

Trabalho de Sistemas Operacionais (SI)

Trabalho Prático 1

Prof. Carlos R. Moratelli

Entrega 22/09/2017

Roteiro

1. Objetivos

1. Utilizar programação multithreading e comunicação entre processos.
2. Praticar técnicas de sincronização de processos.

1. (5 PONTOS) Jantar dos Canibais

Suponha que um grupo de N canibais serve-se a partir de uma grande travessa que comporta M porções. Quando alguém quer comer, ele(a) serve-se da travessa, a menos que esteja vazia. Neste caso, o canibal acorda o cozinheiro e espera até que ele coloque mais M porções na travessa. Desenvolva um código que:

- Modele os canibais e o cozinheiro como threads;
- Implemente e sincronize as ações dos canibais e do cozinheiro:
 - `se_serve()` - chamada pelo canibal;
 - `enche_travessa()` - chamada pelo cozinheiro.

A solução deve evitar deadlock e deve acordar o cozinheiro apenas quando a travessa estiver vazia. Os canibais apenas devem voltar a comer quando a travessa estiver completamente cheia. Da mesma forma, o cozinheiro apenas pode ser acordado quando a travessa estiver completamente vazia. Suponha um longo jantar, onde cada canibal serve-se continuamente e come, sem se preocupar com os outros afazeres da vida de um canibal. O programa deve aceitar N e M como parâmetros de entrada. **Dica:** i) Modele a travessa como uma classe que implementa os métodos `se_serve()` e `enche_travessa()`, toda a sincronização deve ficar limitada ao acesso a classe travessa. ii) Use a primitiva `Semaphore.tryAcquire()` para verificar se um semáforo está fechado sem ficar bloqueado.

O professor irá disponibilizar um skeleton do código no Moodle.

2. (5 PONTOS) Servidor de arquivos multithreading.

Para este exercício deve ser implementado um servidor de arquivos multithreading utilizando o protocolo da camada de transporte TCP. O servidor deverá aceitar conexões de um ou mais clientes simultaneamente e transferir os arquivos solicitados. A transferência de arquivos deve seguir o protocolo especificado abaixo:

- 1) O cliente inicia a conexão com o servidor.
- 2) Cliente envia o nome do arquivo desejado em formato string (terminando com delimitador de linha \n);
- 3) Servidor determina o tamanho do arquivo em bytes e envia esta informação em formato string para o cliente (terminando com delimitador de linha \n);
- 4) O Servidor inicia o envio dos bytes do arquivo em porções de 1024 bytes até a conclusão do envio do arquivo.
 - a) Enquanto isso o cliente recebe os bytes enviados e salva em um arquivo local.
- 5) Ao término da transferência do arquivo, cliente e servidor encerram a conexão. O servidor deve aguardar novas conexões.

No dia da apresentação, os grupos deverão mostrar o servidor tratando ao menos 2 clientes simultaneamente.

3. Entrega

O trabalho pode ser realizado em grupos com no máximo 3 alunos. É permitido realizar o trabalho individualmente ou em grupos de 2 alunos. Junto com a implementação, em linguagem Java, 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.