

Implantação de conceitos de sistemas de tempo real no controle de esteiras de uma envasadora utilizando arduíno

João Paulo Cunha Ávila

Departamento de Ciências da Computação – Centro Universitário de Brasília
jpcunha4@gmail.com

Abstract:

Resumo:

1. Introdução

O envasamento de garrafas é uma atividade custosa que precisa ser muito bem planejada para que a empresa possa maximizar ao máximo sua produção. Para isto, é necessário que o sistema que gerencie a movimentação das esteiras destas máquinas industriais seja preciso e eficiente.

Para que isto seja alcançado a utilização de um sistema de tempo real se torna necessária ao passo que cada tarefa gerada pelo mesmo deve ser executada com restrições temporais.

Este artigo apresenta a implantação de conceitos de sistemas de tempo real na implementação de um sistema gerenciador de esteiras de uma envasadora de garrafas. Primeiramente, a apresentação destes conceitos se faz necessária para que se tenha um maior entendimento do que foi implementado. Em seguida, é apresentada a forma como o sistema foi planejado bem como seu protótipo na plataforma de arduíno.

2. Referencial Teórico

Segundo Buttazzo (2011), sistemas ou softwares de tempo real são aqueles que devem executar suas funções de maneira que atendam a requisitos temporais de maneira precisa. Benveniste e Berry (1991) explicitam também que sistemas de tempo real são sistemas reativos, ou seja, respondem à eventos do mundo externo, que tem restrições de tempo previamente definidas. Shaw (2003) afirma que quando um sistema desse tipo é parte de um outro maior que está embutido, chamamos tal computador componente de sistema embarcado.

Estas restrições de tempo são representadas por deadlines, que nada mais são do que limites de tempo para as tarefas serem executadas. Tendo isto em vista, é possível classificar estes deadlines como rigorosos e tolerantes. Deadlines rigorosos são aqueles que tem de cumprir todos os requisitos de tempo que lhe foram impostos e, caso isto não ocorra, o sistema falha. Deadlines tolerantes possuem também seus requisitos de tempo mas alguns podem falhar sem muitos problemas para a execução da próxima tarefa (SHAW, 2003).

Apesar de tarefas terem deadlines a serem cumpridos, Lehoczky e Sprunt (1989) expõem três diferentes classificações em relação à periodicidade de execução das mesmas. São elas tarefas periódicas, aperiódicas e esporádicas. Tarefas periódicas são executadas entre intervalos de tempo definidos, sendo que seu período é igual ao seu deadline. Tarefas aperiódicas são aquelas que não tem um período definido, ou seja, são executadas em intervalos irregulares. Tarefas esporádicas são tarefas aperiódicas que possuem deadlines rigorosos.

Shaw (2003) explica que nenhum sistema é perfeitamente confiável logo, existirão falhas que podem ter um custo alto. Beveniste e Berry (1991), explicam que estas falhas tem relação à corretudes lógicas, que tem relação a bugs de software, e temporais, restrições de tempo não atendidas. Portanto, é importante evitar estas falhas o máximo possível e criar uma forma de tratar falhas que venha a ocorrer efetivamente e com o menor custo (SHAW, 2003).

Levando em consideração que o tempo é tratado pelo processador um recurso, faz-se necessário utilizar o escalonamento de processos, que refere-se a capacidade de um processador em executar todas as computações, ou seja, se o mesmo tem clocks suficientes para executar tais processos à tempo. Para alcançar este propósito são utilizados diversos tipos de algoritmos (SHAW, 2003). Para este experimento faz-se necessário a análise de apenas três algoritmos: por taxa monotônica, EDF(Earliest Deadline First) e Deadline Monotônico.

O escalonamento por taxa monotônica trabalha com o princípio que tarefas com o menor período tem maior prioridade (BRANICKY; PHILLIPS; ZHANG, 2002). O algoritmo EDF trabalha com o princípio de que a tarefa com o deadline mais próximo tem maior prioridade de execução (SHAW, 2003). Por fim, o algoritmo de Deadline MonoTônico trabalha com o princípio de que a prioridade é definida pelo menor deadline, ou seja, quanto menor o deadline maior a prioridade (AUDSLEY, 1991).

Um dos maiores problemas do escalonamento é o de concorrência entre threads de um mesmo processo que compartilham a mesma região crítica, causando desta maneira a condição de corrida. Existem diversas maneiras de solucionar este impasse, uma delas é pelo uso de semáforos (TANENBAUM, 2009).

Semáforos atuam como variáveis de controle de regiões críticas. Seu funcionamento é baseado em valores “down” e “up” que vão, respetivamente, colocar um processo no seu estado de bloqueado e liberar o acesso à região crítica por esse processo. Isso já é suficiente para evitar a condição de corrida e resolver possíveis problemas de sincronização (TANENBAUM, 2009).

3. Desenvolvimento

O projeto foi desenvolvido na linguagem C, tendo o Arduíno Uno como responsável pela implementação do sistema embarcado. Foi utilizada a biblioteca NiRTOS para trabalhar com conceitos de tempo real no projeto.

O mesmo consiste no controle de uma envasadora de garrafas. Duas esteiras contendo as garrafas moverão em direção a uma terceira esteira onde ocorrerá o processo de preenchimento com um líquido. Assim que a garrafa estiver cheia a máquina encerra o processo e move a esteira, liberando espaço para que uma próxima garrafa possa acessa-la.

