X_SOCKET contains

- **stats fields**
- **error status**

X_SOCKET x_socket(int socktype)

Return a new socket of type *socktype* which is SOCK_STREAM or SOCK_DGRAM. This call will always return a socket structure, however if the socket creation fails the socket structure will record an error status which can be obtained through x_sockstatus().

int x_close(X_SOCKET skt)

Close the socket *skt*. Returns 0 on success and -1 on error. This call returns -1 immediately without changing the error status of *skt* if *skt* is marked with an error on entry.

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. This call returns -1 immediately without changing the error status of *skt* if *skt* is marked with an error on entry.

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. This call returns -1 immediately without changing the error status of *skt* if *skt* is marked with an error on entry.

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. This call returns -1 immediately without changing the error status of *skt* if *skt* is marked with an error on entry.

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. If, on entry, *skt* is marked with an error, this call immediately returns NULL without changing the error status.

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

Returns, in *optval* and *optlen*, the data describing *optname* on socket *skt*. On entry *optlen* is the size of space pointed to by *optval*. On exit, *optlen* is changed to reflect the actual size of the returned data. This call returns 0 on success and -1 on error. If, on entry, *skt* is marked with an error, this call immediately returns -1 without changing the error status. The possible option names are

SOCK_STREAM

SOCK_DGRAM

MTU	int	current MTU

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

Uses the data described by *optval* and *optlen* to set *optname* on socket *skt*. Returns 0 on success and -1 on error. If, on entry, *skt* is marked with an error, this call immediately returns -1 without changing the error status. The possible option names are

SOCK_STREAM

SOCK_DGRAM

QTTL	int	The duration (in msec) that unacknowledged packets will remain available for retransmission.

X_STATS **x_sockstats(X_SOCKET skt)**

Return current performance data for socket *skt*. 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. If, on entry, *skt* is marked with an error, this call immediately returns NULL;

int **x_sockstatus(X_SOCKET skt)**

Returns the status of socket *skt*. This will either be an error code resulting from a prior action on the socket or an indication of the availability of data.

char * **x_errortext(int err)**

Returns a text description of the error code *err*.

int **x_select()**

Return the number of sockets in the given set that have the given status. Returns 0 if the time out is exceeded. Returns -1 on error.

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

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

size_t **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()