NAME

syncloop – synchronous serial loopback test program

SYNOPSIS

/usr/sbin/syncloop [**–cdlstv**] *device*

AVAILABILITY

SUNWcsu

DESCRIPTION

The **syncloop** command performs several loopback tests that are useful in exercising the various components of a serial communications link.

Before running a test, **syncloop** opens the designated port and configures it according to command line options and the specified test type. It announces the names of the devices being used to control the hardware channel, the channel number (ppa) corresponding to the *device* argument, and the parameters it has set for that channel. It then runs the loopback test in three phases.

The first phase is to listen on the port for any activity. If no activity is seen for at least four seconds, **syncloop** proceeds to the next phase. Otherwise, the user is informed that the line is active and that the test cannot proceed, and the program exits.

In the second phase, called the "first-packet" phase, **syncloop** attempts to send and receive one packet. The program will wait for up to four seconds for the returned packet. If no packets are seen after five attempts, the test fails with an excoriating message. If a packet is returned, the result is compared with the original. If the length and content do not match exactly, the test fails.

The final phase, known as the "multiple-packet" phase, attempts to send many packets through the loop. Because the program has verified the integrity of the link in the first-packet phase, the test will not fail after a particular number of timeouts. If a packet is not seen after four seconds, a message is displayed. Otherwise, a count of the number of packets received is updated on the display once per second. If it becomes obvious that the test is not receiving packets during this phase, the user may wish to stop the program manually. The number and size of the packets sent during this phase is determined by default values, or by command line options. Each returned packet is compared with its original for length and content. If a mismatch is detected, the test fails. The test completes when the required number of packets have been sent, regardless of errors.

After the multiple-packet phase has completed, the program displays a summary of the hardware event statistics for the channel that was tested. The display takes the following form:

CRC errors	**Aborts**	**Overruns**	**Underruns**	**In <-Drops-> Out**	
0	**0**	**0**	**0**	**0**	**0**

This is followed by an estimated line speed, which is an approximation of the bit rate of the line, based on the number of bytes sent and the actual time that it took to send them.

OPTIONS

The options for **syncloop** are described in the following table:

Option	Parameter	Default	Description
−c	*packet_count*	100	Specifies the number of packets to be sent in the multiple-packet phase.
−d	*hex_data_byte*	*random*	Specifies that each packet will be filled with bytes with the value of *hex_data_byte*.
−l	*packet_length*	100	Specifies the length of each packet in bytes.
−s	*line_speed*	9600	Bit rate in bits per second.
−v			Sets verbose mode. If data errors occur, the expected and received data is displayed.
−t	*test_type*	*none*	A number, from 1 to 4, that specifies which test to perform. The values for *test_type* are as follows:

1 Internal loopback test. Port loopback is on. Transmit and receive clock sources are internal (baud rate generator).

2 External loopback test. Port loopback is off. Transmit and receive clock sources are internal. Requires a loopback plug suitable to the port under test.

3 External loopback test. Port loopback is off. Transmit and receive clock sources are external (modem). Requires that one of the local modem, the remote modem, or the remote system be set in a loopback configuration.

4 Test using predefined parameters. User defines hardware configuration and may select port parameters using the **syncinit**(1M) command.

All numeric options except **−d** are entered as decimal numbers (for example, **−s 19200**). If you do not provide the **−t** *test_type* option, **syncloop** prompts for it.

EXAMPLES

In the following command **syncloop** uses a packet length of 512 bytes over the first CPU port:

example# syncloop −l 512 zsh0

In response to the above command, **syncloop** prompts you for the test option you want.

The following command performs an internal loopback test on the first CPU port, using 5000 packets and a bit rate of 56Kbps:

example# syncloop −t 1 −s 56000 −c 5000 zsh0

SEE ALSO

syncinit(1M), **syncstat**(1M), **zsh**(7D)

DIAGNOSTICS

device **missing minor device number**
The name *device* does not end in a decimal number that can be used as a minor device number.

invalid packet length: *nnn*
The packet length was specified to be less than zero or greater than 4096.

poll: nothing to read

poll: nothing to read or write.
The **poll**(2) system call indicates that there is no input pending and/or that output would be blocked if attempted.

len *xxx* **should be** *yyy*
The packet that was sent had a length of *yyy*, but was received with a length of *xxx*.

nnn **packets lost in outbound queueing**

nnn **packets lost in inbound queueing**
A discrepancy has been found between the number of packets sent by **syncloop** and the number of packets the driver counted as transmitted, or between the number counted as received and the number read by the program.

WARNINGS

To allow its tests to run properly, as well as prevent disturbance of normal operations, **syncloop** should only be run on a port that is not being used for any other purpose at that time.