

**NAME**

**srl** — serial line speed emulation

**SYNOPSIS**

**srl** [*file* . . .] [*Baud rate*]

**srl** -h | --help

**srl** -v | --version

**DESCRIPTION****Overview**

The **srl** utility reads files sequentially and writes them to the standard output (*stdout*(4)) with a desired output speed, given in the symbol rate *Baud* (*Bd*). Like *cat*(1), but with an extra speed operand for replay at the end of the command line. The replay speed must be a positive *integer number* and - if given - must be the last operand in the command line. The *file* and *Baud* operands are processed in command line order (see “EXAMPLES” below).

**Details**

If no *file* operands are given, standard input (*stdin*(4)) is read.

If just the *Baud* operand is given, *stdin*(4) is read and being replayed with the desired speed.

One *Baud* is considered as being the “symbol rate” of one symbol per second. For **srl** one symbol is equal to one bit. Thus one *Baud* is considered here to be equal to one bit per second. Or in other terms: One *Baud* is equal to 1/8th of a byte per second; for 300 *Bd* being 37.5 bytes or characters (*chars*) per second.

**Options**

The options – besides operands – are as follows:

- h     Print out a short help to *stdout*(4). No other option is viable with -h.  
In case of an error, this short help will be printed out to *stderr*(4) with a brief description of the nature of the error.
- v     Print out information about version, date, author, copyright and contact data to *stdout*(4). No other option is viable with -v.

**EXIT STATUS**

The **srl** utility exits 0 on success, and >0 if an error occurs.

**EXAMPLES**

The following examples read either from *stdin*(4) or from a *file* until the program receives an EOF (‘D’) character. Thereafter it replays the contents of this *data stream*.

1. Reads from *stdin*(4) and prints out:

```
srl
```

2. Reads from *stdin*(4) and prints out with 300 *Bd*:

```
srl 300
```

3. Reads *file.txt* and prints out:

```
srl file.txt
```

4. Reads *file.txt* and prints out with 300 *Bd*:

```
srl file.txt 300
```

The command:

```
srl file1 > file2
```

will sequentially print the contents of *file1* to *file2*, truncating *file2* if it already exists. See the manual page for your shell (e.g., *sh*(1)) for more information on redirection.

**SEE ALSO**

*cat*(1), *head*(1), *hexdump*(1), *lpr*(1), *more*(1), *pr*(1), *tac*(1), *tail*(1), *view*(1), *vis*(1), *fcntl*(2), *stdin*(4), *stdout*(4), *stderr*(4)

## STANDARDS

The **srl** utility is expected to be written in valid ANSI X3.159-1989 (“ANSI C89”).

## HISTORY

A **srl** utility exists since December 24, 2023. It appears to have been *srl*(1) shell scripts, written initially in *ksh*(1), and then being backported to *sh*(1). This version of the program has been reimplemented and extended in C, beginning on May 12, 2024.

## COPYRIGHT

Copyright (C) 2023-25, *Marc Fege*.

This program and this documentation are provided by the terms of the “GNU GPL” in version 2. The original text of the licence could be obtained from:

<http://www.gnu.org/licenses/old-licenses/gpl-2.0>

## AUTHOR

*Marc Fege* designed and implemented this program and wrote this man page.

Ressources	Address
Internet	<a href="https://www.fege.net/software">https://www.fege.net/software</a>
Mailbox	None.

Contact	Connection
E-mail	<i>13mdf@fege.net</i>
QRA (CB)	13MDF
QRA (HAM)	DN9MF

## BUGS

Because of the shell language mechanism used to perform output redirection, the command “**srl** file1 > file1” will cause the original data in file1 to be destroyed! This is performed by the shell before **srl** is run.