**1) Analise os exemplos dados em aula de Sockets TCP, altere os mesmos, e responda as seguintes perguntas:**

**a. O que acontece quando um cliente tenta se conectar a um servidor inativo/inexistente?**

Quando um cliente tenta se conectar a um servidor inativo ou inexistente, a tentativa de conexão geralmente resulta em um erro, indicando que a conexão não pode ser estabelecida. Esse erro pode ser devido a várias razões, incluindo o servidor não estar em execução, estar em uma porta errada, o endereço IP incorreto, ou problemas de rede, entre outros. O cliente receberá uma mensagem de erro que indica que a conexão falhou, e o programa cliente deve tratar esse erro de acordo com sua lógica de tratamento de exceção.

**b. As operações send e receive são bloqueantes ou não? Se sim, por quê?**

As operações send e receive em sockets são bloqueantes porque, por padrão, elas fazem com que o programa espere até que a operação seja concluída. Isso significa que, se não houver dados para enviar ou receber no momento, o programa fica parado até que haja dados disponíveis ou a operação seja bem-sucedida. Essa abordagem é útil para garantir que a comunicação ocorra de maneira confiável, mas pode causar pausas no programa se não houver dados imediatamente disponíveis para envio ou recebimento.

**c. Os timeouts são aplicados por default em operações connect, send e receive? É possível alterá-los (ou defini-los, caso não sejam utilizados)?**

Os timeouts não são aplicados por padrão em operações conecta send e receive

em sockets. Por padrão, essas operações são bloqueantes, o que significa que o programa aguardará indefinidamente até que a operação seja concluída ou ocorra um erro.

No entanto, é possível configurar timeouts para essas operações. Você pode definir um tempo limite (timeout) para a operação de connect usando a função settimeout, que especifica por quanto tempo o socket deve tentar se conectar antes de gerar um erro de tempo limite. Da mesma forma, você pode usar a função settimeout para definir um tempo limite para as operações de send e receive, especificando por quanto tempo o socket deve esperar por dados antes de gerar um erro de tempo limite.

**d. O servidor, ao chamar o método accept para aguardar conexões, fica esperando para sempre ou existe algum timeout? É possível definir/modificar isso?**

O servidor, ao chamar o método accept para aguardar conexões, fica esperando por uma conexão indefinidamente por padrão. Isso significa que o servidor continuará aguardando novas conexões até que uma conexão seja estabelecida ou até que o servidor seja interrompido.

É possível definir um tempo limite (timeout) para a operação accept em alguns sistemas operacionais. Isso permite que o servidor aguarde uma conexão por um período específico e, se nenhuma conexão for estabelecida dentro desse tempo, a operação accept gerará um erro de tempo limite.

A capacidade de definir um timeout para accept pode variar dependendo do sistema operacional e da linguagem de programação utilizada. No entanto, essa funcionalidade não está disponível em todos os sistemas ou implementações de sockets. Em sistemas onde é suportada, a configuração do timeout pode ser feita usando funções ou métodos específicos, como settimeout. Certifique-se de verificar a documentação do sistema operacional ou da linguagem que está utilizando para obter informações específicas sobre como definir timeouts para accept.