NAME

csplit – split files based on context

SYNOPSIS

csplit [**–ks**] [**–f** *prefix*] [**–n** *number*] *file arg1 . . .argn*

AVAILABILITY

SUNWesu

DESCRIPTION

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

OPTIONS

The following options are supported:

–**f** *prefix* Name the created files *prefix***00**, *prefix***01**, . . . , *prefixn*. The default is **xx00** . . .**xx***n*. If the *prefix* argument would create a file name exceeding **{NAME_MAX}** bytes, an error will result; **csplit** will exit with a diagnostic message and no files will be created.

–**k** Leave previously created files intact. By default, **csplit** will remove created files if an error occurs.

–**n** *number* Use *number* decimal digits to form filenames for the file pieces. The default is **2**.

–**s** Suppress the output of file size messages.

OPERANDS

The following operands are supported:

file The path name of a text file to be split. If *file* is **-**, the standard input will be used.

The operands *arg1 . . .argn* can be a combination of the following:

/*rexp*/[*offset*] Create a file 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* must follow the rules for basic regular expressions. The optional *offset* must be a positive or negative integer value representing a number of lines. The integer value must be preceded by **+** or –. 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 will be set to the line that results from the evaluation of the regular expression with any offset applied. The pattern match of *rexp* always is applied from the current line to the end of the file.

%*rexp*%[*offset*] This operand is the same as /*rexp*/[*offset*], except that no file will be created for the selected section of the input file.

line_no Create a file from the current line up to (but not including) the line number *line_no*. Lines in the file will 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 will be applied *num* more times. If it follows a *line_no* operand, the file will be split every *line_no* lines, *num* times, from that point.

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

EXAMPLES

This example creates four files, **cobol00** . . . **cobol03**.

example% csplit –f cobol filename '/procedure division/' /par5./ /par16./

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

example% cat cobol0[0–3] > filename

Note: This example overwrites the original file.

This example splits the file at every 100 lines, up to 10,000 lines. The **–k** option causes the created files to be retained if there are less than 10,000 lines; however, an error message would still be printed.

example% csplit –k filename 100 {99}

If **prog.c** follows the normal C coding convention (the last line of a routine consists only of a **}** in the first character position), this example creates a file for each separate C routine (up to 21) in **prog.c**.

example% csplit –k prog.c '%main(%´ '/^}/+1' {20}

ENVIRONMENT

See **environ**(5) for descriptions of the following environment variables that affect the execution of **csplit**: **LC_COLLATE**, **LC_CTYPE**, **LC_MESSAGES**, and **NLSPATH**.

EXIT STATUS

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

SEE ALSO

sed(1), **split**(1), **environ**(5)

DIAGNOSTICS

The diagnostic messages are self-explanatory, except for the following:

arg – **out of range** The given argument did not reference a line between the current position and the end of the file.