

NAME

split - split a file into pieces

SYNOPSIS

split [-cd] [-l *line_count*] [-a *suffix_length*] [*file* [*prefix*]]

split [-cd] -b *byte_count*[**K**|**k**|**M**|**m**|**G**|**g**] [-a *suffix_length*] [*file* [*prefix*]]

split [-cd] -n *chunk_count* [-a *suffix_length*] [*file* [*prefix*]]

split [-cd] -p *pattern* [-a *suffix_length*] [*file* [*prefix*]]

DESCRIPTION

The **split** utility reads the given *file* and breaks it up into files of 1000 lines each (if no options are specified), leaving the *file* unchanged. If *file* is a single dash ('-') or absent, **split** reads from the standard input.

The options are as follows:

-a *suffix_length*

Use *suffix_length* letters to form the suffix of the file name.

-b *byte_count*[**K**|**k**|**M**|**m**|**G**|**g**]

Create split files *byte_count* bytes in length. If **k** or **K** is appended to the number, the file is split into *byte_count* kilobyte pieces. If **m** or **M** is appended to the number, the file is split into *byte_count* megabyte pieces. If **g** or **G** is appended to the number, the file is split into *byte_count* gigabyte pieces.

-c Continue creating files and do not overwrite existing output files.

-d Use a numeric suffix instead of a alphabetic suffix.

-l *line_count*

Create split files *line_count* lines in length.

-n *chunk_count*

Split file into *chunk_count* smaller files. The first *n* - 1 files will be of size (size of *file* / *chunk_count*) and the last file will contain the remaining bytes.

-p *pattern*

The file is split whenever an input line matches *pattern*, which is interpreted as an extended regular expression. The matching line will be the first line of the next output file. This option is incompatible with the **-b** and **-l** options.

If additional arguments are specified, the first is used as the name of the input file which is to be split. If a second additional argument is specified, it is used as a prefix for the names of the files into which the file is split. In this case, each file into which the file is split is named by the prefix followed by a lexically ordered suffix using *suffix_length* characters in the range "a-z". If **-a** is not specified, two letters are used as the initial suffix. If the output does not fit into the resulting number of files and the **-d** flag is not specified, then the suffix length is automatically extended as needed such that all output files continue to sort in lexical order.

If the *prefix* argument is not specified, the file is split into lexically ordered files named with the prefix "x" and with suffixes as above.

By default, **split** will overwrite any existing output files. If the **-c** flag is specified, **split** will instead create files with names that do not already exist.

ENVIRONMENT

The LANG, LC_ALL, LC_CTYPE and LC_COLLATE environment variables affect the execution of **split** as described in environ(7).

EXIT STATUS

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

EXAMPLES

Split input into as many files as needed, so that each file contains at most 2 lines:

```
$ echo -e "first line\nsecond line\nthird line\nforth line" | split -l2
```

Split input in chunks of 10 bytes using numeric prefixes for file names. This generates two files of 10 bytes (x00 and x01) and a third file (x02) with the remaining 2 bytes:

```
$ echo -e "This is 22 bytes long" | split -d -b10
```

Split input generating 6 files:

```
$ echo -e "This is 22 bytes long" | split -n 6
```

Split input creating a new file every time a line matches the regular expression for a "t" followed by either "a" or "u" thus creating two files:

```
$ echo -e "stack\nstock\nstuck\nanother line" | split -p 't[au]'
```

SEE ALSO

csplit(1), re_format(7)

STANDARDS

The **split** utility conforms to IEEE Std 1003.1-2001 ("POSIX.1").

HISTORY

A **split** command appeared in Version 3 AT&T UNIX.

Before FreeBSD 14, pattern and line matching only operated on lines shorter than 65,536 bytes.