X_SOCKET x_socket(int socktype)

Return a new socket of type *socktype* which is SOCK_STREAM or SOCK_DGRAM.

int x_close(X_SOCKET skt)

Close the socket *skt*.

int x_connect(X_SOCKET skt, const struct sockaddr *addr, int len)

For a SOCK_DGRAM socket *skt* this call specifies the peer address, defined by *addr* and *len*, to be used in sending and receiving datagrams. For a SOCK_STREAM socket *skt* this call attempts a connection operation with the peer defined by *addr* and *len.* Returns 0 on success and -1 on error.

int x_bind(X_SOCKET skt, const struct sockaddr *addr, int len)

Associate the socket *skt* with the specified *addr*. Returns 0 on success and -1 on error.

int x_listen(X_SOCKET skt, int backlog)

Configures socket *skt* to begin listening for incoming connection requests using *backlog* as the size of the queue for pending requests. Returns 0 on success and -1 on error.

X_SOCKET x_accept(X_SOCKET skt, struct sockaddr *addr, int *len)

For a socket skt in the listening state, this call removes the first pending connection request from the queue and returns a socket for use with the connection.

x_getsockopt(X_SOCKET skt,)

int x_setsockopt(X_SOCKET skt, int optname, void *optval, int optlen)

For socket *skt*, use data described by *optval* and *optlen* to set *optname*. Options are

QTTL	int	msec after which data will be retransmitted

X_STATS x_sockstats(X_SOCKET sock)

Return current performance data for socket *sock*. For a SOCK_STREAM this will include the max rate and the current rate. For SOCK_DGRAM this will include max rate, current rate and buffer occupancy.

int **x_sockerror(X_SOCKET skt)**

x_select()

int **x_sendfile(X_SOCKET skt, int fd, int offset, int size)**

Mmap the file described by *fd* and send *size* bytes of data starting at *offset*.

x_send(X_SOCKET skt, const void * buf, int len)

Send *len* bytes from *buf* on socket *skt*. If *skt* is of type SOCK_DGRAM then the data is treated as a single message.

x_recvfile()

x_recv()