**TERMINAL.H**

A biblioteca terminal.h contém as funções para manipular os terminais do netmap. Esta é dividida em funções principais, que atendem requisitos de funcionamento do netmap, e funções auxiliares, que facilitam a comunicação com as implementações de outras bibliotecas.

A lista de terminais é do tipo sem sentinela e simplesmente encadeada. A escolha desse tipo de lista parte da sua simplicidade e da variedade que trás ao projeto. Abaixo seguem descrições dos elementos principais presentes na implementação da biblioteca “terminal.h”.

**Estrutura do terminal (struct terminal)**

|  |  |
| --- | --- |
| char\*  nome | CelRot\*  rot |
| char\* localizacao |

**Estrutura da célula de uma lista/célula de terminais (struct celTerm)**

Célula de roteador na lista de roteadores ao qual o terminal está conectado (struct celRot)

Nome do terminal

Nome da localização

Uma lista sem sentinela, simplesmente encadeada não circular é manipulada com o auxílio de um ponteiro para a primeira posição da lista. Portanto, a lista e suas células compartilham a mesma estrutura.

Célula seguinte (struct celTerm)

Tipo terminal (struct terminal)

|  |  |
| --- | --- |
| Terminal\* term | CelTerm\* prox |

**CadastraTerminal**

A função adiciona um terminal à lista de terminais do netmap. Ela recebe como parâmetros o nome do terminal (char\*), o nome da localização do terminal (char\*) e o ponteiro para lista de terminais (CelTerm\*). A função retorna a lista de terminais atualizada, pois esta é sem sentinela e o endereço da lista é sujeito à mudança. Após execução: uma estrutura terminal deve ter sido criada e preenchida com os dados fornecidos; uma célula criada e encadeada na lista preenchida com ponteiro para o terminal.

Ao iniciar, duas células (CelTerm\*) são declaradas, uma auxiliar apontando para a mesma célula para a qual a lista fornecida aponta e outra alocada dinamicamente que será encadeada na lista. Em seguida, chama-se a função interna criaTerminal que aloca um tipo terminal preenchido com nome e localização fornecidos e que não está conectado a roteador.

Para facilitar a impressão do arquivo “saida.dot”, novas células são encadeadas ao final da lista de terminais. Portanto, a célula nova recebe um ponteiro para o terminal criado e seu ponteiro para a próxima célula recebe “NULL”.

A função então verifica se a lista fornecida está vazia. Caso esteja, o ponteiro para a lista passa a apontar para a nova célula, tornando-a o primeiro e único elemento da lista. Caso não esteja, a célula auxiliar percorre a lista até a última posição, permitindo o encadeamento da nova célula ao fim da lista.

**RemoveTerminal**

A função remove um terminal existente da lista de terminais do netmap. Ela recebe como argumentos o nome do terminal (char\*) e o ponteiro para a lista de terminais (CelTerm\*). A função retorna a lista de terminais atualizada, pois esta é sem sentinela e o endereço da lista é sujeito à mudança. Após execução: o tipo terminal cujo nome é idêntico ao nome fornecido será liberado; a célula que aponta para o tipo terminal em questão é desencadeada da lista e liberada.

Ao iniciar, são declaradas três células auxiliares: uma que recebe a célula da lista fornecida que aponta para o terminal cujo nome foi fornecido através da função auxiliar BuscaTerminal; outra que aponta para a primeira posição da lista fornecida; e outra não inicializada encarregada de reter a posição anterior a medida que a lista é percorrida.

A lista de terminais é percorrida pelas células auxiliares. Uma vez que o terminal não é encontrado na lista, uma mensagem de erro é escrita em um vetor que é passado como parâmetro para a função EscreveLOG, que imprime mensagens de erro no arquivo “log.txt”.

Caso o terminal seja encontrado, a função verifica se sua célula é a primeira da lista. Caso seja, a lista de terminais passa a apontar para a próxima posição (que pode ser “NULL”). Caso não seja, a posição anterior à célula a ser removida passa a apontar para a próxima posição. Por fim, célula e seu tipo terminal são liberados.

**ConectaTerminal**

A função conecta um terminal a um roteador, ambos existentes no netmap. Ela recebe como argumentos o nome do terminal (char\*), o nome do roteador (char\*), o ponteiro para a lista de terminais (CelTerm\*) e o ponteiro para a lista de roteadores (char\*). Não há retorno pois a lista de terminais permanece inalterada. Após execução: o tipo terminal do terminal cujo nome foi fornecido agora aponta para a célula do roteador cujo nome foi fornecido na lista de roteadores.

