

UNIVERSIDADE FEDERAL DE SANTA CATARINA

**BLU3020-08754 (20182) - SISTEMAS COMPUTACIONAIS PARA CONTROLE E
AUTOMAÇÃO**

LUCAS ENDO PRESTES

NÍCOLAS WILSON SOUZA

TRABALHO PRÁTICO 2

BLUMENAU, 21 DE NOVEMBRO DE 2018

Sumário

1. INTRODUÇÃO	3
2. DESENVOLVIMENTO	4
a. INTRODUÇÃO AO TEMA PROPOSTO	4
b. JANTAR DOS FILÓSOFOS	4
3. CONSIDERAÇÕES FINAIS	6

1. INTRODUÇÃO

Este trabalho tem como objetivo apresentar uma solução ao problema que foi discutido no trabalho anterior na matéria de Sistemas Computacionais para Controle e Automação. O problema definido pelo professor tem que ser diferente do que foi implementado em linguagem C. Será feito uma abordagem sobre o problema escolhido pela dupla e assim fazendo sua implementação em freeRTOS.

Após sua implementação, analisaremos e faremos um feedback do mesmo mostrando assim as dificuldades encontradas na hora de implementar no freeRTOS e claro os pontos de futuras melhorias para um próximo trabalho. A solução que será apresentada, não ocorre presença de DeadLocks, ou Starvation.

2. DESENVOLVIMENTO

O trabalho desta presente disciplina deve-se demonstrar um problema simples com intuito de encontrar uma solução para o mesmo, como relatado no item anterior. Porém a solução deve ser eficiente para as mais variadas formas de otimizá-lo. Este problema que será discutido e resolvido, será o problema de concorrência do jantar dos filósofos. Um problema que aborda a programação concorrente e será realizado sua solução em freeRTOS.

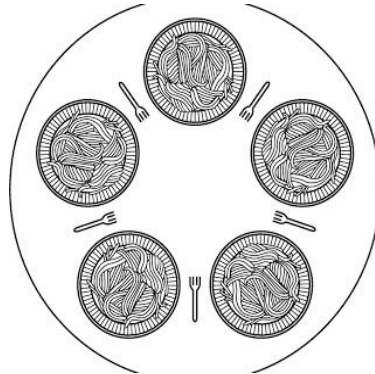
A Programação concorrente é um setor da programação de computadores que faz a execução simultânea de muitas tarefas (multitarefa), juntas ou separadas através de threads, assim tem muitos problemas com a presença de threads e comunicação entre processos.

O freeRTOS é um kernel de um sistema operacional de tempo real para dispositivos na área de eletrônica e computação.

a. JANTAR DOS FILÓSOFOS

Cinco filósofos estão em torno de uma mesa de jantar redonda. Um garfo é colocado entre cada filósofo juntamente com um prato à sua frente com espaguete. O espaguete está muito liso e o filósofo necessita de dois garfos para poder comer. Os dois garfos devem ser aqueles logo a sua esquerda e a sua direita. O filósofo deve alternadamente comer e pensar. O filósofo tenta pegar os garfos à sua direita e à sua esquerda, sendo um de cada vez, não importando a ordem de escolha. Após comer, o filósofo deve liberar o garfo que utilizou. Um filósofo pode segurar o garfo da sua direita ou o da sua esquerda assim que estiverem disponíveis, mas, só pode começar a comer quando ambos estiverem sob sua posse. Se um filósofo pegar um dos garfos e o outro a sua esquerda ou direita não estiver disponível, o mesmo deve soltar o garfo pego para que outro filósofo possa pegar.

Figura 1 - Representação Jantar dos Filósofos



Fonte: papeldiario.blogspot.com (2018)

A falta de garfos para os filósofos está relacionada a parte de recursos compartilhados em uma programação de computador. Quando um programa está interessado em um recurso que está sendo utilizado por outro programa, o mesmo aguarda até que seja terminado. Quando se tem mais de um programa envolvido na parte de no travamento de recursos, um *deadlock* pode ocorrer.

3. CONSIDERAÇÕES FINAIS

Neste presente trabalho, nosso grupo teve muitas dificuldades na implementação do trabalho freeRTOS. O problema que escolhemos para fazer a solução foi o do Jantar dos filósofos.

Tivemos dificuldades em trabalhar com a aplicação de threads em vetores. Não tínhamos o conhecimento desta abordagem, mas com exemplos na programação do arduino pudemos ter o entendimento e poder realizar a mesma em nossa solução.

A parte de projetar o programa (solução) também foi um problema para o nosso grupo. Pois o fato de que os filósofos somente poderiam poder pegar os garfos do seu lado direito e esquerdo, deu uma dificuldade no problema. Tivemos que apresentar nossa ideia e o professor nos orientou a realizar de uma forma que desse certo. Através da utilização de semáforos.

Outra dificuldade foi a parte de comandos no freeRTOS onde não sabíamos alguns comandos para a realização do mesmo, sua biblioteca de comandos. Isso fez com que o trabalho adiasse o prazo estabelecido pelos envolvidos.

No meio da solução do problema houve a parte do Delay que tem a função de pausar o programa. O mesmo não havíamos pensado em colocar na programação. Porém com isso resolveu o problema do filósofo pegar o garfo.

Outra dificuldade foi criar múltiplas tasks com o mesmo nome na programação. Além do fato de se colocar sempre parâmetros para que os filósofos possam comer randomicamente, não havendo uma ordem a ser executada.

Houve ainda um obstáculo que conseguimos solucionar, que é o fato de transformar uma variável sem usar struct na programação. Deixando assim, mais simples.

Uma melhoria que realizamos foi a partir da ideia do professor de colocar o *define* na programação. Onde o programador pode alterar o número de garfos e filósofos de acordo com a sua vontade. Outra ideia foi a questão do *printf*, mostrando assim qual filósofo está comendo e qual está pensando.

Com as dificuldades apresentadas, nosso grupo conseguiu encontrar uma solução para o problema do jantar dos filósofos, com a orientação e supervisão do professor. Com as

dificuldades encontradas e as melhorias realizadas, este trabalho foi de grande valia e obtivemos um grande conhecimento na área de freeRTOS, um sistema de programação diferente da linguagem C a qual estamos acostumados a trabalhar.