

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

Jéferson Costa, Natália Saraiva, Victor Hugo Borges

Curso de Bacharelado em Ciência da Computação – Centro Universitário de Brasília
(UNICEUB) – Campus de Taguatinga
Taguatinga - DF - Brasil

jefersonc.neves@gmail.com, natalia_saraiva@outlook.com,
victor.hugoborcas18@gmail.com

***Resumo.** Este artigo descreve os principais elementos utilizados na construção de um sistema de tempo real, utilizando uma plataforma de prototipagem eletrônica de hardware livre e de placa única – 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.*

1. Introdução

O controle do tráfego de veículos nas grandes cidades é um dos sérios problemas enfrentados na atualidade, principalmente nos horários de maior fluxo. Segundo o Departamento Nacional de Trânsito - Denatran (1979), 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 libere o acesso à via.

O objetivo deste artigo consiste na apresentação de uma simulação do controle de tráfego no cruzamento de três vias por meio de semáforos, utilizando a arquitetura arduino. O semáforo consiste em controlar o tempo de acesso dos veículos a uma via, a fim de evitar colisões que podem proporcionar grandes acidentes.

Este artigo está organizado em cinco partes. Na seção 2 é apresentado alguns dos principais conceitos abordados no desenvolvimento do simulador. A seção 3 contém todos os processos da implementação do simulador. A seção 4 apresenta os resultados obtidos com a execução do simulador. Por fim a seção 5 apresenta uma conclusão dos resultados obtidos.

2. 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 Tanenbaum (2009), iniciada uma operação de semáforo, nenhum outro processo pode ter acesso ao semáforo até que a operação tenha bloqueado ou sido bloqueada. Isso torna eficientes as resoluções dos problemas de sincronização e evita que aconteça às condições de corrida.

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[Tanenbaum, 2009].

Tolerância às Falhas

Segundo o artigo publicado pelo Centro de Informática da UFPE (2016), para que haja uma confiança com o sistema é necessário à tolerância a falhas que é um conjunto de técnicas utilizadas para detectar, mascarar e tolerar falhas no sistema.

Segundo esse mesmo artigo, o tempo faz-se necessário quanto à obtenção do resultado esperado, com isso há a sua redundância que consiste em executar a mesma computação, em instantes distintos de tempo, a fim de verificar a existência de falhas temporais no sistema.

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).

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 [De Oliveira 2000].

Paradigma de programação

Segundo o artigo de Sampaio e Maranhão (2008), 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.

Segundo o artigo publicado pela Faculdade de Informática de Taquara – FIT (2008), 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.

Referências

- TANENBAUM, Andrew S. Sistemas operacionais modernos. 3 ed. São Paulo: Pearson, 2009. 625 p.
- 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.
- CIN. Sistemas de Tempo Real. Disponível em: <<http://www.cin.ufpe.br/~jvob/introducao.html>>. Acesso em: 29 abr. 2016.
- ROMULOSILVADEOLIVEIRA. Sistemas de Tempo Real. Disponível em: <<http://www.romulosilvadeoliveira.eng.br/livro-tr.pdf>>. Acesso em: 29 abr. 2016.
- 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.
- 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.