

COMO CONTROLAR DE MANEIRA COMPUTACIONAL O TEMPO DE ACESSO DE VEÍCULOS EM UM CRUZAMENTO DE VIAS

JEFERSON COSTA
Ciência da Computação
Centro Universitário de Brasília
Brasília, Brasil

NATÁLIA SARAIVA
Ciência da Computação
Centro Universitário de Brasília
Brasília, Brasil

VICTOR HUGO BORGES
Ciência da Computação
Centro Universitário de Brasília
Brasília, Brasil

Resumo—Este artigo descreve os principais elementos utilizados na construção de um sistema de tempo real, utilizando arduino para simular o controle computacional do tempo de acesso de veículos em um cruzamento de vias protegidos pelo uso de semáforos. A partir deste sistema é possível visualizar, analisar e determinar o ajuste dos tempos de um semáforo, sendo assim melhorando o trânsito e evitando acidentes.

Palavras-chaves—Semáforo, Tempo Real, Arduino.

I. INTRODUÇÃO

O controle do tráfego de veículos nas grandes cidades, especialmente nos horários de rush, é um dos mais sérios problemas urbanos enfrentados na atualidade. Nas grandes cidades, cerca de 50% dos tempos de viagens e 30% do consumo de gasolina são gastos com os veículos parados nos cruzamentos, esperando que o sinal passe do período vermelho para o verde (DENATRAN, 1979)[2].

Este artigo apresenta a criação do controle do tempo de acesso de veículos em um cruzamento de três vias, utilizando uma plataforma de prototipagem eletrônica de hardware livre e de placa única – Arduino, com sistema tolerante a falhas e tempo de resposta eficaz. Para que desde modo haja uma melhoria na circulação do trânsito e diminuição de acidentes nas vias.

Será demonstrado a estrutura desenvolvida e os resultados obtidos. O artigo está organizado em uma seção que apresenta descrição teórica II Base Teórica, a seção III apresenta o desenvolvimento, a Seção IV os resultados obtidos e V apresentando uma conclusão sobre o assunto.

II. BASE TEÓRICA

Tempo Real

Os Sistemas de Tempo Real são sistemas que operam sobre restrições de tempo, os quais consistem de subsistemas de controle e subsistemas controlados interagindo através de triagem de amostras, processamento e respostas.

Os Sistemas de Tempo Real são classificados em Hard Real Time e Soft Real Time. Este artigo abordará o sistema Hard Real Time que segue as restrições de tempo para evitar consequências catastróficas que estão normalmente relacionadas a vida das pessoas, deste modo controlando o tempo de determinadas atividades em ambientes diferentes.

Semáforo

Semáforo é uma variável especial protegida que tem como função o controle de acesso a recursos compartilhados (por exemplo, um espaço de armazenamento) num ambiente multitarefa.

Segundo [1], iniciada uma operação de semáforo, nenhum outro processo pode ter acesso ao semáforo até que a operação tenha terminado ou sido bloqueada. Isso torna eficiente as resoluções dos problemas de sincronização e evita que aconteça as condições de corridas.

Operações de Up e Down (generalização de wakeup e sleep, respectivamente), sugeridos por Dijkstra (1965), em que o valor 0 sugere que nenhum sinal de acordar foi salvo ou algum valor positivo se um ou mais sinais de acordar estivessem pendentes [1].

Tolerância a Falhas

Para que haja uma confiança com o sistema é necessário a tolerância à falhas que é um conjunto de técnicas utilizadas para detectar, mascarar e tolerar falhas no sistema[3].

O tempo faz-se necessário quanto a obtenção do resultado esperado, com isso há a sua redundância que consiste em executar a mesma computação, em instantes distintos de tempo, afim de verificar a existência de falhas temporais no sistema[3].

Restrições de tempo

Para que se tenha um comportamento temporal desejado, é necessário que todas as restrições temporais sejam respeitadas, pois as tarefas de tempo real estão limitadas a prazos, mais especificamente os seus “deadlines”(tempo máximo que uma tarefa deve ser executada)[4].

Tratando-se do controle de vias através de semáforos, as tarefas executadas são tratadas como críticas, pois ao serem completadas após seu *deadline* pode causar falhas catastróficas, o que, deste modo, associa-se como uma tarefa periódica pois há uma regularidade e portanto uma previsibilidade[4].

Paradigma de programação

Desde o surgimento da primeira linguagem de programação de alto nível, Fortran, na década de 1950, uma grande variedade de linguagens de programação tem sido proposta, como consequência de domínios de aplicação distintos, avanços tecnológicos e interesses comerciais, dentre outros aspectos. Algumas linguagens compartilham características em comum e são ditas pertencerem a um mesmo paradigma: um modelo, padrão ou estilo de programação[5].

Um paradigma de programação determina uma forma particular de abordar os problemas e de formular respectivas soluções, os quais são classificados enquanto seu conceito base, podendo ser: Imperativo, funcional, lógico, orientado a objetos e estruturado. Contudo, uma linguagem de programação pode combinar dois ou mais paradigmas para potencializar as análises e soluções. Com isso, o programador é o responsável por escolher o paradigma mais adequado para resolver e analisar cada problema[6].

III. DESENVOLVIMENTO

Parte do Desenvolvimento

IV. RESULTADOS

Parte do Resultado

V. CONCLUSÃO

Parte da Conclusão

REFERENCES

[1] TANENBAUM, Andrew S. Sistemas operacionais modernos. 3 ed. São Paulo: Pearson, 2009. 625 p.

[2] LEITE, Gilleddson Frytys Menezes; ALVES, Antônio César Baleeiro. IMPLEMENTANDO UM SIMULADOR DE TRÁFEGO URBANO PARA UMA INTERSEÇÃO COM SEMÁFOROS. **SciELO**, Goiânia, v. 1, n. 1, p. 95-107, set./abr. 2016. Disponível em: <http://wsmartins.net/ermacs/trabalho_18.pdf>. Acesso em: 28 abr. 2016.

[3] CIN. **Sistemas de Tempo Real**. Disponível em: <<http://www.cin.ufpe.br/~jvob/introducao.html>>. Acesso em: 29 abr. 2016.

[4] ROMULOSILVADEOLIVEIRA. Sistemas de Tempo Real. Disponível em: <<http://www.romulosilvadeoliveira.eng.br/livro-tr.pdf>>. Acesso em: 29 abr. 2016.

[5] SAMPAIO, Augusto; MARANHÃO, Antônio. Conceitos e Paradigmas de Programação via Projetos de Interpretadores. Disponível em: <<http://www.cin.ufpe.br/~in1007/transparencias/jai/Jai2008Augusto.pdf>>. Acesso em: 30 abr. 2016.

[6] JUNGTHON, Gustavo; GOULART, Cristian Machado. Paradigmas de Programação. Disponível em: <https://fit.faccat.br/~guto/artigos/Artigo_Paradigmas_de_Programacao.pdf>. Acesso em: 30 abr. 2016.