

Redes – TP3

Gabriel Bastos – 2016058204

Novembro de 2019

Sumário

1	Proposta	1
1.1	Número de clientes	1
1.2	Tamanho das mensagens	1
1.3	Mensagens	2
1.4	Nome de usuário	2
2	Projeto	2
2.1	Mensagens	2
2.1.1	Cliente → Servidor	2
2.1.2	Servidor → Cliente	3
2.2	Destinatário	3
3	Comandos	3
3.1	Desconectar	3
3.2	Nome de usuário	3
3.3	Listar usuários	3
3.4	Mensagem <i>broadcast</i> :	3
3.5	Mensagem <i>unicast</i> :	3

1 Proposta

Com o objetivo de adquirir experiência e conhecimento em aplicações reais que lidam com redes, extendi os requisitos do trabalho de forma a se assemelhar mais com um programa real, que lida com diversas condições de rede e usuários sob os quais não se têm controle.

1.1 Número de clientes

O número máximo de clientes será sempre limitado pelo número de portas no protocolo de rede. Apesar disso, julguei interessante eliminar da aplicação este tipo de restrição. Recebendo os clientes de forma dinâmica, é possível conter essa limitação aos recursos do sistema, e não no código. Desta forma, o número de clientes simultâneos não deve ser limitado por uma constante no programa.

1.2 Tamanho das mensagens

O tamanho das mensagens também deve ser dinâmico, de forma a fazer o melhor uso de banda e memória. Contudo, uma limitação se faz necessária: para evitar que clientes maliciosos possam executar um ataque *DDoS* através de mensagens muito grandes, um tamanho máximo deve ser estipulado. Este tamanho deve ser facilmente alterável em tempo de compilação, de forma a permitir mensagens maiores, caso necessário.

1.3 Mensagens

Aplicações reais devem ser capazes de lidar, em uma única operação de leitura, com múltiplas mensagens consecutivas, bem como mensagens divididas e incompletas. É possível que, caso um cliente envie dois pacotes, um sofra atraso e chegue separado. De forma semelhante, uma congestão na rede pode fazer com que vários pacotes, mesmo que enviados separadamente, cheguem juntos. O servidor deve ser capaz de lidar com tais situações, de forma que não ocorra perda de dados.

1.4 Nome de usuário

Com o objetivo de permitir maior flexibilidade aos clientes, deve ser permitido usuários anônimos. Estes usuários não serão capazes de receber mensagens diretas, mas participarão de *broadcasts*, e poderão receber a lista de usuários conectados.

2 Projeto

De forma a se obter um melhor aproveitamento dos recursos computacionais, o servidor utiliza a chamada de sistema `poll` para lidar com múltiplos clientes. Optei por essa chamada por ser uma versão moderna e robusta da chamada `select`, indicada na especificação original.

2.1 Mensagens

Para permitir mensagens consecutivas, se faz necessária a delimitação destas. Optei por utilizar os caracteres de controle da table unicode ¹ como delimitadores. Os caracteres utilizados são:

Caractere	Significado
SOH	Start of Heading
STX	Start of Text
IND	Index
EOT	End of transmission
ENQ	Enquiry character
NAK	Negative acknowledge
US	Unit Separator
PM	Private Message

2.1.1 Cliente → Servidor

Definição de nome: Mensagem para definir o nome do usuário.

Um nome vazio indica anonimidade.

SOH IND <name> EOT

Lista de usuários: Mensagem para requerer a lista de usuários.

SOH ENQ EOT

Broadcast: Mensagem de texto para todos usuários conectados.

SOH STX <texto> EOT

Unicast: Mensagem de texto para um usuário específico.

SOH PM <destinatário> STX <texto> EOT

¹[Unicode control codes](#)

2.1.2 Servidor → Cliente

Erro: Mensagem de erro, resposta à uma requisição inválida do cliente.

SOH NAK <código de erro> EOT

Código de erro	Significado
0x01	Nome inválido (em uso)
0x02	Destinatário inválido

Lista de usuários: Mensagem com a lista de usuários.

SOH ENQ <usuário> US <usuário> US <usuário> ... EOT

Texto: Mensagem de texto.

SOH PM <remetente> STX <texto> EOT

2.2 Destinatário

Para a identificação rápida do destinatário em mensagens *unicast*, um índice de clientes foi implementado. Desta forma, não é necessário buscar na coleção de clientes o alvo da mensagem.

3 Comandos

3.1 Desconectar

Comando:

exit

3.2 Nome de usuário

Ao entrar no servidor, o usuário é inicialmente anônimo. Seu nome pode ser alterado a qualquer momento, quantas vezes desejar.

Comando:

name;<nome>

Para retornar à anonimidade:

name;

3.3 Listar usuários

Comando:

users

3.4 Mensagem *broadcast*:

Comando:

all;<mensagem>

3.5 Mensagem *unicast*:

Comando:

uni;<destinatário>;<mensagem>