

Introdução às Redes de Comunicação

Formulário – Teste sockets UDP

CRIAÇÃO, ASSOCIAÇÃO A UM PORTO LOCAL E FECHO DE SOCKETS WINDOWS

```
SOCKET socket(int af, int type, int protocol); /* PF_INET, SOCK_DGRAM, IPPROTO_UDP */  
  
int bind(SOCKET s, const struct sockaddr *name, int namelen);  
  
int closesocket(SOCKET s);
```

INDICAÇÃO DE ERRO E CÓDIGOS DE ERRO

```
SOCKET_ERROR  
  
INVALID_SOCKET  
  
int WSAGetLastError(void); /* WSAETIMEDOUT */
```

LOCALIZAÇÃO E CONVERSÃO DE FORMATOS

```
struct sockaddr_in a; /* a.sin_family, a.sin_addr.s_addr, a.sin_port */  
  
...htons(...); /* host to network short */  
  
...htonl(...); /* host to network long */  
  
...ntohs(...); /* network to host short */  
  
...ntohl(...); /* network to host long */  
  
unsigned long inet_addr(const char *cp);  
  
char* inet_ntoa(struct in_addr in); /* network to ascii */
```

ENVIO E RECEPÇÃO DE DATAGRAMAS

```
int sendto(SOCKET s, const char *buf, int len, int flags, struct sockaddr *to, int tolen);  
  
int recvfrom(SOCKET s, char *buf, int len, int flags, struct sockaddr *from, int *fromlen);
```

OBTENÇÃO DE INFORMAÇÃO LOCAL ASSOCIADA AOS SOCKETS

```
int getsockname(SOCKET s, struct sockaddr *name, int *namelen);
```

```
int strcmp(const char *s1, const char *s2);
```

```
char * strcpy_s(char * strDestination, int sizeStrDestination, const char * strSource);
```

CONFIGURAÇÃO DE OPÇÕES/PARÂMETROS

```
int setsockopt(SOCKET s, int level, int optname, const char *optval, int optlen);
```

```
/* level = SOL_SOCKET, optname = SO_RCVTIMEO, optval = (char *)&timeoutMsec (DWORD **  
timeoutMsec;) */
```