NAME

estub – EMTS transputer stub

SYNOPSIS

estub [*options*]

DESCRIPTION

estub is a program that copies data between "target" transputer systems and remote servers. The stub can handle multiple target transputer systems within the same stub host, each target with its own host interface. A transputer "target" can be a single transputer development system such as an Inmos B004 or it can be a network of transputers of arbitrary complexity. Each transputer target normally requires its own copy of a server and is accessed via its host interface; the exception to this is if multiple targets are interconnected via transputer links and accessed via a single host interface. A server can only access a single target transputer host interface. If no options are specified, **estub** displays brief usage instructions and configuration information. It shows the maximum number of targets that can be handled (a limit set at compile time).

OPTIONS

–si Information mode. By default, the stub generates no information or warning messages. Supplying the **–si** option puts the stub in **verbose** mode causing it to print a small number of information messages and warning messages (if any) to the standard error stream.

–sd Debug mode. By default, the stub generates no debugging messages. Supplying the **–sd** option puts the stub in **debug** mode causing it to print copious debugging messages to the standard error stream. This option should not normally be required.

–ss Causes the stub to enter its serving loop in which it listens for incoming TCP connection set-up requests from servers, listens for incoming UDP status request packets (see **ewho(1)** for more information), listens for **SP** request packets from any of its target transputer systems and relays these requests to the appropriate server, and listens for **SP** reply packets from any of its servers and relays these replies to the appropriate target. Note that this option is implicit in the **–sb** option (see below).

–se[*n***]** Causes the stub to monitor the transputer error flag on target *0* (if *n* is not specified), on target *n* (if *n* is specified and is a single digit), or on all targets (if *n* is specified and is the * character). This feature is normally selected using an equivalent **eserver(1)** or iserver command-line option rather than using the **–se** option on the stub.

–sr[*n***]** Causes the stub to reset target *0* (if *n* is not specified), target *n* (if *n* is specified and is a single digit), or all targets (if *n* is specified and is the * character). This feature is normally selected using an equivalent **eserver(1)** or iserver command-line option rather than using the **–sr** option on the stub. Note that this option is implicit in the **–sb** option (see below).

–st[*n***]** *linkname*

Causes the stub to open the named link and to map the *linkname* to the target identifier *0* (if *n* is not specified), or the target identifier *n* (if *n* is specified and is a valid number). Under MS-DOS, *linkname* is the hexadecimal i/o address of the target system which may optionally be preceded by a # character. Under Unix, *linkname* is the device file name of the form */dev/b004[xy]* which maps onto the i/o address specified at kernel link time (see *EMTS Installation Guide*). If the stub is required to handle multiple target systems, there will need to be multiple repetitions of this option, each specifying a mapping between a linkname and a target identifier (see EXAMPLES below). This option does not need to be specified if appropriate environment variables are set instead (see ENVIRONMENT below) or if there is a single target transputer system accessible via the default linkname (*#150* under MS-DOS and */dev/b004* under Unix).

–sc[*n***]** *filename*

Causes the stub to copy the specified *filename* to target *0* (if *n* is not specified), to target *n* (if *n* is specified and is a single digit), or to all targets (if *n* is specified and is the * character). This

feature is normally selected using an equivalent **eserver(1)** or iserver command-line option rather than using the **–sc** option on the stub. Note that this option is implicit in the **–sb** option (see below).

–sb[*n***]** *filename*

Causes the stub to reset the target transputer system(s), to boot target *0* (if *n* is not specified), target *n* (if *n* is specified and is a single digit), or all targets (if *n* is specified and is the ∗ character), and to start handling requests from the target transputer system(s). This is equivalent to specifying the **–sr –ss –sc[***n***]** *filename* options. This option may be useful if it is required to boot a target with a bootable file from the stub host's filing system instead of from the server host's filing system. Normally, however, targets are booted from the server using an equivalent **eserver(1)** or iserver command-line option rather than using the **–sb** option on the stub.

ENVIRONMENT

EMTS_TARGET*n*

The *linkname* to map to the target identifier *n*. The value of EMTS_TARGETn is overriden by the value of the command-line option **–st***n*

EMTS_INFO*n*

A description of the target transputer system *n* which will be returned to users of the **ewho(1)** program. Example descriptions might include "Inmos B004, T800, 4Mb" or "Wacky experimental network".

EXAMPLES

example% *estub*

Display brief configuration and help information.

example% *estub -si -ss*

Instruct the stub to enter **verbose** mode, to open the default link, to map (implicitly) the link to the single target identifier *0*, and to enter its server loop.

example% *estub -st0 #200 -st1 #250 -st3 #300 -sb∗ foo.btl -sb0 bar.btl*

Instruct the MS-DOS stub to open the specified links, to map i/o address *#200* to target identifier *0*, to map i/o address *#250* to target identifier *1*, to map i/o address *#300* to target identifier *2*, to boot target *0* with the file *bar.btl*, to boot targets *1* and *2* with the file *foo.btl*, to reset all targets (implicit in the **–sb** options), and to enter into its serving loop (implicit in the **–sb** options).

example% *estub -st0 /dev/b00400 -st1 /dev/b00401 -st3 /dev/b00402 -si -ss*

Instruct the Unix stub to open the specified links, to map device file */dev/b00400* to target identifier *0*, to map device file */dev/b00401* to target identifier *1*, to map device file */dev/b00402* to target identifier *2*, to enter **verbose** mode, and to enter its server loop.

CONFIGURATION

estub looks up the service name *emts* in an attempt to determine which port number to use when listening for incoming TCP connection requests from servers; if the service lookup fails, the default port number **1073** is used.

estub looks up the service name *ewho* in an attempt to determine which port number to use when listening for incoming UDP status request packets from the **ewho(1)** program; if the service lookup fails, the default port number **1072** is used.

See **services(5)** for more information on selecting port numbers.

DIAGNOSTICS

If the stub detects a fatal error *before entering its serving loop*, it terminates returning a non-zero exit code to the operating system. It also prints an abort message to the standard error stream in the following format:

estub: Abort *procid*:*abortid* - *abortmessage*

procid is a number identifying the program procedure from which the abort has originated. *abortid* is a unique number identifying the reason for the abort. *abortmessage* is a brief message indicating the reason for the abort.

If the stub detects a fatal error *after entering its serving loop*, it sends an error packet to the server (causing the server to terminate with an appropriate error message) and closes the connection. It also prints an error message to the standard error stream in the following format:

estub: Error *procid*:*errorid* - *errormessage*

procid is a number identifying the program procedure from which the error has originated. *errorid* is a unique number identifying the reason for the error. *errormessage* is a brief message indicating the reason for the error. The stub does not terminate since there may be other valid stub-server sessions in progress. This behaviour ensures that an error on one stub-server session does not affect any other stub-server session.

IMPLEMENTATION

All the EMTS programs use the BSD socket libraries and the TCP/TP protocol suite for handling communications over the network. They should be portable to a wide variety of platforms that support these libraries.

The current version of the stub runs on MS-DOS (using either FTP's PC/TCP or Sun Microsystem's PC-NFS networking environments) as well as Unix System V Release 4. The stub may be ported to Linux in a later release. The stub supports arbitrary link protocols, with special optimisations for the Inmos SP (iserver) protocol.

IDENTIFICATION

Author: Brendan Murphy.
Revision Number: 2.1; Release Date: 1993/11/01.
Copyright © Brendan Murphy 1991-1993. All Rights Reserved.

SEE ALSO

eserver(1), ewho(1)
EMTS User Guide, *EMTS Installation Guide.*