

Apr 08, 14 23:01

eco_serv_select_tcp.c

Page 1/2

```

#include <sys/socket.h>
#include <netinet/in.h>
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/select.h>

#define TAM 64

int main(int argc, char *argv[]) {
    int i, s, s_conec, lido, ndesc;
    unsigned int tam_dir;
    struct sockaddr_in dir, dir_cliente;
    char buf[TAM];
    int opcion=1;
    fd_set desc_sockets;
    fd_set desc_sockets_copia;

    if ((s=socket(PF_INET, SOCK_STREAM, IPPROTO_TCP)) < 0) {
        perror("Error creating socket");
        return 1;
    }

    if (setsockopt(s, SOL_SOCKET, SO_REUSEADDR, &opcion, sizeof(opcion))<0){
        perror("Setsockopt error");
        return 1;
    }

    // configurando para o socket escutar em
    // qualquer endereco IP
    dir.sin_addr.s_addr=INADDR_ANY;
    dir.sin_port=htons(56789);
    dir.sin_family=PF_INET;
    if (bind(s, (struct sockaddr *)&dir, sizeof(dir)) < 0) {
        perror("Bind error");
        close(s);
        return 1;
    }

    if (listen(s, 5) < 0) {
        perror("Listen error");
        close(s);
        return 1;
    }

    FD_ZERO(&desc_sockets);
    // "colocou" o descritor s encarregado de receber
    // as conexoes no conjunto de descritores
    FD_SET(s, &desc_sockets);

    while (1) {
        desc_sockets_copia=desc_sockets;

        // tratamento de erro; o select e' feito sobre 1024
        // descritores, que e' o valor de FD_SETSIZE
        if ((ndesc=select(FD_SETSIZE, &desc_sockets_copia,
                        NULL, NULL, NULL))<0) {
            perror("Select error");
            close(s);
            return 1;
        }
        // se o descritor que recebe as conexoes esta' marcado para
        // receber dados,
        if (FD_ISSET(s, &desc_sockets_copia)) {
            // tire ele do conjunto de descritores marcados
            ndesc--;

            // crie uma nova conexao para os dados a serem recebidos
            tam_dir=sizeof(dir_cliente);

```

Apr 08, 14 23:01

eco_serv_select_tcp.c

Page 2/2

```

        if ((s_conec=accept(s, (struct sockaddr *)&dir_cliente,
                        &tam_dir))<0){
            perror("Accept error");
            close(s);
            return 1;
        }
        // incluindo o novo descritor no conjunto que deve ser
        // tratado no proximo loop
        FD_SET(s_conec, &desc_sockets);
    }

    // para cada posicao i do conjunto de descritores, enquanto
    // ndesc for positivo
    for (i=0; ndesc; i++) {
        // se o descritor i esta' setado, trate ele
        if (FD_ISSET(i, &desc_sockets_copia)) {

            // um descritor a menos ainda nao tratado
            // echo caso algo tenha sido enviado
            ndesc--;

            if ((lido=read(i, buf, TAM))>0) {
                // tratamento de erro da resposta
                if (write(i, buf, lido)<0) {
                    perror("Write error");
                    close(s);
                    return 1;
                }
            }
            // tratamento de erro na leitura
            if (lido<0) {
                perror("Read error");
                close(s);
                return 1;
            }
            // se nao ha' mais nada a ser dado echo,
            // feche a conexao
            if (lido==0) {
                close(i);
                FD_CLR(i, &desc_sockets);
            }
        }
    }

    close(s);

    return 0;
}

```