



Red Hat Enterprise Linux Release 9.2 Manual Pages on 'csplit.1p' command

\$ man csplit.1p

CSPLIT(1P) POSIX Programmer's Manual CSPLIT(1P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

csplit ? split files based on context

SYNOPSIS

csplit [-ks] [-f prefix] [-n number] file arg...

DESCRIPTION

The csplit utility shall read the file named by the file operand, write all or part of that file into other files as directed by the arg operands, and write the sizes of the files.

OPTIONS

The csplit utility shall conform to the Base Definitions volume of POSIX.1?2017, Section 12.2, Utility Syntax Guidelines.

The following options shall be supported:

-f prefix Name the created files prefix00, prefix01, ..., prefixn. The default is xx00 ... xxn. If the prefix argument would create a filename exceeding {NAME_MAX} bytes, an error shall result, csplit shall exit with a diagnostic message, and no files shall be created.

- k Leave previously created files intact. By default, csplit shall remove created files if an error occurs.
- n number Use number decimal digits to form filenames for the file pieces. The default shall be 2.
- s Suppress the output of file size messages.

OPERANDS

The following operands shall be supported:

file The pathname of a text file to be split. If file is '-', the standard input shall be used.

Each arg operand can be one of the following:

/rexp/[offset]

A file shall be created using the content of the lines from the current line up to, but not including, the line that results from the evaluation of the regular expression with offset, if any, applied. The regular expression rexp shall follow the rules for basic regular expressions described in the Base Definitions volume of POSIX.1?2017, Section 9.3, Basic Regular Expressions. The application shall use the sequence "\/" to specify a <slash> character within the rexp. The optional offset shall be a positive or negative integer value representing a number of lines. A positive integer value can be preceded by '+'. If the selection of lines from an offset expression of this type would create a file with zero lines, or one with greater than the number of lines left in the input file, the results are unspecified. After the section is created, the current line shall be set to the line that results from the evaluation of the regular expression with any offset applied. If the current line is the first line in the file and a regular expression operation has not yet been performed, the pattern match of rexp shall be applied from the current line to the end of the file. Otherwise, the pattern match of rexp shall be applied from the line following the current line to the end of the file.

`%exp%[offset]`

Equivalent to `/rexp/[offset]`, except that no file shall be created for the selected section of the input file. The application shall use the sequence `"\%"` to specify a `<percent-sign>` character within the `rexp`.

`line_no` Create a file from the current line up to (but not including) the line number `line_no`. Lines in the file shall be numbered starting at one. The current line becomes `line_no`.

`{num}` Repeat operand. This operand can follow any of the operands described previously. If it follows a `rexp` type operand, that operand shall be applied `num` more times. If it follows a `line_no` operand, the file shall be split every `line_no` lines, `num` times, from that point.

An error shall be reported if an operand does not reference a line between the current position and the end of the file.

STDIN

See the INPUT FILES section.

INPUT FILES

The input file shall be a text file.

ENVIRONMENT VARIABLES

The following environment variables shall affect the execution of `csplit`:

`LANG` Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of POSIX.1?2017, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

`LC_ALL` If set to a non-empty string value, override the values of all the other internationalization variables.

`LC_COLLATE`

Determine the locale for the behavior of ranges, equivalence classes, and multi-character collating elements within regular expressions.

LC_CTYPE Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files) and the behavior of character classes within regular expressions.

LC_MESSAGES

Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.

NLSPATH Determine the location of message catalogs for the processing of **LC_MESSAGES**.

ASYNCHRONOUS EVENTS

If the **-k** option is specified, created files shall be retained. Otherwise, the default action occurs.

STDOUT

Unless the **-s** option is used, the standard output shall consist of one line per file created, with a format as follows:

"%d\n", <file size in bytes>

STDERR

The standard error shall be used only for diagnostic messages.

OUTPUT FILES

The output files shall contain portions of the original input file; otherwise, unchanged.

EXTENDED DESCRIPTION

None.

EXIT STATUS

The following exit values shall be returned:

- 0 Successful completion.
- >0 An error occurred.

CONSEQUENCES OF ERRORS

By default, created files shall be removed if an error occurs. When the **-k** option is specified, created files shall not be removed if an error occurs.

The following sections are informative.

APPLICATION USAGE

None.

EXAMPLES

1. This example creates four files, `cobol00 ... cobol03`:

```
csplit -f cobol file '/procedure division/' /par5./ /par16./
```

After editing the split files, they can be recombined as follows:

```
cat cobol0[0-3] > file
```

Note that this example overwrites the original file.

2. This example would split the file after the first 99 lines, and ev?

ery 100 lines thereafter, up to 9999 lines; this is because lines

in the file are numbered from 1 rather than zero, for historical

reasons:

```
csplit -k file 100 {99}
```

3. Assuming that `prog.c` follows the C-language coding convention of

ending routines with a `}` at the beginning of the line, this exam?

ple creates a file containing each separate C routine (up to 21) in

`prog.c`:

```
csplit -k prog.c '%main(%' '^)/+1' {20}
```

RATIONALE

The `-n` option was added to extend the range of filenames that could be handled.

Consideration was given to adding a `-a` flag to use the alphabetic file?

name generation used by the historical `split` utility, but the function?

ality added by the `-n` option was deemed to make alphabetic naming un?

necessary.

FUTURE DIRECTIONS

None.

SEE ALSO

`sed`, `split`

The Base Definitions volume of POSIX.1?2017, Chapter 8, Environment

Variables, Section 9.3, Basic Regular Expressions, Section 12.2, Util?

ity Syntax Guidelines

COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.opengroup.org/unix/online.html> .

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see https://www.kernel.org/doc/man-pages/reporting_bugs.html .

IEEE/The Open Group

2017

CSPLIT(1P)