NAME

term – server for multiple communication channels on a serial link

SYNOPSIS

term [-s speed] [-n on|off] [-c on|off] [-r] [-f arg] [-w arg] [-t arg] [-o] [-a] [-d lev] [-l file] [-v file] [-1]

DESCRIPTION

Term and its clients support multiple, compressed, and error-corrected communication channels over a regular serial link, typically a modem connection. The clients can set up channels to run an interactive shell on the other system similarly *rlogin*(1), to execute commands on the other system similarly *rsh*(1), to transfer files in either direction, to redirect X-server connections from one system to the other, or to redirect any network port connection from one system to a port on the other. Any of these can occur simultaneously, and in either direction.

To support these multiple clients, a term daemon on each system takes control of the device connected to the modem. So you must compile *term* and the clients on both systems. Source is available by anonymous ftp from tartarus.uwa.edu.au:/pub/oreillym/term/term???.tar.z and nic.funet.fi:/pub/OS/Linux/BETA/term???.tar.z

DEFINITION

Due to the fact that anything *term* and its clients can initiate from one end of the link they will just as happily initiate from the other, the following frame of reference will be used throughout this documentation. The remote system refers to the system at the other end of the modem link, and the local host refers to the machine you are typing on, typically the user's home system.

USAGE

Term is run as a daemon which accepts connections from its clients and establishes channels for each client's data. *Term* must be run on both ends. On the remote system, execute something like

term -r -n off

Then, on the local system, either exit your comm program and type

term -v /dev/tty?? [-n off] &

where tty?? is the device name for your modem or serial port. You may be able to start the local term from within your comm program, e.g. for xcomm type

control-A x "$ term"

Once the term daemons are running at both ends they handle the transmission of data over the link. These clients connect to a local *term* daemon which establishes a communication channel with the remote daemon and/or remote processes. The clients are discussed in detail in *term_clients*(1). Briefly, *trsh* runs an interactive shell or commands on the remote system, *tupload* transfers files from the local system to the remote, *tredir* redirects connections from a port on the local system to a port on the remote system, *txconn* redirects X-server connections from the local system to the remote, and *tmon* monitors client statistics.

OPTIONS

Term accepts a number of command-line options which override settings established by the termrc file (see below):

–c off
Turns off compression. Still does error correction. Consider this option if you have hardware compression (ie. v.32.bis) or are transferring compressed files. The default is to have compression on.

–n on|off
Turns on line noise echoing. Talk requests, writes, and biffs to the login which is running term will then appear. This is a useful option to use on the local end (the end you are sitting at). Without this, you will never see any talk requests directed to your remote login. Default is on.

–f <number>
: Makes term send XON at specified interval. Zero is no flow control emulation, 10 is a single XON every 10 characters. The default is zero. 100 is a reasonable value.

–s <number>
: Specifies a maximum number of bits per second (baudrate) term will try to send over the serial link. Term will avoid sending characters at a higher data rate than this. This overrides the BAUDRATE environment variable. The default is 2400. This option is needed as most systems will buffer the data sent to the serial port. Unfortunately this (unknown) buffering can interfere with term's packet timeout mechanism. During setup and tuning it is better to have <number> be too small rather than too large. For high speed links (> 9600), making it unlimited is probably advantageous. This is achieved by setting the number to zero. Term will then rely solely on the packet windows to do flow control.

–w <number>
: Sets the transmission window size, that is, the number of unacknowledged packets that term may send. Lower numbers may lead to better interactive response times. Higher numbers may lead to better throughput, particularly if the serial link latencies are large and the timeout length is set large. Higher numbers can also improve efficiency on noisy lines. Default is 3. This will be too low for higher speed links. A reasonable value is 10 with a timeout (see below) of 150 for 14.4Kbps link.

–t <number>
: Timeout length in 20ths of a second. This specifies how long term will wait for an acknowledgement before retransmitting a packet. Default is 50, maximum 200. Increasing both the window size and timeout can lead to improved throughput, but at the expense of greater latencies for interactive work and during noise recovery.

–r Use this on the remote term so that client numbers won't clash.

–a Turns on seven bit line mode. Use only if you have a seven bit line, as determined by *linecheck.*

–l <filename>
: Output all log/noise/debugging to the file <filename>

–v <filename>
: Set the modem device to be <filename>. Appropriate usage is something like "-v/dev/ttys1"

–1 Use stdout instead of stdin as the modem port. This is ignored if a –v is also present.

–d <number>
: This sets the debuging level. This is useful for monitoring packet timeout conditions and other things. Level 64 is suggested, 478 is verbose. Read debug.h and the other source files to find out what the levels do. If you are not familiar with packet protocols and pouring through source code, then this option is unlikely to help you.

