Relatório Trabalho Prático 1

Redes de Computadores

Introdução

Neste trabalho foi criado um jogo de forca simplificado utilizando aplicações servidor e cliente. O servidor escolhe uma palavra e envia seu tamanho ao cliente, que então envia palpites de letras ao servidor, que responde positiva ou negativamente com suas posições de ocorrência na palavra. As falhas do cliente não são contabilizadas e ele não pode "chutar" a resposta. O tema escolhido para as palavras foi nomes de frutas.

Implementação

Foram escritos cinco arquivos para implementar as funções requeridas, na linguagem C. O servidor e o usuário, que possuem apenas a função `main` para a execução de suas atividades, além de um arquivo com outras funções necessárias para sua execução e seu cabeçalho. Por fim, há um makefile que compila os programas e gera os executáveis.

O arquivo `cliente.c` executa a função do cliente. Primeiramente cria um storage para o socket do cliente e o inicializa com uma função que está implementada no arquivo `funcoes.c` e depois inicializa o próprio socket. Se conecta, então, ao servidor e recebe a mensagem de início de jogo. O cliente então entra em um loop para enviar os palpites e receber as respostas do servidor, verificando sempre o código do tipo de mensagem recebida, para identificar o fim de jogo. Quando acerta todas as letras da palavra, o cliente é encerrado. A função do cliente é bastante interativa com o usuário durante sua execução, por exemplo sempre imprimindo na tela a palavra com as letras inseridas até então, para deixar a experiência do jogo mais dinâmica.

O arquivo `servidor.c` executa a função do servidor. É criado um storage para o socket do servidor, que é inicializado em uma função que está em `funcoes.c` e depois é inicializado o próprio socket. A porta passada como entrada é atribuída ao servidor, e então ele passa a esperar conexões. Existe um loop para conexão de clientes, em que um storage e um socket são criados e inicializados para o cliente, depois o servidor aceita a conexão, gera uma palavra aleatória usando uma função do arquivo `funcoes.c` e envia a mensagem de início de jogo para o cliente. Ele então entra em outro loop para receber e responder os palpites do cliente e encerra a conexão com ele quando acerta toda a palavra. O servidor então volta a aguardar por conexões de outros clientes.

No arquivo `funcoes.c` foram colocadas funções necessárias para os demais arquivos. `logexit` ajuda a encontrar a localização de erros na execução. `client_init` inicializa os atributos do cliente e `server_init` os do servidor. `gera_palavra` escolhe aleatoriamente uma palavra dentre uma lista fornecida em um arquivo `.txt` , colocando-a na string passada como entrada e retorna seu tamanho. O arquivo `funcoes.h` é seu cabeçalho.

Estruturas

Foram utilizadas estruturas `sockaddr_storage` para armazenar os dados da estrutura `sockaddr` tanto para o cliente quanto para o servidor, já que a estrutura `sockaddr` é pequena, e assim evitamos preocupações com seu tamanho durante a execução. Após ser instanciada `sockaddr_storage` passa por um cast para `sockaddr_in` ou `sockaddr_in6`, dependendo do protocolo usado. Essa estrutura é inicializada na função `client_init` ou `server_init`. No caso do servidor, há uma constante `int versao` que deve ser manualmente modificada para 0 ou 1 dependendo da versão do protocolo que se deseja usar (IPv4 ou IPv6).

Para as mensagens trocadas pelo servidor e cliente foram usados vetores de chars, cuja primeira posição sempre contém o tipo da mensagem (1 a 4) e as demais variam de acordo com esse tipo.