****

**Trabalho**

**Prof.ª Renato.**

**Disciplina: Sistemas Operacionais I**

**­Curso: A.D.S – Análise e Desenvolvimento de Sistemas.**

**Turno: Noite.**

**Nome: Cristian Augusto Paulista.**

**RA: 0220481713044**

**Nome: Eduardo Henrique O. Coró.**

**RA: 0220481713038**

**Taquaritinga – SP**

**19/04/2018**

**Desenvolvimento com Socket em C**

O referente programa consiste em duas partes de funcionamento, servidor e o cliente, no qual a aplicação será através de um servidor simples no domínio da internet usando TCP. O número da porta é passado como um argumento.

**Server.c**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

void error(const char \*msg) **/\* Função de exibir erro \*/**

{

perror(msg);

exit(1);

}

int main(int argc, char \*argv[]) **/\* Passado por argumentos \*/**

{

int sockfd, newsockfd, portno; **/\* declaração das variaveis do tipo inteiro \*/**

socklen\_t clilen;

char buffer[255];

struct sockaddr\_in serv\_addr, cli\_addr; **/\* Criando e declarando os elementos da estrutura \*/**

int n;

if (argc < 2) {

fprintf(stderr,"ERROR, no port provided\n"); **/\* verificar se existe conexao ou comunicação \*/**

exit(1); **// efetivamente irá terminar o processo do programa e fechar.**

}

sockfd = socket(AF\_INET, SOCK\_STREAM, 0); **/\* Cria um socket servidor \*/**

if (sockfd < 0)

error("ERROR opening socket**"); // passando valores no parametro da função**

bzero((char \*) &serv\_addr, sizeof(serv\_addr));

portno = atoi(argv[1]);

**/\* Define o endereço do servidor \*/**

serv\_addr.sin\_family = AF\_INET;

serv\_addr.sin\_addr.s\_addr = INADDR\_ANY; **/\* coloca IP automaticamente \*/**

serv\_addr.sin\_port = htons(portno);

**// Liga o socket em um endereço e a uma porta específica**

if (bind(sockfd, (struct sockaddr \*) &serv\_addr, sizeof(serv\_addr)) < 0) **// verificar se existe ligação do socket com o endereço ou algo espefico.**

error("ERROR on binding"); **// passando valores no parametro da função**

listen(sockfd,5);

clilen = sizeof(cli\_addr);

newsockfd = accept(sockfd, (struct sockaddr \*) &cli\_addr, &clilen);

if (newsockfd < 0)

error("ERROR on accept"); **// passando valores no parametro da função**

while(1)

{

bzero(buffer,256);

n = read(newsockfd,buffer,255);

if (n < 0) error("ERROR reading from socket"); **// verificando se têm ligação**

printf("Client: %s\n",buffer); **// Envia os dados**

bzero(buffer,256);

fgets(buffer,255,stdin);

n = write(newsockfd,buffer,strlen(buffer));

if (n < 0) error("ERROR writing to socket");

int i=strncmp("Bye" , buffer, 3);

if(i == 0)

break;

}

**// fecha conexão do socket servidor**

close(newsockfd);

close(sockfd);

return 0;

}

**Cliente.c**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <netdb.h>

void error(const char \*msg) **/\* Função de exibir erro \*/**

{

perror(msg);

exit(0);

}

int main(int argc, char \*argv[]) **/\* Passado por argumentos \*/**

{

int sockfd, portno, n; **/\* declaração das variaveis do tipo inteiro \*/**

struct sockaddr\_in serv\_addr; **// Especifica um endereço para o socket**

struct hostent \*server;

char buffer[256];

if (argc < 3) {

fprintf(stderr,"usage %s hostname port\n", argv[0**]); /\* verificar conexão e exibir o endereço da port \*/**

exit(0);

}

portno = atoi(argv[2]);

sockfd = socket(AF\_INET, SOCK\_STREAM, 0); **/\* Cria um socket servidor \*/**

if (sockfd < 0)

error("ERROR opening socket"); **// passando valores no parametro da função**

server = gethostbyname(argv[1]);

if (server == NULL) { **// Garantindo se ocorre conexão com o host**

fprintf(stderr,"ERROR, no such host\n");

exit(0);

}

bzero((char \*) &serv\_addr, sizeof(serv\_addr));

serv\_addr.sin\_family = AF\_INET;

bcopy((char \*)server->h\_addr, (char \*)&serv\_addr.sin\_addr.s\_addr,server->h\_length);

serv\_addr.sin\_port = htons(portno);

**// conexão do socket**

if (connect(sockfd,(struct sockaddr \*) &serv\_addr,sizeof(serv\_addr)) < 0) **// verificar se existe erro de ligação**

error("ERROR connecting"); **// passando valores no parametro da função**

printf("Client: ");

while(1)

{

**// envia dados**

bzero(buffer,256);

fgets(buffer,255,stdin);

n = write(sockfd,buffer,strlen(buffer));

if (n < 0) **// verificar se têm passagem de write**

error("ERROR writing to socket");

bzero(buffer,256);

n = read(sockfd,buffer,255);

if (n < 0) **// verificar se têm passagem de read**

error("ERROR reading from socket");

printf("Server : %s\n",buffer);

int i = strncmp("Bye" , buffer , 3);

if(i == 0) **// finalizando o a repetição**

break;

}

**// fecha o socket**

close(sockfd);

return 0;

}