

Definição do Trabalho

A definição do trabalho consiste em implementar uma aplicação que simula o funcionamento de uma rede local em anel. A aplicação deverá implementar a transmissão de mensagens e arquivos entre as máquinas que compõem o anel, utilizando o protocolo UDP como transporte. Deve ser implementada uma fila para armazenar as mensagens e arquivos que serão enviados por cada máquina da rede, e somente um item (mensagem ou arquivo) pode ser transmitido por vez.

Em uma rede em anel, há o uso de *tokens*, que são pacotes especiais que circulam na rede e permitem a transmissão das mensagens e arquivos na rede. Desta forma, a aplicação deverá implementar o token que ficará circulando na rede.

O programa deverá possuir dois tipos de pacotes: o token e os dados. Ao iniciar o programa, o usuário deverá informar o endereço IP da máquina que está a sua direita, um apelido e o tempo do token e dos dados.

As máquinas da rede serão identificadas por apelidos e deve-se especificar o tempo que a estação permanecerá com os pacotes (para fins de depuração), em segundos. Tais informações devem ser inseridas na aplicação através de um arquivo de configuração. Além disso, apenas uma determinada máquina será responsável por gerar o token a primeira vez (esta máquina deve ter o valor *true* no arquivo de configuração, enquanto as outras terão o valor *false*).

O arquivo de configuração deverá seguir o seguinte formato:

```
<ip_destino_token>:porta  
<apelido_da_máquina_atual>  
<tempo_token>  
<token>
```

Exemplo:

```
192.168.1.12:6000  
Bob  
1  
true
```

A máquina que gera o token deve enviá-lo para a máquina que está a sua direita no anel (IP configurado no arquivo de configuração). Caso a máquina que recebeu o token não tenha dados para transmitir (fila de mensagens e arquivos vazia), o token será enviado para a próxima máquina do anel (máquina a direita - IP configurado no arquivo de configuração). Caso contrário, a primeira mensagem ou arquivo é retirado da fila e é enviado para a máquina a sua direita, ou seja, os dados também devem seguir a ordem do anel.

Os dados enviados deverão retornar à máquina origem e somente depois disso o token poderá ser enviado para a próxima estação.

Quando a máquina origem enviar um pacote de dados, um campo no cabeçalho do pacote deverá ser marcado como “nãocopiado”. Então, se esse pacote retornar para a máquina origem com uma das seguintes configurações:

- “nãocopiado”: significa que a máquina destino não se encontra na rede ou está desligada. Neste caso, uma mensagem na tela deve informar o ocorrido, a mensagem ou arquivo deve ser retirada da fila e o token deve ser transmitido para a próxima máquina do anel;
- “erro”: significa que a máquina destino identificou um erro no pacote e o mesmo deverá ser retransmitido pela origem. Neste caso, uma mensagem na tela deve informar o ocorrido, o token deve ser transmitido para a próxima máquina do anel, e a mensagem ou arquivo não deve ser retirada da fila, sendo retransmitida na próxima passagem do token;
- “OK”: significa que o pacote foi recebido corretamente pela máquina destino. Neste caso, uma mensagem na tela deve informar o ocorrido, a mensagem ou arquivo deve ser retirada da fila e o token deve ser transmitido para a próxima máquina do anel.

A aplicação deve implementar um módulo de inserção de falhas que force as máquinas a marcar os pacotes com “erro” pela máquina destino. Este módulo deve trabalhar com alguma probabilidade e informar que o pacote recebido chegou com “erro” ou “OK”.

Além disso, deverá ser implementada uma fila em cada máquina. Esta fila poderá estar vazia ou não. A fila poderá conter até 10 arquivos ou mensagens. Para cada arquivo ou mensagem adicionada, deve ser armazenado também o apelido da máquina destino.

Os serviços de envio de dados oferecidos devem contemplar duas formas de transmissão:

- *Unicast*: envia o pacote para um único destino;
- *Broadcast*: envia o pacote para todas as máquinas da rede usando o apelido TODOS. Neste caso, o módulo de inserção de falhas deve manter o pacote em “nãocopiado”.

1. Descrição dos Pacotes

A implementação deve seguir fielmente o formato dos pacotes descrito a seguir, pois durante a apresentação do trabalho, aplicações de grupos diferentes deverão se comunicarem. A correta interação entre as diferentes implementações fazem parte da avaliação do trabalho.

1.1 Token

O token será formado por uma sequência numérica em formato string e terá o valor 1234, como mostra o exemplo a seguir:

1234

1.2 Pacote de dados

Um pacote de dados é formado por outra sequência numérica em formato string e terá o valor 2345. Neste caso, o valor 2345 será seguido de um ';' e pelos campos:

<controle de erro>:<apelido de origem>:<apelido do destino>:<tipo de dado>:<mensagem ou dados do arquivo>.

O campo <tipo de dado> indica o conteúdo do dado, sendo A para arquivo e M para mensagem.

Exemplo:

2345;naocopiado:Bob:Alice:M:Oi Mundo!

Ao receber um pacote de dados, a estação verifica se o mesmo é endereçado a ela, verificando o apelido do destino. Caso não seja, este pacote deve ser enviado para seu vizinho da direita. Caso o pacote seja para ela, a aplicação deve imprimir o apelido da origem e a mensagem, e deve também enviar o pacote de volta, alterando o campo `naocopiado` para `OK` ou `erro`. Caso o pacote de dados contenha um arquivo, o mesmo deve ser salvo na máquina.

Caso o pacote de dados seja recebida por quem a originou (o apelido de origem é igual ao seu), será necessário verificar o controle de erro. Ao receber um pacote com este campo em `naocopiado` ou `OK`, um token deve ser enviado para seu vizinho da direita. Caso o pacote venha com `erro`, o mesmo deve ser retransmitida apenas uma vez na rede, trocando o `erro` por `naocopiado`, e enviando a mensagem para a máquina a sua direita na próxima passagem do token.

2. Visualização e depuração

A demonstração deverá acontecer, no mínimo, em 3 máquinas.

Deve ser possível:

- Especificar, a qualquer momento, uma mensagem ou arquivo a ser enviado por uma máquina;

- Visualizar onde o token e o pacote de dados se encontram durante a execução do programa;
- Avisar quando houver retransmissões;
- Saber o que está acontecendo.

Regras Gerais

Grupos: Até 3 componentes.

Data de entrega e apresentação: **05/11**

Obs.: Todos participantes devem estar presentes

Entrega final:

- Texto descrevendo a estrutura da solução dada, envolvendo estruturas de dados, threads, classes, mecanismos de sincronização utilizados, etc.
- Código fonte comentado.