

Instruções:

- Utilize o Linux para realizar a atividade. Sempre que estiver em dúvida sobre o funcionamento de uma função ou chamada, utilize o manual do Linux, disponível através do comando *man*.
- Submeta as respostas preenchendo o gabarito disponível no Moodle.

Exercício de laboratório 03 – Problemas clássicos – O Jantar dos Filósofos

Introdução

Este lab guide apresenta um exemplo de código para solução do problema “O problema do Jantar dos Filósofos”, também conhecido como ‘Filósofos Glutões’. Este problema trata-se de uma analogia para um grupo de processos (os filósofos) que necessitam adquirir recursos (os garfos) compartilhados com outros processos para realizar uma tarefa (o jantar). Uma explicação mais detalhada pode ser vista na seção 2.5 do livro “Sistemas Operacionais Modernos, A. S. Tanenbaum, 4ed”. Antes de responder aos questionário abaixo, leia a seção mencionada para se familiarizar com o problema.

O código apresentado no lab guide foi baseado no que é mostrado na Figura 2-47 do livro supracitado. Contudo, a implementação foi feita representando os filósofos como threads ao invés de processos, utilizando a biblioteca *pthread*.

Outras sugestões de leitura:

http://wiki.icmc.usp.br/images/7/76/Aula06_2.pdf

<https://blog.pantuza.com/artigos/o-jantar-dos-filosofos-problema-de-sincronizacao-em-sistemas-operacionais>

→ **Compile o arquivo *filosofos.c* e execute o programa algumas vezes para observar seu comportamento.**

Questões:

1) Considerando o problema de forma genérica, sendo N o número de filósofos, qual o máximo de filósofos que podem estar jantando ao mesmo tempo?

- a) N-1
- b) N-3
- c) N/2, arredondado para baixo
- d) N/2, arredondado para cima
- e) 2

2) Qual o principal problema da solução mostrada da figura 2-46 do livro?

- a) Starvation
- b) Deadlock
- c) Exclusão mútua
- d) Falta de temporização
- e) N.R.A.

3) Qual a razão dos semáforos dos garfos serem iniciados com o valor 0?

- a) Trata-se de uma decisão de implementação, pois um semáforo pode ser iniciado com qualquer valor inteiro.
- b) Porque estes semáforos devem ter o mesmo valor de início do mutex.
- c) Garantir que um garfo só estará disponível quando os vizinhos não estiverem comendo.
- d) Garantir a ordem em que os filósofos jantam.
- e) O valor deve ser 0 porque os semáforos são compartilhados entre as threads, conforme pode ser visto no manual, utilizando *man sem_init*.

4) Sobre a solução apresentada, qual das afirmativas abaixo está INCORRETA?

- a) Trata-se de uma solução onde há ausência de *deadlock*.
- b) Trata-se de uma solução onde há ausência de *starvation*.
- c) Trata-se de uma solução eficiente em termos de paralelismo.
- d) Trata-se de uma solução que funciona também para outros valores de N.
- e) Trata-se de uma solução que evita o bloqueio das threads.

5) Considerando as afirmações abaixo sobre o uso da variável *mutex* no código apresentado, é CORRETO afirmar:

- I - Poderia ser iniciada com o valor 1, desde que removida a instrução *up(&mutex)* da função *main*.
- II - É usada apenas para garantir exclusão mútua no acesso aos estados dos filósofos.
- III – Tecnicamente, não opera como um mutex, mas como um semáforo binário.

- a) Somente I está errada.
- b) Somente II está errada.
- c) Somente III está errada.
- d) Nenhuma está errada.
- e) Todas estão erradas.