



UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
DEPARTAMENTO DE INFORMÁTICA APLICADA

INF01151 – SISTEMAS OPERACIONAIS II N
SEMESTRE 2014/1

TRABALHO 2: SEMÁFOROS E VARIÁVEIS DE CONDIÇÃO

1. ESPECIFICAÇÃO DO TRABALHO:

O trabalho consiste em implementar em C (não em C++), usando pthreads, o problema do jantar dos filósofos. O programa deve ser implementado em duas versões diferentes (detalhes a seguir). Os programas deverão executar obrigatoriamente em ambientes Unix (Linux) mesmo que tenham sido desenvolvidos sobre outras plataformas.

O problema do jantar dos filósofos é descrito em vários livros de sistemas operacionais e de programação concorrente. A seguir é feita uma breve descrição do problema. Em caso de dúvidas, é possível consultar os livros do Tanenbaum (Modern Operating Systems), e do Silberchatz (Operating System Concepts). Ainda, nesses livros e na Internet, se encontram soluções em pseudo-C, pseudo-Java e, as vezes, códigos completos em Java.

2. JANTAR DOS FILÓSOFOS

Existem N filósofos que passam suas vidas comendo spaghetti e pensando. Cada filósofo possui seu próprio lugar em uma mesa circular e na sua frente há um prato de spaghetti. Existem N garfos, um entre cada dois filósofos, e os únicos garfos que cada filósofo pode utilizar são aqueles que estão imediatamente a sua esquerda e a sua direita. Para comer, um filósofo necessita de dois garfos.

O trabalho consiste em simular o comportamento dos filósofos evitando situações de deadlock (bloqueio permanente) e de postergação indefinida. Da definição do problema tem-se que nunca dois filósofos vizinhos poderão estar comendo ao mesmo tempo e que, no máximo, N/2 filósofos poderão estar comendo de cada vez.

O programa receberá como parâmetro de execução o número N de filósofos. Cada filósofo deverá ser simulado por uma thread. O programa executará um laço infinito. Durante a execução do programa cada filósofo passará por três estados distintos: comendo, pensando e tentando adquirir os garfos. Os estados pensando e comendo deverão ser simulados com a execução da função *sleep(n)*, onde *n* é um número de segundos entre 1 e 10 sorteado aleatoriamente com o auxílio da função *rand()*. O estado "tentando adquirir os garfos" é implementado pelo bloqueio da thread a espera de adquirir o recurso "dois garfos".

Para cada mudança de estado de cada filósofo, o programa deverá imprimir na tela uma mensagem na forma de uma string E (para *eating*), T (para *thinking*) e H (para *hungry*, indicando que ele está tentando adquirir os garfos). Portanto, a saída terá um formato do tipo:

```
....  
TTTTT  
HTETE  
EHETT  
....
```

onde cada letra representa um filósofo. O filósofo que está a direita é vizinho imediato do que está mais a esquerda, ou seja, a representação acima é de uma lista circular. Da esquerda para direita, temos os filósofos 0, 1, 2, 3 e 4.

Dica: durante a execução do programa NUNCA poderá haver dois 'E's consecutivos (atenção à lista circular na representação) e mais do que N/2 'E's. Isso auxilia a ver se o programa está correto.

3. VERSÕES: SEMÁFOROS E VARIÁVEIS DE CONDIÇÃO

Você deverá implementar duas versões para a solução do problema dos filósofos: uma versão ilustrando a utilização da biblioteca de semáforos *unnamed* do UNIX, e outra versão que faça uso de variáveis de condição. DICA 1: lembre-se que apesar de pthreads não suportarem diretamente a implementação de monitores, os mesmos podem ser simulados através do uso de variáveis de condição e combinados com mutexes (para garantir que procedimentos são executados de forma mutuamente exclusiva, o corpo de todos os procedimentos devem ser envoltos por um mutex lock e um mutex unlock). DICA 2: o livro do Silberchatz, "Operating System Concepts", 8th edition, apresenta na seção 6.7.2 uma solução para o problema dos filósofos utilizando monitores e variáveis de condição.

4. DESCRIÇÃO DO RELATÓRIO A SER ENTREGUE

Deverá ser produzido e entregue um relatório fornecendo os seguintes dados:

- Descrição do ambiente de teste: versão do sistema operacional e distribuição, configuração da máquina (processador(es) e memória) e compiladores utilizados (versões).
- Explique claramente como as diferentes soluções garantem que as threads não sofrerão **deadlock** ou **postergação indefinida**. JUSTIFIQUE, e **descreva a solução implementada**.

Entregar dois programas C diferentes (jantar dos filósofos com semáforos e jantar dos filósofos com variáveis de condição).

5. DATAS E MÉTODO DE AVALIAÇÃO

O trabalho pode ser feito em grupos de **2 OU 3 INTEGRANTES**. Não esquecer de identificar claramente os componentes do grupo no relatório.

Faz parte do "pacote" de entrega os fontes e o relatório em um arquivo ZIP. O trabalho deverá ser entregue até 23:55 horas do dia **24/04/2014 (Turma A)**, e até 23:55 horas do dia **28/04/2014 (Turma B)**. A entrega deverá ser via moodle (link para submissão na Aula 07).

Após a data de entrega o trabalho deverá ser entregue via e-mail para alberto@inf.ufrgs.br (subject do e-mail deve ser "INF01151: Trabalho 2"). Neste caso, será descontado 02 (dois) pontos por semana de atraso. O atraso máximo permitido é de duas semanas, isto é, nenhum trabalho será aceito após 08/05/2014 (Turma A), e 12/05/2014 (Turma B).