DCC023 Redes de Computadores– Turma TE (METATURMA)
Universidade Federal de Minas Gerais – UFMG
Marcelo Costa Monteiro – 2014139487 - Belo Horizonte - 30/09/2020

TRABALHO PRÁTICO 2 - SERVIDOR DNS

Resumo: O trabalho consiste em uma aplicação socket que simula o comportamento de servidores DNS. Ou seja, o servidor é responsável por traduzir nomes de hosts para seus respectivos IPs. Dessa forma, ele recebe instruções via entrada padrão do teclado ou via arquivo e pode adicionar um novo par ip-host, pesquisar por um IP ou adicionar um outro servidor para se comunicar. A aplicação foi desenvolvida na linguagem C, usando padrão POSIX, protocolo UDP e compatível com IPv4 e IPv6.

Executando

Para executar o projeto:

1. Abra o diretório raiz, onde há o Makefile e execute o comando: make
2. Execute o servidor com o comando: ./servidor_dns <porta> <arquivo opcional>
   (ex: ./servidor_dns 51511 file.txt)

Implementação

Para armazenar os hosts e servidores foram utilizados arrays de estruturas de dados host e server. Ao iniciar a função main, cria-se uma thread responsável por receber conexões de outros servidores e respondê-las. Então, faz-se a leitura do arquivo e por fim inicia-se um loop para fazer a leitura do teclado.

A função runCommand foi criada para executar os comandos lidos, seja via arquivo ou via entrada padrão. Ela então separa a string lida usando strtok para identificar o comando e os parâmetros. No caso de "add"e "link", os dados são adicionados aos arrays hosts e servers, respectivamente.

Em caso de "search", utiliza-se a função searchForIP, que recebe o hostname e verifica se este encontra-se no array hosts. Em caso positivo, imprime-se o IP na tela. Em caso negativo, a aplicação tentará conectar-se com os servidores que possui salvos no array servers e, para cada um sequencialmente, fazemos uma requisição.

Esta se dá pela função request_handler que, inicializa o socket, envia a mensagem do host e aguarda um retorno que pode ser o IP ou "-1" e no caso de um não retorno, a função também retornará "-1". Caso o retorno seja o IP, o meso é impresso na tela. Caso o retorno seja "-1" tentamos o próximo servidor. Caso a lista se esgote e o IP não tenha sido encontrado, imprime-se o aviso "Host não encontrado".

As requisições que chegam no servidor são tratadas na thread response_handler. Ela inicializa o socket e fica esperando algo no recvfrom(). Ao receber um nome de host, busca-se no seu array de hosts e caso não encontre ele busca nos demais servidores de forma similar como é feito na função searchForIP, no entanto, o resultado não é impresso na tela e sim enviado de volta para o socket.

Detalhes adicionais

Há constantes definidas no início do arquivo. DATASIZE representa a quantidade máxima de hosts e servers que a aplicação poderá armazenar. STRSIZE representa o tamanho máximo das entradas nome e ip. BUFSIZE representa o tamanho dos buffers que serão utilizados.