



Everton Agilar

Jáder Adiel Schmitt

TRABALHO DE SISTEMAS DISTRIBUÍDOS

DESENVOLVIMENTO DE UM SOFTWARE

CLIENTE/SERVIDOR EM C

DO TIPO CHAT

Santa Maria, RS

2010

1. Objetivo do Trabalho

Desenvolver um software cliente/servidor do tipo “chat” para aplicar os conhecimentos vistos em sala de aula na disciplina de Sistemas Distribuídos referente aos conteúdos de Threads, Processos, Sincronização de Processos e Sockets em linguagem C.

2. Linguagem de Programação Utilizada

Linguagem C e compilador gcc disponível em <http://gcc.gnu.org/>.

3. Sistema Operacional

Sistema operacional Linux.

4. Orientações para Execução do Software

Para executar o servidor basta invocar no prompt o seguinte comando:

```
:> ./servidor
```

Para executar o cliente (ou os clientes) basta invocar no prompt o seguinte comando:

```
:> ./cliente localhost
```

Onde *localhost* é o nome ou IP do host onde o servidor do chat está executando. O servidor pode estar sendo executado no mesmo computador ou em um computador diferente via TCP/IP.

5. Funcionamento do Servidor

Ao iniciar o servidor do chat através do prompt de comando, o servidor chat automaticamente cria o socket para o servidor. O socket é do tipo Internet. Foi utilizado conexão TCP para fornecer um fluxo confiável entre o servidor e os clientes. Se ocorrer um erro ao criar o socket do servidor, será impresso uma mensagem de erro na tela e o programa será abortado.

Após criar o socket, será realizado o Bind do socket do servidor com um endereço de rede. Caso ocorra um erro, será impresso uma mensagem de erro na tela e o programa será abortado.

Se não ocorrer nenhum erro na chamada a *Bind*, o servidor marca a si mesmo como executando. O flag *server_running* é um sinalizador utilizado para saber se o servidor está executando e não está em processo de finalização. As threads que serão instanciadas para atender a processos clientes sempre verificam este flag, e caso o servidor esteja sendo finalizado, as threads encerram seus serviços.

Depois de sinalizar que o servidor está executando, é invocado a função *createListener*, responsável por instanciar a thread que escuta por conexões de clientes. Esta thread coloca o servidor em modo *LISTENER* e fica aguardando conexões na porta 3000.

Ao receber uma solicitação de cliente, é criada uma nova thread, através da função *createServidorChat*. Esta função cria um thread responsável por atender a nova conexão ao chat. Após a chamada a *createServidorChat*, o servidor volta imediatamente ao modo *LISTENER*, através do qual, fica aguardando novas conexões.

Durante o estado de *LISTENER*, o servidor pode receber inúmeras conexões de clientes. Cada nova conexão é armazenada em uma estrutura do tipo lista encadeada, através do qual, o servidor tem como saber quem está conectado. Para cada conexão, o servidor armazena o IP, nome do usuário, o descritor do socket cliente, um flag indicando se o cliente está conectado, etc.

Ao receber uma mensagem, o servidor chat automaticamente repassa para todos os clientes conectados, a mensagem recebida, exceto para o próprio emissor da mensagem.

Toda vez que um cliente entra no chat ou sai do chat, o servidor marca a conexão como inativa. Através do flag conectado, pode-se saber quais clientes estão conectados.

6. Funcionamento do Cliente

O programa cliente tem um funcionamento semelhante ao programa servidor. Para executar, o usuário deve passar como parâmetro o nome do servidor. Caso não seja passado nenhum nome, é informado um erro no console e o programa é abortado.

Após inicializar, é criado um socket para conexão com o servidor e invocado a função *createServidorMensageiro*, responsável por criar a thread que trata as mensagens recebidas dos usuários que estão conectados ao chat. Após, é invocado a função *iniciarConversacao*, através do qual, é possível ao usuário digitar mensagens no chat e enviar para os demais usuários. Para sair do programa, o usuário deve digitar o comando “quit”. Caso não for possível enviar uma mensagem digitada para o servidor, será exibido um erro na tela e encerrado o programa imediatamente.