Ao iniciar, é declarada uma célula de terminal auxiliar que recebe um ponteiro para a célula do terminal cujo nome foi fornecido através da função BuscaTerminal, que retorna “NULL” caso não encontre o terminal passado. Se a célula for “NULL”, o terminal não existe no netmap e uma mensagem de erro é então criada e passada como parâmetro para a função EscreveLOG, que imprime a mensagem em “log.txt”.

Caso o terminal seja encontrado, a função verifica se o roteador existe no netmap, e resgata sua célula caso exista através da declaração de uma célula auxiliar para roteador e chamada da função BuscaRoteador. Novamente, se a célula receber “NULL”, uma mensagem de erro é criada e passada como parâmetro para a função EscreveLOG, que imprime a mensagem em “log.txt”.

Se terminal e roteador existem no netmap, o tipo terminal do terminal cujo nome foi fornecido passa a apontar para a célula do roteador cujo nome foi fornecido na lista de roteadores.

**DesconectaTerminal**

A função desconecta um terminal existente no netmap de seu roteador. Ela recebe como argumentos o nome do terminal (char\*) e o ponteiro para a lista de terminais (CelTerm\*). Não há retorno pois a lista de terminais permanece inalterada. Após execução: o tipo terminal do terminal cujo nome foi fornecido agora aponta para “NULL”, indicando que está desconectado.

Ao iniciar, é declarada uma célula de terminal auxiliar que recebe um ponteiro para a célula do terminal cujo nome foi fornecido através da função BuscaTerminal, que retorna “NULL” caso não encontre o terminal passado. Se a célula for “NULL”, o terminal não existe no netmap e uma mensagem de erro é então criada e passada como parâmetro para a função EscreveLOG, que imprime a mensagem em “log.txt”.

Caso a célula do terminal seja encontrada, o tipo terminal para o qual ela aponta passa a apontar para um roteador “NULL”, indicando que o terminal está desconectado.

**EnviarPacotesDados**

A função avalia a possibilidade de envio de pacotes de dados de um terminal para outro. Ela recebe como argumentos os nomes de ambos os terminais (char\*) e a lista de terminais (CelTerm\*). Não há retorno. Após execução: nomes dos terminais e resultado SIM ou NÃO escrito em arquivo “saida.txt”.

Ao inciar, declara células auxiliares para os terminais passados e, com o auxílio da função BuscaTerminal, verifica se os terminais existem no netmap. Caso algum terminal não exista, uma mensagem de erro é criada e passada como parâmetro para a função EscreveLOG, que imprime a mensagem em “log.txt”.

A função então verifica se os terminais passados estão conectados a algum roteador, caso algum não esteja, a comunicação entre eles não é possível, portanto uma mensagem de resultado NÃO é criada e passada como parâmetro para a função EscreveSAIDA, que imprime a mensagem em “saida.txt”.

Caso ambos estejam conectados a um roteador, o nome de cada roteador é resgatado e comparado. Se o nome for o mesmo, ambos estão conectados ao mesmo roteador e o envio de dados é possível, portanto uma mensagem SIM é criada e passada como parâmetro para a função EscreveSAIDA. Caso estejam conectados a roteadores diferentes, é chamada a funcaoBusca, detalhada na descrição de “roteador.h”, que retorna 1 para SIM e 0 para NÃO. O procedimento de escrita do resultado segue de acordo com o retorno da funcaoBusca.

**FrequenciaTerminal**

A função conta a quantidade de terminais cadastrados no netmap que são de uma determinada localização. Ela recebe como argumentos o nome da localização (char\*) e a lista de terminais. Não há retorno. Após execução: o nome da localização e sua frequência absoluta de ocorrência na lista de terminais são escritos no arquivo de texto “saida.txt”.

Ao iniciar, é declarada uma variável do tipo “int” encarregada de armazenar o valor da contagem de ocorrências de terminais pertencentes à localização fornecida (inicialmente zero). Também é declarada uma célula auxiliar de terminal inicializada apontando para a primeira posição da lista de terminais.

A função anda pela lista com a célula auxiliar e compara a localização de cada terminal apontado por cada célula com a localização desejada, realizando a contagem com o auxílio da variável declarada no início.

Por fim, uma mensagem contendo o nome da localização e sua frequência é passada como argumento para a função EscreveSAIDA, que imprime a mensagem em “saida.txt”.