Trabalho 1 de MC833 — Programação em Redes de Computadores

Servidor de informação baseado em localização com socket TCP e UDP

Pedro Henrique Moura Andere, 10xxxx, turma A
Raul Rabelo Carvalho, 105607, turma A

21 de Abril de 2014

1 Casos de uso do sistema

O sistema neste Trabalho utiliza um modelo de cliente/servidor, no qual existe um servidor de informação que acessa e processa os dados dos estabelecimentos cadastrados e um conjunto de clientes que requerem parte ou a totalidade dos dados para serem exibidos aos usuários.

O sistema aceita comandos do usuário pelo *software* cliente. Os comandos são:

posicao <x>,<y> — informa a posição do usuário ao servidor.

posicao <x> <y> — idem o comando anterior.

categorias — exibe uma lista com os nomes das categorias de estabelecimentos existentes no sistema.

listar todos — exibe uma lista com o identificador (*ID*) e nome de todos os estabelecimentos.

buscar todos — idem o comando anterior.

listar perto todos — exibe uma lista com o *ID* e o nome dos estabelecimentos a menos de 100m da posição atual do usuário.

buscar perto todos — idem o comando anterior.

listar perto categoria <categoria> — exibe uma lista com o *ID* e o nome dos estabelecimento a menos de 100 metros da posição atual do usuário e que sejam da categoria `<categoria>`.

listar perto categoria <categoria> — idem o comando anterior.

listar categoria <categoria> — exibe uma lista com o *ID* e o nome dos estabelecimentos da categoria `<categoria>`.

buscar categoria <categoria> — idem o comando anterior.

info <ID> — exibe as informações do estabelecimento com *ID* `<ID>`.

votar <ID> <nota> — registra nas informações do estabelecimento com *ID* `<ID>` uma nota (inteiro `<nota>` de 1 a 10) dada pelo usuário.

sair — encerra o cliente.

1.1 Caso de uso: buscar por um estabelecimento próximo de uma categoria específica

Ator:
Usuário do cliente.

Escopo:
Sistema cliente/servidor.

Fluxo básico:

1. Executar o cliente com os argumentos endereço IP e porta do servidor.
2. Informar ao servidor a posição atual do cliente: `posicao x,y`.
3. Listar as categorias de estabelecimentos disponíveis: `categorias`.
4. Listar os estabelecimentos próximos da posição atual na categoria desejada: `listar perto categoria <categoria>`.
5. Exibir as informações do estabelecimento escolhido: `info <ID>`.
6. Encerrar o cliente: `sair`.

Extenção:
Após o passo 5, o usuário pode dar uma nota ao estabelecimento: `votar <ID> <nota>`.

1.2 Caso de uso: buscar por um estabelecimento próximo

Ator:
Usuário do cliente.

Escopo:
Sistema cliente/servidor.

Fluxo básico:

1. Executar o cliente com os argumentos endereço IP e porta do servidor.
2. Informar ao servidor a posição atual do cliente: `posicao x,y`.
3. Listar os estabelecimentos próximos da posição atual: `listar perto todos`.
4. Exibir as informações do estabelecimento escolhido: `info <ID>`.
5. Encerrar o cliente: `sair`.

Extenção:
Após o passo 4, o usuário pode dar uma nota ao estabelecimento: `votar <ID> <nota>`.

1.3 Caso de uso: buscar por um estabelecimento de uma categoria específica

Ator:
Usuário do cliente.

Escopo:
Sistema cliente/servidor.

Fluxo básico:

1. Executar o cliente com os argumentos endereço IP e porta do servidor.
2. Listar as categorias de estabelecimentos disponíveis: `categorias`.
3. Listar os estabelecimentos na categoria desejada: `listar categoria <categoria>`.
4. Exibir as informações do estabelecimento escolhido: `info <ID>`.
5. Encerrar o cliente: `sair`.

Extenção:
Após o passo 4, o usuário pode dar uma nota ao estabelecimento: `votar <ID> <nota>`.

1.4 Caso de uso: listar todos os estabelecimentos

Ator:
Usuário do cliente.

Escopo:
Sistema cliente/servidor.

Fluxo básico:

1. Executar o cliente com os argumentos endereço IP e porta do servidor.
2. Listar os todos os estabelecimentos cadastrados: `listar todos`.
3. Encerrar o cliente: `sair`.

Extenção:
Após o passo 2, o usuário pode:

- Exibir as informações de um estabelecimento: `info <ID>`.
- Dar uma nota a um estabelecimento: `votar <ID> <nota>`.

2 Armazenamento e estruturas de dados

Cada estabelecimento cadastrado no servidor de informação foi modelado por uma estrutura de dados na linguagem C (`struct item`) definida como um tipo `item_t`. A estrutura contém cinco campos do tipo inteiro e três do tipo vetor de caracteres. Os campos são os seguintes:

id
: (Inteiro.) O identificador único de cada estabelecimento.

posx
: (Inteiro.) A coordenada no eixo-x da posição do estabelecimento.

posy
: (Inteiro.) A coordenada no eixo-y da posição do estabelecimento.

categoria
: (Vetor de caracteres.) O nome da categoria ao qual o estabelecimento pertence.

nome
: (Vetor de caracteres.) O nome do estabelecimento.

endereco
: (Vetor de caracteres.) O endereço do estabelecimento.

pontuacao
: (Inteiro.) O total acumulado das notas dadas ao estabelecimento.

votos
: (Inteiro.) O total de vezes que o estabelecimento recebeu uma nota.

Todos os inteiros são inteiros com sinal (apesar de que nenhum dos campos necessita armazenar valores negativos) e os vetores de caracteres têm todos comprimento máximo de 256 posições. Foi escolhido dar um tamanho máximo para esses campos, pois torna a implementação mais simples e menos propensa a erros; além disso, bancos de dados comumente trabalham com campos de tamanho fixo.

Um vetor do tipo `item_t` pode empregado para guardar na memória para uso pelo servidor uma parte ou a totalidade (como é o caso da implementação feita para este Trabalho) do conjunto de estabelecimentos cadastrados.

Para o armazenamento permanente dos dados em disco, foi usado um arquivo de texto simples contendo por linha as informações de um estabelecimento, com cada campo separado por um ponto-e-vírgula. A ordem dos campos é a mesma da lista acima. A única exceção é a primeira linha que contém somente um inteiro com o número de estabelecimentos contidos no arquivo. É importante que esteve número esteja correto, pois o servidor irá ignorar qualquer entrada além da quantidade dada na primeira linha do arquivo, e irá encerrar com erro se o número de entradas for menor que esta quantidade.

Outra estrutura de dados empregada pelo servidor é a estrutura do tipo `res_busca_t` que contém dois campos: um inteiro para guardar identificadores e um vetor de caracteres (também com 256 posições) para armazenar nomes. Um vetor deste tipo é usado para passar o resultado de buscas sobre o conjunto de estabelecimentos para as funções de impressão e comunicação do servidor.