UNIVERSIDADE DE SÃO PAULO

INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO (ICMC)

BERNARDO MAIA COELHO

Nº USP: 12542481

Professor: Vanderlei Bonato

Disciplina: SSC0140 - Sistemas Operacionais I

Curso: Bacharelado de Ciência de Computação

RELATÓRIO DA ATIVIDADE DOS FILÓSOFOS

THE DINING PHILOSOPHERS PROBLEM

06 de novembro de 2022

# Introdução

Esse projeto busca implementar uma solução na linguagem C, usando o sistema POSIX, para o problema dos filósofos (“The dining philosophers problem”) discutido no livro “Operating System Concepts”. Neste problema, tem-se que N filósofos estão jantando ao redor de uma mesa redonda. À esquerda e à direita de cada filosofo há um palito “hashi” e no centro da mesa há um pote de arroz. Os filósofos precisam de dois palitos para comer do pote de arroz, porém eles devem dividir tais palitos com os seus colegas adjacentes.

Dessa forma, podemos modelar cada filosofo como sendo uma thread e cada palito como um recurso a ser compartilhado pelas threads. Em uma execução ideal, cada filosofo passa uma determinada quantidade de tempo pensando, depois passa determinada quantidade de tempo comendo. É importante impedir que dois filósofos peguem o mesmo palito ao mesmo tempo e impedir que um ou mais filósofos fiquem sem comer (starvation).

Diagrama

Descrição gerada automaticamente

# Orientações para a execução do código.

A implementação está disponível em: <https://github.com/bTheMage/HungryPhilosophers>

2.1. Em uma máquina com um sistema GNU-Linux da distribuição Debian ou similar, ou com apt instalado, garanta que os pacotes estão atualizados com:

sudo apt update && apt upgrade -y

2.2. Em seguida, garanta o compilador gcc esteja atualizado com:

sudo apt install build-essential -y

2.3. Em seguida, garanta que a biblioteca POSIX Thread (pthread.h) esteja instalada com:

sudo apt install libpthread-stubs0-dev

2.4. Caso não tenha git instalado, a ferramenta apt pode ser usada novamente:

sudo apt install git -y

2.5. Em qualquer pasta que desejar, clone o repositório com:

git clone https://github.com/bTheMage/HungryPhilosophers

2.6. Para compilar o programa, use:

make all

Ou

gcc -pthread main.c -o main

2.7. Para executar o programa, use:

make run

Ou

./main

2.8. Observe o programa executando e imprimindo na tela o estado de cada filosofo pelo tempo que desejar e, quando estiver satisfeito, pressione ctrl + c para encerrar a execução.

# O código e explicações

Devido à ausência de objetos e métodos na linguagem c, e como o algoritmo não muito grande, variáveis e estados relacionadas ao monitor foram implementadas como variáveis globais e métodos foram implementados como funções.

Texto

Descrição gerada automaticamenteAs bibliotecas utilizadas estão listadas logo abaixo. Destaca-se que as ferramentas providas por “inttypes.h” e “stdbool.h” não são necessárias, mas facilitam a legibilidade do código.

A constante N representa a quantidade de filósofos presentes. Ela pode ser modificada antes da compilação, porém recomenda-se mantê-la em um valor não muito elevado, para facilitar a visualização do funcionamento do código e sua execução.

Texto

Descrição gerada automaticamenteA macro “PANIC” serve para interromper o programa em caso de erro grave e “RIGHT\_PHI” e “LEFT\_PHI” calculam os ids dos filósofos adjacentes a algum id de acordo com o valor de N.

Texto

Descrição gerada automaticamenteOs estados em que um filosofo pode se encontrar são definidos em um enum e as seguintes variáveis globais são declaradas.

Na função “main”, há uma fase de inicialização de variáveis, seguida pela criação das threads. Note que o id dos filósofos é convertido e passado como ponteiro.

Texto

Descrição gerada automaticamente

Por fim, entra-se em um loop infinito em que se imprime na tela os estados dos filósofos com determinada frequência. As linhas de código subsequentes não são executadas, porém foram adicionadas para exemplificar uma sequência de finalização de um programa como este.

A função “runner” é executada por cada thread e tudo o que ela faz é chamar a função “pickup” e “putdown” para pegar e soltar os palitos, além de dormir por um determinado tempo, isso em loop infinito.

A função “pickup” muda o estado do filosofo para faminto (“HUGRY”) e checa se os palitos adjacentes estão disponíveis (o que é equivalente a checar se os filósofos adjacentes estão comendo, no estado “EATING”). Se estiverem indisponíveis, o filosofo se coloca numa lista de espera e fica aguardando.

Texto

Descrição gerada automaticamente

A função putdown muda o estado do filosofo para pensado (“THINKING”) e sinaliza que os palitos estão disponíveis (acordado as threads adjecentes).

Texto

Descrição gerada automaticamente

A função test, usada nas funções anteriores, serve para checar a disponibilidade dos palitos e muda o estado para “EATING”. Ela também sinaliza à thread a disponibilidade dos palitos.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

O código não foi mostrado em sua plenitude nesse tópico, então recomenda-se que se cheque o repositório git. Ademais, pode haver pequenas mudanças como em comentários ou no número das linhas no repositório.

# Resultados

A execução do programa mostra que os filósofos conseguem revezar o acesso ao palitos com sucesso e sem starvation. Indícios de starvation foram observados quando a frequência de impressão dos estados estava em sincronia com a frequência de espera (sleep) das thread. Porém, ao colocar tal frequência fora de fase, tais indícios cessaram. Portanto, em execuções futuras do programa, recomenda-se que a impressão dos resultados seja testada em múltiplas fases e frequências.

A imagem abaixo é advinda de uma impressão com um período de 0.090909 segundos.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente