

Universidade do Vale do Itajaí
Curso: Engenharia de Computação
Disciplina: Sistemas em Tempo Real
Professor: Felipe Viel
Avaliação M2 – Concorrência e multithread em STR

Instruções

- Deverá ser usada as linguagens C/C++/Rust para o desenvolvimento.
- Não será permitido o uso de bibliotecas ou funções prontas e disponibilizadas em base de dados ou em outros meios, salvo situações expressas no trabalho.
- Os códigos desenvolvidos devem ser postados e estarem funcionais. Caso a aplicação não esteja correta, será atribuída nota proporcional ao trabalho.
- O trabalho poderá ser feito sozinho, dupla ou trio.
- O(s) aluno(s) deverá(ão) apresentar o trabalho presencialmente. Caso seja requisitado, deverão dar esclarecimentos adicionais. O não comparecimento a explicação, mesmo a adicional, implicará em nota máxima (proporcional ao desenvolvido) 5,0.
- Trabalhos copiados implicará na nota zero para todos os envolvidos.
- Não serão aceitos trabalhos entregues em atraso.
- As aplicações desenvolvidas deverão exibir resultados corretos para qualquer caso de teste possível dentro do especificado pelo enunciado.
- O código fonte deverá estar comentado para auxiliar o entendimento. A postagem deverá ser feita no github. O usuário do professor é: VielF.
- A responsabilidade pela demonstração do código funcionando é do(s) aluno(s).
- Deverá ser entregue (postado) um relatório contendo: a) Identificação do autor e do trabalho; b) enunciado de cada projeto; c) imagens, tabelas e/ou quadros demonstrando os resultados obtidos; d) explicação dos resultados obtidos; e e) descrição das técnicas utilizadas. Poderá ser usado modelo de artigo, usando o template IEEE.
- O trabalho deverá ser postado por um dos alunos no material didático.
- O trabalho deverá ser postado no material didático até as **8h00 do dia 23/10/2022**.
- Itens avaliados:
 - Qualidade do relatório
 - Apresentação, descrição do problema, utilização correta do português, profundidade de análise a partir das questões elencadas, diagramação (uso de imagens e diagrama para explicar);
 - Codificação totalmente funcional e atendimento aos requisitos.
 - Qualidade da apresentação do trabalho (presencialmente). Compilação do software no momento da apresentação é obrigatório.
 - Pontualidade.

Problemática

Uma fabricante de automóveis quer testar um novo sistema de monitoramento do comportamento de várias áreas de um veículo no qual ela está projetando. O projeto é novo e permitirá controle e noção do que está acontecendo em todo o veículo. A empresa, em um primeiro momento, dividiu o carro em 4 subsistemas: (i) motor; (ii) frenagem; (iii) equipamentos de suporte a vida; e (iv) LVT (luz, vidros e travas). No motor, deve-se monitorar e controlar a injeção eletrônica e temperatura interna. Na frenagem, deve-se monitorar e controlar o ABS nas duas rodas dianteiras. Nos equipamentos de suporte a vida, deve-se monitorar e controlar o airbag e cinto de segurança. Já no LVT, deve-se monitorar e controlar a luz dos faróis dianteiros, sistema de vidros elétricos do motorista e passageiro (só parte da frente) e trava das duas portas da frente. Todos os sistemas se conectam a um computador de bordo central, que deve fornecer, reter e controlar ações destes sistemas. Cada sistema conta com um sensor e um controlador (totalizando 12 sensores e 12 controladores). Você deve assumir que todos os sensores e controladores estão na mesma distância (1 m) do

computador de bordo e que o tempo de envio de dados é de 1 Mbps, sendo que os dados podem demorar até 10us para se propagar pelo fio (do nodo até o computador). Os deadlines relativos para uma ação de controle (após detecção do sensor) são:

- 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

O computador de bordo deve também alimentar um display para o usuário com alertas de informações (mensagens com caracteres). A atualização do display deverá sempre ser de 500 ms.

A empresa avaliou em uma reunião técnica que os resultados das implementações poderiam ser melhores com o uso de múltiplos núcleos (mais que 2) e que a aplicação de multithread (concorrência e paralelismo) se torna viável. Ela gostaria de monitorar todos os sensores indicados acima e, após amostrar todos os sensores, exibir em um display (ou tela) o estado dos sensores (que indicam o estado dos subsistemas monitorados). A empresa exige que você expanda sua solução para trabalhar com multithread. Além disso, ele gostaria de que você apresentasse um análise, com implementação, de utilização de diferentes esquemas de escalonamento e despacho por afinidade (carga de trabalho) por processador usando API disponíveis na linguagem de escolha.

Outras considerações para a análise temporal:

- sensor demora 1 us para fazer a aquisição da amostra
- o controlador demora 5 us para agir.

Questões

- 1) Você(s) deverá(ão) desenvolver o software requisitado pela empresa, seguindo a especificação sobre sensores, controladores e threads.
- 2) Com o software, vocês deverão fazer uma análise temporal criteriosa do sistema desenvolvido. Poderão usar as bibliotecas `time.h` e `intrin.h` (`x86intrin.h`) ou `clock_gettime` usando `CLOCK_MONOTONIC`.
- 3) **Devem ser identificadas a threads com requisitos temporais hard e soft, e descrever o que levou a essa análise.** Além disso, as operações podem ser consideradas sem requisito temporal (essa escolha deve ser embasada). Lembre-se de atualizar as prioridades de execução das threads do sistema no qual você faz uso.
- 4) Deverá ser avaliado se os tempos de respostas para as situações descritas acima consegue ser atendido na máquina usada para os teste.
- 5) **Deverá ser avaliado o tempo de resposta médio, melhor caso e pior caso de cada sub-rotina avaliada.** Deverá ser exposto as especificações da máquina usada para prototipar o supervisor e o estado do sistema (Sistema Operacional) no momento da execução do supervisor (softwares em execução, utilização de interfaces, sobrecarga, etc).
- 6) O uso de uma técnica de sincronização com mutex é obrigatório. O uso de mutex com protocolo de herança de prioridade dá uma bonificação de 0,5 ponto na prova.
- 7) Você deve analisar se trocar o esquema de escalonamento e indicar o processador a ser usado (afinidade do processador) afetou positivamente ou negativamente o software. Justifique sua análise.
- 8) Você deve aplicar técnica de herança de prioridade e avaliar se traz ganho de desempenho ou não para sistema, além de avaliar de se a resposta das threads de maior prioridade melhora.
- 9) Em caso de problemas na implementação, deverá ser indicado quais as melhorias podem ser feitas a fim de atender os prazos. Essa análise pode contemplar desde a implementação, quanto da estrutura física.