As duas esteiras que contém as garrafas vazias serão tratadas como duas threads diferentes que tentarão acessar a região crítica, que é representada pela terceira esteira onde as garrafas serão preenchidas. O controle de acesso a terceira esteira é feito a partir de um semáforo que decidirá qual das esteiras, número um ou dois, ira mover.

O autômato finito abaixo representa a máquina de estados do sistema sugerido:

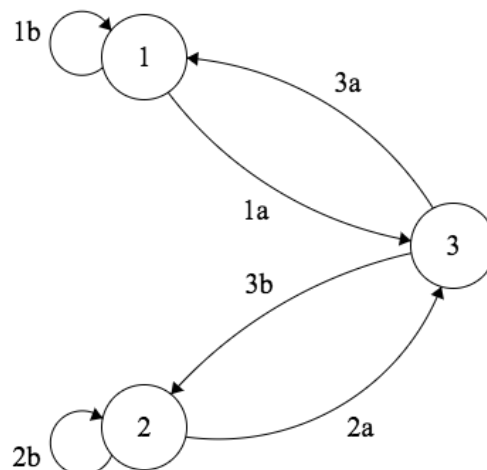


Figura 1. Máquina de Estados de uma Envasadora

Como é possível observar, a máquina possui três estados de funcionamento. O primeiro estado é responsável por mover a esteira 1. Primeiramente ele tenta acessar a região crítica (1a) e caso tenha sucesso irá para o próximo estado, se o acesso for negado (1b) continuará esperando até que tenha acesso à mesma. O segundo estado representa a segunda esteira. Suas funções de transição funcionam de maneira idêntica ao estado de número 1. É importante observar que o semáforo ficará bloqueado assim que um destes tiver acesso à região crítica. O terceiro estado representa a região crítica que é responsável por encher as garrafas e mover a terceira esteira assim que as mesmas estiverem preenchidas. O fim de sua execução é representado por um retorno ao estado que acessou a mesma (3a/3b) informando que tudo ocorreu como esperado e libera o semáforo para o acesso da outra esteira.

A execução das tarefas propostas são feitas de maneira periódica e são escalonáveis utilizando o algoritmo de taxa monotônica, como apresentado matematicamente abaixo:

Tabela 1. Tabela de tarefas periódicas com suas informações

Tarefas periódicas	Período	Deadline	Tempo de Computação
Mover esteira 1	15	15	5
Mover esteira 2	15	15	5

4. Resultados

O desenvolvimento do projeto foi baseado nos conceitos apresentados sobre sistemas de tempo real tendo como base a biblioteca do NilRTOS para a utilização dos mesmos.

A partir da implantação destes conceitos apresentados no protótipo sugerido pode-se observar que o escalonamento por taxa monotônica funciona de maneira eficiente para o conjunto de tarefas periódicas apresentadas.

O período de cada tarefa foi escolhido levando em consideração o fato de que ao acessar a região crítica é levado um tempo de aproximadamente 10 segundos para que a garrafa seja preenchida e a terceira esteira se mova. Os deadlines foram planejados para que não houvesse nenhum hiato entre uma execução e outra levando em consideração que não é interessante para uma empresa que haja demora na execução do processo.

As tarefas relacionadas à região crítica, ou seja, encher a garrafa e mover a terceira esteira, não foram adicionadas à tabela de prioridades devido ao fato de que são tarefas que devem ser executadas apenas quando existe acesso à esta região. Isto quer dizer que elas não poderiam entrar com uma prioridade menor que as outras e, quando o período de outra tarefa mais prioritária chegar, simplesmente pararem sua execução e outra thread tivesse acesso a esta região.

5. Conclusão

Neste artigo foi apresentada uma implementação de um algoritmo para o controle de esteiras de uma envasadora utilizando a abordagem de tempo real com o uso de escalonamento de processos por taxa monotônica.

A partir do que foi implementado é possível observar que é viável a utilização de escalonamento de processos utilizando o algoritmo de taxa monotônica em uma abordagem de tempo real.

O próximo passo é a implementação e aperfeiçoamento do algoritmo utilizado para um ambiente em escala industrial, onde os dados devem ser precisamente medidos para que haja ganhos no processo de envasamento de garrafas.

Referências

G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," *Phil. Trans. Roy. Soc. London*, vol. A247, pp. 529–551, April 1955. SHAW, Alan C. **Sistemas e software de tempo real**. Bookman, 2003.

BUTTAZZO, Giorgio. **Hard real-time computing systems: predictable scheduling algorithms and applications**. Springer Science & Business Media, 2011.

SPRUNT, Brinkley; SHA, Lui; LEHOCZKY, John. Aperiodic task scheduling for hard-real-time systems. **Real-Time Systems**, v. 1, n. 1, p. 27-60, 1989.

TANENBAUM, Andrew. **Modern operating systems**. 2009.

BRANICKY, Michael S.; PHILLIPS, Stephen M.; ZHANG, Wei. Scheduling and feedback co-design for networked control systems. In: **Decision and Control, 2002, Proceedings of the 41st IEEE Conference on**. IEEE, 2002. p. 1211-1217.

AUDSLEY, Neil C. et al. Real-time scheduling: the deadline-monotonic approach. In: **in Proc. IEEE Workshop on Real-Time Operating Systems and Software**. 1991.

BENVENISTE, Albert; BERRY, Gérard. The synchronous approach to reactive and real-time systems. **Proceedings of the IEEE**, v. 79, n. 9, p. 1270-1282, 1991.