Pontifícia Universidade Católica do Rio Grande do Sul

Faculdade de Informática

Programa de Graduação em Engenharia da Computação

Trabalho Prático 1

Jantar para N Filósofos

Maiki Buffet e Marcelo Pereira

Professor: Carlos R. Moratelli

Porto Alegre

Abril, 2015

**Implementação**

Para este trabalho, criamos uma classe especifica para os filósofos, onde cada um será um thread, e cada filosofo possuirá um garfo, que será seu respectivo mutex (semáforo binário). Nesta classe, teremos os contadores das vezes que cada filosofo pensou, comeu, ou tentou pegar ambos os garfos simultaneamente sem sucesso.

O programa inicia perguntando ao usuário o número de threads (filósofos) que ele deseja, e o tempo de execução do programa em questão, caso, algum desses valores seja menor ou igual a zero, o programa não será executado, uma vez que, não existe um jantar com filósofos negativos, ou inexistentes, e um programa com um tempo de execução negativo ou zero. Para isto, utilizamos algumas funções como “setNumeroFilosofos” e “setTempoExecucao”.

Criamos uma variável denominada “fim” que é a responsável por terminar a execução dos threads, além da criação e utilização de um semáforo para a cuidado na hora de tentar “reservar” um garfo. Fora utilizado o comando “malloc” para alocar memória para o uso dos threads e mutexes.

Iteramos de acordo com o número de filósofos escolhido, alocando memória para a quantidade solicitada, além de criar os threads e fazer os testes para verificar a disponibilidade dos garfos na função “Testar”. Nesta função, decrementamos o número de garfos disponíveis, para o filosofo atual, e verificamos a disponibilidade do garfo a sua direita, passando pela função “Verificar”. Nesta função, o filosofo que está sendo testado, pegara o seu garfo, e caso não consiga pegar, ficara pensando aleatoriamente por um tempo entre 0ms e 20ms. Caso consiga, tentara pegar o garfo do filosofo a sua direita, e repetira o passo anterior, além de largar o seu garfo. Caso o filosofo em questão, tenha conseguido pegar ambos garfos (tanto o dele quanto o do filosofo a sua direita), ele comera, e então libera os garfos para poder ficar pensando.

Tanto a função “pensarAleatorio”, quanto as funções “Comer” e “Pensar”, são funções na qual o filosofo descansa pelo tempo proposto no exercício do jantar dos filósofos, 30ms para comer e 200ms para pensar, e faz o incremento do respectivo thread.

Após todas as iterações serem concluídas dentro do tempo de execução, faz-se a impressão das tentativas que cada filosofo obteve (comer, pensar, tentar comer). E para finalizar, após a execução “principal” do programa, os threads são esperados para conclusão, através da função “join”, e são feitos processos de deslocação de memória, e “destruição” dos threads e mutexes, utilizados, assim, encerrando a execução.