–o Turns on packet send forcing. Re-transmits oldest packet if nothing to send. Default off. Of dubious utility. Not properly debugged. Very bad idea if speed is unlimited.

INITIALIZATION FILE

The file *termrc* in the *.term* directory in your home directory may be used to specify default settings for a user. These settings are overruled by values set in the command line, but several options can only be set in this file. Blank lines and lines beginning with '#' are comments, other lines must begin with one of the options, and, if an argument is required, it must be separated from the option by exactly one space. Legal options are:

escape <number>
: This is one of the most important options for term. Usage is either 'escape <some decimal number>' or 'escape <start of range>-<end of range>'. This tells term to never transmit the

character or characters in the range. This is essential for serial lines that aren't fully transparent. For example, lines that use software flow control will want to do 'escape 17' and 'escape 19'. If you only have a seven-bit link, DO NOT use 'escape 128-255', see the termrc 'sevenbit' option below. See also the linecheck program, and *term_setup*(1).

ignore <number>
This tells term to silently strip this character if it is received over the serial link. Its use should correspond to 'escape' options used by the term on the other system. E.g. if you use 'escape 126' on one end, you should use 'ignore 126' on the other.

compress [on|off].
Turns the default compression mode on or off. The default is 'compress on'. With this on, all data will be compressed by term before sending over the serial line, and then un-compressed at the other end. If you are sending already compressed data, it is recommended that you turn compression off. Similarly, if your modem already does compression you may consider leaving it off. You can turn compression on and off on a client-by-client basis with the '-c' option for clients. See *term_clients*(1).

baudrate <number>.
This is used to limit the rate at which term sends data. Set this to the minimum of your modem speed, and the computer-to-modem baud rate. Values that are too high shouldn't hurt too much, as long as *timeout* is not set too low. Default is 2400.

timeout <number>.
Set the number of 1/20th of a second to wait before re-sending packets that haven't been acknowledged. Low values will provide faster recovery from line noise, but higher values are required if the latencies in your link are large. Values between 50 and 120 are typical. The default value is 70.

window <number>
The size of the packet window. Default value is 3. Increase this if your latencies (and timeout) are large.

noise on
If this is set, then term will send anything it doesn't understand to the standard error or the –l file. This is where talk requests, mail biffs, writes, and corrupted packets will end up. This is recommended for the local end.

remote Sets this to the remote side. It should always be specified on one, and only one end of the link. This prevents client number clashes.

sevenbit
Use this if your line is a sevenbit line. Use this instead of 'escape 128-255'.

breakout <number>
The value of breakout character. Default is 48 ('0'). If either 'term' daemon receives five breakout characters in a row from the link, outside a packet, it will exit.

chdir <path>
Sets the home directory for the term daemon. This will be the directory for processes started started by this daemon for remote *trsh* clients. And it will be used by a remote *tupload* when relative paths are used.

denyrsh on
If this is set, any *trsh* request from the other end will be rejected. This is to provide better security, especially when run as root. You can set this on both ends separately, allowing access on one end and denying on the other.

chroot <directory>
Runs term in a *chroot* environment (see *chroot*(2), *ftpd*(8)). If term is run as root, a *tupload*

from the other side could clobber every file on the system. To avoid this, a *chroot* environment can be set up to give access only to specific directories. It accordingly restricts the commands that can be used by *trsh.* The *chroot* takes place after every file access in the initialisation process, in particular after the server socket is opened.

ENVIRONMENT VARIABLES

BAUDRATE
Used to set the speed. Overridden by termrc or command line setting.

SHELL
Default shell for *trsh.*

DISPLAY
Used by *txconn* to determine which X-server port to use.

TERMDIR
Where to make the .term directory (which contains the socket for *term* itself). Default is HOME.

DIAGNOSTICS

The **–d** option provides debugging output. See *term_setup*(1), *linecheck*(1), or *linerem*(1) for further diagnostic information.

SEE ALSO

term_clients(1)
The man page for the term clients.

term_setup(1)
The man page for the linecheck program and the test program for debugging your serial link and exercising term and clients.

term/README
The original documents for *term* from which these man pages were written.

term/OPTIONS
Another list of options term accepts.

term/TERMRC
Another description of the syntax and options the user may set in ˜/.term/termrc.

term/CHANGES
A list of changes to the program since these manuals were edited. (version 1.0.6a)

BUGS

If a remote client stops consuming its input but leaves the channel open, the local term daemon will continue to resend unacked packets. *Term* by default requires a line which passes through all ascii values from 0 to 255. Xyplex-type terminal servers may require the use of set session passall to work correctly.

AUTHOR

Michael O'Reilly, oreillym@tartarus.uwa.edu.au.