

Desenvolvimento de um Sistema Distribuído

Prof. Carlos Roberto Moratelli

1 Informações Gerais

- Primeiro trabalho de Sistemas Computacionais (TP1).
- Semestre 2023/1.
- Entrega: 28/04.
- Equipes de até 3 integrantes.

2 Descrição do Projeto

Um sistema distribuído é um conjunto de computadores interconectados que trabalham juntos para realizar uma tarefa, compartilhando recursos e coordenando suas ações para fornecer um serviço transparente e consistente aos usuários.

Este projeto visa desenvolver um sistema distribuído utilizando programação C, pthreads e um ambiente de execução Linux. O sistema consiste em um servidor que gerencia a comunicação entre clientes e workers, onde os workers são responsáveis por processar as solicitações enviadas pelos clientes.

2.1 Principais componentes do sistema

1. **Servidor:** Gerencia a comunicação entre clientes e workers, distribuindo as solicitações de processamento de acordo com a disponibilidade dos workers. O servidor também é responsável por encaminhar as respostas dos workers para os respectivos clientes.
2. **Workers:** São unidades de processamento que recebem solicitações de processamento enviadas pelos clientes, por meio do servidor. O professor irá disponibilizar a implementação do worker.
3. **Clientes:** São processos que enviam solicitações de processamento para os workers por meio do servidor. Os clientes devem ser capazes de se conectar ao servidor e enviar as solicitações.

3 Requisitos

1. O projeto deve ser desenvolvido em linguagem C, utilizando a biblioteca de pthreads para a implementação de threads no servidor. Workers e clientes são single threads.

2. O sistema operacional utilizado deve ser baseado em Linux.
3. O servidor deve conseguir gerenciar múltiplas conexões TCP simultâneas com clientes e workers.
4. Os workers processam as solicitações e respondem a uma única requisição por vez.
5. O sistema deve ser escalável, permitindo a adição de novos workers conforme necessário.
6. O servidor deve manter uma lista de workers, esta lista deve informar se um worker está ocioso ou em processamento.
7. O servidor deverá disparar uma nova thread para cada cliente conectado, para realizar o encaminhamento das mensagens.
8. Se um cliente conectar e não houver worker disponível, a mensagem “Sistema ocupado. Tente mais tarde.” deve ser retornada e a conexão encerrada.

4 Implementação

Cliente, servidor e workers devem trocar mensagens em formato string. O exemplo de worker implementado e disponibilizado pelo professor implementa 4 operações simples, que servem como exemplo: add, subtract, multiply and divide. Por exemplo, se um cliente desejar realizar uma operação de soma, deverá enviar a seguinte string:

```
"add 2 2"
```

Ao receber tal mensagem, o servidor irá procurar um worker ocioso e reencaminhar a mensagem. O worker irá processar, realizando *parser* da mensagem e respondendo. Para este exemplo, a mensagem será:

```
"4.0"
```

O servidor irá receber a mensagem e reencaminhá-la para o cliente. Observe que cada mensagem é uma string, portanto, o carácter `'\0'` deve ser obrigatoriamente enviado ao final da mensagem.

Tanto cliente quanto worker, logo ao estabelecer uma conexão com o servidor, devem enviar a sua identificação, ou seja, se for um cliente, deve enviar a string "client" e um worker deve enviar a string "worker".

Dica: Lembre-se que o TCP é um *socket stream*, ou seja, mensagens podem ser concatenadas e entregues em grupos, como um fluxo de bytes. Assim, recebe-as procurando por um fim de string (`'\0'`) para saber onde uma mensagem termina.

5 Fontes

Um esqueleto do código e a implementação do worker pode ser encontrado em: <https://github.com/crmoratelli/TP1-2023-1.git>

Para download faça:

```
git clone https://github.com/crmoratelli/TP1-2023-1.git
```

Para compilar, entre no diretório e use o utilitário make:

```
make
```

6 Entregáveis

1. Código-fonte do servidor, workers e clientes.
2. Shell script para disparar um certo número de clientes por segundo. a taxa de clientes deve ser informada como parâmetro.
3. A equipe deve mostrar em funcionamento e explicar para o professor a implementação.