Full credit is given to the above companies including the OS that this PDF file was generated!

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

$ man cut.1p

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

cut ? cut out selected fields of each line of a file

SYNOPSIS

cut -b list [-n] [file...]

cut -c list [file...]

cut -f list [-d delim] [-s] [file...]

DESCRIPTION

The cut utility shall cut out bytes (-b option), characters (-c op? tion), or character-delimited fields (-f option) from each line in one or more files, concatenate them, and write them to standard output.

OPTIONS

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

The application shall ensure that the option-argument list (see options -b, -c, and -f below) is a <comma>-separated list or <blank>-separated list of positive numbers and ranges. Ranges can be in three forms. The first is two positive numbers separated by a <hyphen-minus> (low-high),

which represents all fields from the first number to the second number. The second is a positive number preceded by a <hyphen-minus> (-high), which represents all fields from field number 1 to that number. The third is a positive number followed by a <hyphen-minus> (low-), which represents that number to the last field, inclusive. The elements in list can be repeated, can overlap, and can be specified in any order, but the bytes, characters, or fields selected shall be written in the order of the input data. If an element appears in the selection list more than once, it shall be written exactly once.

The following options shall be supported:

-b list Cut based on a list of bytes. Each selected byte shall be output unless the -n option is also specified. It shall not be an error to select bytes not present in the input line.

-c list Cut based on a list of characters. Each selected character shall be output. It shall not be an error to select charac? ters not present in the input line.

-d delim Set the field delimiter to the character delim. The default is the <tab>.

-f list Cut based on a list of fields, assumed to be separated in the file by a delimiter character (see -d). Each selected field shall be output. Output fields shall be separated by a single occurrence of the field delimiter character. Lines with no field delimiters shall be passed through intact, unless -s is specified. It shall not be an error to select fields not present in the input line.

-n Do not split characters. When specified with the -b option, each element in list of the form low-high (<hyphen-mi? nus>-separated numbers) shall be modified as follows:

* If the byte selected by low is not the first byte of a character, low shall be decremented to select the first byte of the character originally selected by low. If the byte selected by high is not the last byte of a charac? ter, high shall be decremented to select the last byte of

the character prior to the character originally selected by high, or zero if there is no prior character. If the resulting range element has high equal to zero or low greater than high, the list element shall be dropped from list for that input line without causing an error.

Each element in list of the form low- shall be treated as above with high set to the number of bytes in the current line, not including the terminating <newline>. Each element in list of the form -high shall be treated as above with low set to 1. Each element in list of the form num (a single num? ber) shall be treated as above with low set to num and high set to num.

-s Suppress lines with no delimiter characters, when used with the -f option. Unless specified, lines with no delimiters shall be passed through untouched.

OPERANDS

The following operand shall be supported:

file A pathname of an input file. If no file operands are speci? fied, or if a file operand is '-', the standard input shall be used.

STDIN

The standard input shall be used only if no file operands are speci? fied, or if a file operand is '-'. See the INPUT FILES section.

INPUT FILES

The input files shall be text files, except that line lengths shall be unlimited.

ENVIRONMENT VARIABLES

The following environment variables shall affect the execution of cut:

LANG Provide a default value for the internationalization vari? ables that are unset or null. (See the Base Definitions vol? ume of POSIX.1?2017, Section 8.2, Internationalization Vari? ables 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_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).

LC_MESSAGES

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

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

ASYNCHRONOUS EVENTS

Default.

STDOUT

The cut utility output shall be a concatenation of the selected bytes, characters, or fields (one of the following):

"%s\n", <concatenation of bytes>

"%s\n", <concatenation of characters>

"%s\n", <concatenation of fields and field delimiters>

STDERR

The standard error shall be used only for diagnostic messages.

OUTPUT FILES

None.

EXTENDED DESCRIPTION

None.

EXIT STATUS

The following exit values shall be returned:

0 All input files were output successfully.

>0 An error occurred.

CONSEQUENCES OF ERRORS

Default.

The following sections are informative.

APPLICATION USAGE

The cut and fold utilities can be used to create text files out of files with arbitrary line lengths. The cut utility should be used when the number of lines (or records) needs to remain constant. The fold utility should be used when the contents of long lines need to be kept contiguous.

Earlier versions of the cut utility worked in an environment where bytes and characters were considered equivalent (modulo <backspace> and <tab> processing in some implementations). In the extended world of multi-byte characters, the new -b option has been added. The -n option (used with -b) allows it to be used to act on bytes rounded to charac? ter boundaries. The algorithm specified for -n guarantees that:

```
cut -b 1-500 -n file > file1
cut -b 501- -n file > file2
```

ends up with all the characters in file appearing exactly once in file1 or file2. (There is, however, a <newline> in both file1 and file2 for each <newline> in file.)

EXAMPLES

Examples of the option qualifier list:

1,4,7 Select the first, fourth, and seventh bytes, characters, or fields and field delimiters.

1-3,8 Equivalent to 1,2,3,8.

-5,10 Equivalent to 1,2,3,4,5,10.

3- Equivalent to third to last, inclusive.

The low-high forms are not always equivalent when used with -b and -n and multi-byte characters; see the description of -n.

The following command:

```
cut -d : -f 1,6 /etc/passwd
```

reads the System V password file (user database) and produces lines of the form:

```
<user ID>:<home directory>
```

Most utilities in this volume of POSIX.1?2017 work on text files. The cut utility can be used to turn files with arbitrary line lengths into

a set of text files containing the same data. The paste utility can be used to create (or recreate) files with arbitrary line lengths. For example, if file contains long lines:

```
cut -b 1-500 -n file > file1
cut -b 501- -n file > file2
```

creates file1 (a text file) with lines no longer than 500 bytes (plus the <newline>) and file2 that contains the remainder of the data from file. (Note that file2 is not a text file if there are lines in file that are longer than 500 + {LINE_MAX} bytes.) The original file can be recreated from file1 and file2 using the command:

```
paste -d "\0" file1 file2 > file
```

RATIONALE

Some historical implementations do not count <backspace> characters in determining character counts with the -c option. This may be useful for using cut for processing nroff output. It was deliberately decided not to have the -c option treat either <backspace> or <tab> characters in any special fashion. The fold utility does treat these characters specially.

Unlike other utilities, some historical implementations of cut exit after not finding an input file, rather than continuing to process the remaining file operands. This behavior is prohibited by this volume of POSIX.1?2017, where only the exit status is affected by this problem.

The behavior of cut when provided with either mutually-exclusive options or options that do not work logically together has been deliberately left unspecified in favor of global wording in Section 1.4, Utility Description Defaults.

The OPTIONS section was changed in response to IEEE PASC Interpretation 1003.2 #149. The change represents historical practice on all known systems. The original standard was ambiguous on the nature of the output.

The list option-arguments are historically used to select the portions of the line to be written, but do not affect the order of the data. For example:

echo abcdefghi | cut -c6,2,4-7,1

yields "abdefg".

A proposal to enhance cut with the following option:

-o Preserve the selected field order. When this option is specified, each byte, character, or field (or ranges of such) shall be written in the order specified by the list option-argument, even if this requires multiple outputs of the same bytes, characters, or fields.

was rejected because this type of enhancement is outside the scope of the IEEE P1003.2b draft standard.

FUTURE DIRECTIONS

None.

SEE ALSO

Section 2.5, Parameters and Variables, fold, grep, paste

The Base Definitions volume of POSIX.1?2017, Chapter 8, Environment Variables, Section 12.2, Utility Syntax Guidelines

COPYRIGHT