

## Laboratório II – Sistemas Operacionais

Introdução: o objetivo deste projeto é praticar a concorrência em sistemas operacionais. A implementação deverá ser feita em uma distribuição Linux e usando linguagem C.

### Descrição:

O aluno deverá escrever um programa que resolva o problema conhecido como jantar dos filósofos. Este problema pode ser entendido da seguinte maneira:

Um conjunto N de filósofos está sentado em torno de uma mesa circular. Cada filósofo tem um prato de espaguete. O espaguete está tão escorregadio que o filósofo precisa de dois garfos para comê-lo. Entre cada par de pratos está um garfo. A vida de um filósofo consiste em alternar períodos de comer e pensar. Quando o filósofo fica com fome, ele tenta pegar os garfos a sua esquerda e a sua direita, um de cada vez. Se ele conseguir pegar os dois garfos, ele comerá durante um determinado tempo e, então, colocará os garfos na mesa novamente e continuará a pensar.

Veja que podemos considerar os garfos como recursos compartilhados, e se todos os filósofos pegarem o seu garfo esquerdo (ou direito) ao mesmo tempo, teremos a ocorrência de um impasse (deadlock).

1) Implemente um programa que receba como parâmetro fornecido pelo usuário a quantidade N de filósofos. Cada filósofo será identificado por um número inteiro de 0 até N-1. Este programa deve simular o jantar dos filósofos conforme descrito acima. Um garfo não pode estar de posse por mais de um filósofo.

Solução 1) Resolva o problema dos filósofos utilizando um saleiro (semáforo) para controlar quem está comendo. Apenas um filósofo poderá comer por vez e quando ele devolver o saleiro o próximo filósofo poderá comer. (Esta solução equivale a 70% da nota se for implementada corretamente)

Solução 2) A solução em 1 resolve o problema dos filósofos, porém é ineficiente, pois outros filósofos que não fossem vizinhos poderiam comer ao mesmo tempo de outro filósofo. Altere a solução acima de modo que o filósofo verifique se o filósofo da esquerda e o da direita estão comendo. Caso negativo, o filósofo poderá comer. Caso positivo ele ficará bloqueado até que os dois vizinhos deixem de comer. (Esta solução equivalerá ao restante da nota caso funcione corretamente).

### #DICAS

1) A tarefa do filósofo estar comendo ou pensando pode ser representada pela impressão de uma mensagem no terminal do usuário dizendo que o filósofo i (i representando o identificador do filósofo) está pensando ou comendo. Ao mudar do estado de comer para pensar, pode ser impressa uma mensagem também na tela do usuário, para facilitar a compreensão pelo usuário da dinâmica ocorrida no jantar

2) No início da execução do programa, todos os filósofos estarão no estado “pensando” e tentarão comer de maneira aleatória.

3) Para implementar a solução 2 desta atividade, considere a fórmula  $(i+N-1)\%N$  para representar o filósofo da esquerda e a fórmula  $(i+1)\%N$  para o filósofo da direita (% representa a operação de módulo).

Esta atividade deve ser feita em dupla e enviada via sistema acadêmico até o final do dia 23/05. Caso sejam identificadas cópias dos trabalhos de outros colegas a nota será zerada.

Alguns comandos úteis em C para a implementação da atividade:

- `int pthread_mutex_init(pthread_mutex_t *restrict mutex, const pthread_mutexattr_t *restrict attr);` → inicializa um semáforo do tipo mutex
- `int pthread_mutex_lock(pthread_mutex_t *mutex);` → adquire o lock em um mutex
- `int pthread_mutex_unlock(pthread_mutex_t *mutex);` → libera um semáforo mutex
- `int pthread_mutex_destroy(pthread_mutex_t *mutex);` → destroi um semáforo mutex