todos os sensores são alterados simultaneamente.

Conclusão

Durante o processo de implementação do código foi possível compreender o funcionamento da rede CAN e simular um sistema parecido, possuindo funcionalidades de leitura, transmissão, controle e escrita de dados entre diferentes threads paralelas.

Pode-se observar que devido ao fato de manter o código enxuto e eficiente foi possível se manter dentro das deadlines solicitadas, tanto quando os sensores foram testados individualmente quanto no pior caso, onde tudo acontece simultaneamente.

Referencias

PEREIRA, Sergio Venturi; MELATO, Thiago Zipper; IVO, Fabio. **TRABALHO de Sistemas em Tempo Real (STR) – Concorrência e multithread em STR.** [*S. l.*], 23 out. 2023.
Disponível em: https://github.com/sergio-venturi/Trabalho-M2/tree/main
Acesso em: 23 out. 2023.

Getting the Time (The GNU C Library).
Disponível em: https://www.gnu.org/software/libc/manual/html_node/Getting-the-Time.html
Acesso em: 23 out. 2023.

Integers (The GNU C Library).
Disponível em:
https://www.gnu.org/software/libc/manual/html_node/Integers.html
Acesso em: 23 out. 2023.