

FOCO NO MERCADO DE TRABALHO

PROCESSOS E COMUNICAÇÃO EM SISTEMAS DISTRIBUÍDOS

Marluce Rodrigues Pereira

Ver anotações

MODELO DE COMUNICAÇÃO ENTRE PROCESSOS CLIENTE-SERVIDOR

A comunicação entre dois processos (cliente e servidor) via rede ocorre através de uma interface de software denominada *socket*, que é uma interface entre a camada de aplicação e a de transporte.



Fonte: Shutterstock.

Deseja ouvir este material?

Áudio disponível no material digital.

SEM MEDO DE ERRAR

a. A implementação do chat com mensagens de texto pode ser realizada utilizando *sockets* e um modelo de comunicação entre processos cliente-servidor. No contexto desta atividade, o cliente e o servidor poderiam ter as seguintes características:

- **Servidor:** é o processo que centraliza todas as mensagens (requisições) de clientes recebidas via *socket*. Tem como responsabilidade o envio de mensagem com resposta a cada requisição recebida. Quando uma requisição chega ao servidor, é criado um *thread* para atender àquela requisição.
- **Cliente:** o usuário cria uma instância do processo cliente que faz uma requisição ao servidor via *socket*. Na requisição do cliente, são informados o endereço do servidor e a respectiva porta.

O servidor do chat utiliza *sockets* e aguarda por conexões de clientes. A cada cliente que requisita uma conexão, é criado um novo *thread* para controle dele.

O cliente é formado por um processo principal, o qual habilita o console para envio de mensagens, porém há um *thread* que aguarda por mensagens que venham do servidor, as quais vieram de outros clientes.

- b. Para qual tipo de dispositivo deve ser implementada a parte da aplicação que tem o processo cliente? E para o processo servidor?

A parte cliente da aplicação deve ser implementada para smartphone com acesso via navegador ou aplicativo instalado localmente, já que permite acesso mais fácil por motoristas. Para haver conexão com o servidor, o smartphone deverá ter acesso à internet por rede de dados celulares ou rede sem fio.

A parte servidor da aplicação deverá ser implementada para uma máquina que tenha maior capacidade de processamento e que deverá executar em alta disponibilidade (24 horas por dia, 7 dias por semana).

- c. Como poderá ser implementada a comunicação realizada entre os colaboradores? Pense na linguagem de programação, na biblioteca ou API e como coordenar a troca das mensagens.

A comunicação poderá ser implementada utilizando API de *sockets* da linguagem adotada pela empresa para desenvolvimento, como C++, Python ou Java. A troca de mensagens é realizada passando pelo gerenciamento do servidor que coordena a ordem das mensagens.

- d. Como o servidor poderá ser implementado para atender a um número maior de conexões?

O servidor poderá ser implementado usando vários threads de execução para atender às requisições dos clientes.

AVANÇANDO NA PRÁTICA

IMPLEMENTANDO O JOGO LIG4, DA ESTRELA

O jogo possui quatro passos:

1. Separar as fichas pretas e brancas, sendo 21 de cada, e sortear o jogador que iniciará a partida. O jogador que começa a partida fica com as fichas brancas.
2. Os jogadores intercalam a colocação de suas fichas no quadro (tabuleiro com seis linhas e sete colunas).
3. O objetivo de cada jogador é enfileirar quatro fichas de sua cor, seja na horizontal, na vertical ou na diagonal.
4. O vencedor é o primeiro jogador a enfileirar quatro fichas.

Descreva como este jogo pode ser implementado para dois jogadores usando um modelo cliente-servidor, no qual os jogadores são os clientes e o servidor recebe as requisições dos clientes (jogadas), atualiza o tabuleiro e sinaliza quando há vencedor.

RESOLUÇÃO

A solução para este problema envolve um modelo cliente-servidor, no qual os jogadores são os clientes. Uma jogada é definida no cliente de acordo com o estado do tabuleiro e as jogadas possíveis, obedecendo às regras do jogo. Ao fazer uma jogada, uma requisição é enviada para o servidor, que atualiza o tabuleiro e envia um retorno para o cliente.

O servidor é responsável por sortear qual jogador começa o jogo e atender às requisições. Quando um cliente envia requisição (jogada) para o servidor, este verifica se a jogada é válida e se há um ganhador. Se houver um ganhador, o servidor deverá avisar qual é o vencedor e encerrar o jogo. Se não houver um ganhador naquela jogada, o servidor deverá atualizar o tabuleiro e enviar para o cliente, como resposta à requisição. A comunicação entre processos cliente e servidor poderá ser realizada utilizando uma API de sockets da linguagem de programação escolhida.