



Trabalho de Sistemas Operacionais - CC
Turma 128
Trabalho Prático 1 (TP1)



Prof. Carlos R. Moratelli
Entrega 26/04/2017

Roteiro

Objetivos

1. Utilizar programação multithreading.
2. Praticar técnicas de sincronização de processos.
3. Estudar e compreender o funcionamento do mecanismo de comunicação entre processos no UNIX.

1. (6 pontos) Jantar para N filósofos

Este exercício prático consiste na implementação de uma aplicação multithreading capaz de modelar o problema do jantar dos filósofos para N filósofos.

Cada filósofo deve ser modelado como uma thread em linguagem C utilizando a biblioteca POSIX Thread (PThread) conforme visto em aula. Os garfos consistem em semáforos. Cada par de filósofos adjacentes possuem um garfo compartilhado.

Os filósofos devem alternar entre os estados PENSANDO e COMENDO. Um filósofo gasta 5 segundos pensando e 2ms comendo. Para poder comer, o filósofo deve ter sido capaz de segurar os garfos da esquerda e da direita simultaneamente. Se ele não conseguir pegar os garfos ele deve voltar a pensar durante um tempo aleatório entre 0 e 3 segundos antes de tentar novamente.

O usuário da aplicação deve ser capaz de **escolher a quantidade de filósofos e o tempo de execução da aplicação**. Ao final da execução, a aplicação deve exibir um relatório indicando quantas vezes cada filósofo comeu, pensou e tentou mas não conseguiu pegar os garfos.

2. (4 pontos) Comunicação entre processos com memória compartilhada.

Crie dois processos que se comuniquem através de memória compartilhada no Linux. A

implementação seguirá o modelo cliente/servidor. O processo servidor irá criar um segmento de memória compartilhada usando a chamada de sistema `shmget()` e ficará esperando o envio de mensagens de uma aplicação cliente. O servidor deverá utilizar um semáforo (`sem_t`) para realizar a espera da mensagem, evitando espera ocupada. O cliente deverá sinalizar o servidor do envio da mensagem que será exibida na tela. O servidor deve ficar em loop infinito aguardando e exibindo as mensagens, enquanto que o cliente envia uma mensagem e é encerrado. Maiores explicações sobre o uso da chamada `shmget()` pode ser visto no link abaixo. Ainda, o link fornece um exemplo cliente/servidor que pode ser usado como ponto de partida.

IPC Shared Memory: <http://users.cs.cf.ac.uk/Dave.Marshall/C/node27.html>

Entrega

O trabalho pode ser realizado preferencialmente em grupos com no máximo 2 alunos. É permitido realizar o trabalho individualmente ou em grupos de até 3 alunos. Junto com a implementação, em linguagem C, deve ser entregue um relatório de no máximo 2 páginas explicando detalhes da implementação e dificuldades encontradas. A entrega do trabalho deve ser realizada pelo Moodle (código fonte e relatório) e o mesmo deve ser apresentado em sala de aula na data indicada no cronograma da disciplina.