

```
struct MyPkt {
    uint32_t len;
    char payload[0];
}
```



```
O adresa in socket.h = un IP + un port
struct sockaddr_in // IPv4
Toate apelurile de functii primesc adresele ca (struct sockaddr *) + sizeof(adresa_ta).
Toate functiile intorc < 0 in caz de eroare

errno

int sockfd = socket(...);
if(sockfd < 0) {
    perror("ceva"); // ceva: No resources available
    exit(-1);
}

DIE((sockfd = socket(...)) < 0, "ceva");

// Sever: obtine un socket (socket()), bind, recvfrom, sendto
// Client Obtine un socket, [bind], sendto, recvfrom

// Explicatii

int sockfd = socket(fam_adrese, tip_socket, protocol = 0):
fam_adrese = AF_INET // IPv4
tip_socket = SOCK_DGRAM // UDP
SOCK_STREAM // TCP
```

```

SOCK_RAW //

htonl(int), htons(unsigned short)
ntohl(int)

// Server
int listenfd = socket(AF_INET, SOCK_STREAM, 0);

if(listenfd < 0) {

    perror("socket");

    exit(-1);
}

struct sockaddr_in serveraddr;
serveraddr.sin_family = AF_INET;
serveraddr.sin_port = htons((unsigned short)portno);
serveraddr.sin_addr.s_addr = htonl(INADDR_ANY); // pe adresele de pe placa de retea (this host)

int enable = 1;

rc = setsockopt(sockfd, SOL_SOCKET, SO_REUSEADDR, &enable, sizeof(int));

if(rc < 0) {

    perror("opt");

    exit(-1);
}

int rc = bind(listenfd, (struct sockaddr *)&serveraddr, sizeof(serveraddr));
if(rc < 0) {

    perror("bind");

    exit(-1);
}

rc = listen(listenfd, 10);

if(rc < 0) {

    perror("listen");

    exit(-1);
}

struct sockaddr_in clientaddr;

socklen_t clen = sizeof(clientaddr);

int clientfd = accept(listenfd, (struct sockaddr *)&clientaddr, &clen);

// int clientfd2 = accept(listenfd, NULL, NULL);

while(1) {

    recv(clientfd, buff, sizeof(buff), 0);

    send(clientf, buff, 10, 0);

}

// CLIENT

int sockfd = socket(AF_INET, SOCK_STREAM, 0);

if(sockfd < 0) {

    perror("socket");

    exit(-1);
}

```

```

struct sockaddr_in serveraddr;
serveraddr.sin_family = AF_INET;
serveraddr.sin_port = htons((unsigned short)portno);
// serveraddr.sin_addr.s_addr = inet_addr("ana are mere");

rc = inet_aton("192.168.0.1", &serveraddr.sin_addr);

if(rc < 0) {
    perror("inet_aton");
    exit(-1);
}

rc = connect(sockfd, (struct sockaddr *)&serveraddr, sizeof(serveraddr));
if(rc < 0) {

...

}

while(1) {

rc = recv(sockfd, buff, sizeof(buff), 0);
if(rc == 0) {

    // s-a inchis conexiunea
    break;

} else if (rc < 0) {

    perror("recv");
    exit(-1);
}

close(sockfd);

// BASH Server
nc -l PORT // nc -lu 5000

// BASH Client
nc 127.0.0.1 5000

strace ./program

```