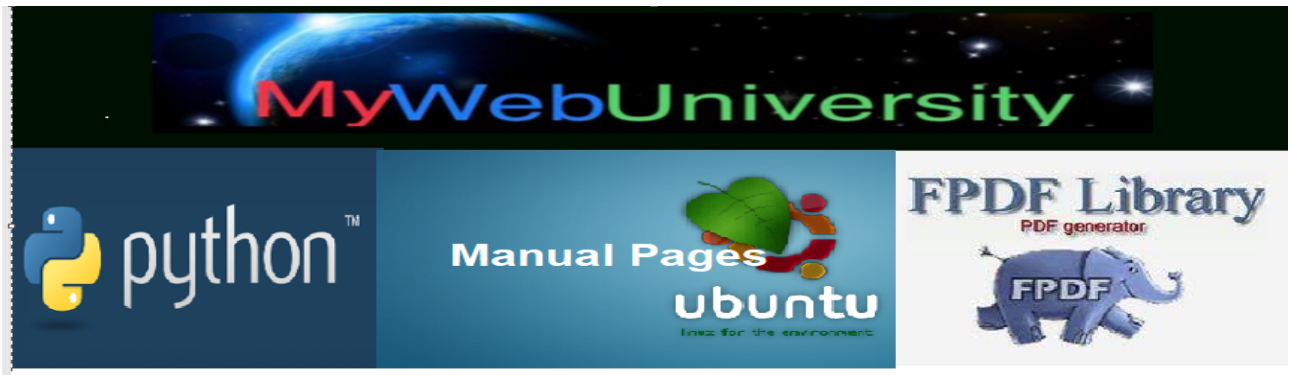

Full credit is given to the above companies including the OS that this PDF file was generated!

Linux Ubuntu 26.04 Manual Pages on command 'cut.1'

$ man cut.1

CUT(1) General Commands Manual CUT(1)

NAME

cut - Prints specified byte or field columns from each line of stdin or the input files

SYNOPSIS

cut [-b|--bytes] [-c|--characters] [-d|--delimiter] [-w] [-f|--fields] [--complement]

[-s|--only-delimited] [-z|--zero-terminated] [--output-delimiter] [-n] [-h|--help]

[-V|--version] [file]

DESCRIPTION

Prints specified byte or field columns from each line of stdin or the input files

OPTIONS

-b, --bytes <LIST>

filter byte columns from the input source

-c, --characters <LIST>

alias for character mode

-d, --delimiter <DELIM>

specify the delimiter character that separates fields in the input source. Defaults

to Tab.

-w Use any number of whitespace (Space, Tab) to separate fields in the input source

(FreeBSD extension).

-f, --fields <LIST>

filter field columns from the input source

--complement

invert the filter - instead of displaying only the filtered columns, display all but

those columns

-s, --only-delimited

in field mode, only print lines which contain the delimiter

-z, --zero-terminated

instead of filtering columns based on line, filter columns based on \0 (NULL charac?

ter)

--output-delimiter <NEW_DELIM>

in field mode, replace the delimiter in output lines with this option's argument

-n (ignored)

-h, --help

Print help

-V, --version

Print version

EXTRA

Each call must specify a mode (what to use for columns), a sequence (which columns to print), and provide a data source

Specifying a mode

Use --bytes (-b) or --characters (-c) to specify byte mode

Use --fields (-f) to specify field mode, where each line is broken into fields identified by a delimiter character. For example for a typical CSV you could use this in combination with setting comma as the delimiter

Specifying a sequence

A sequence is a group of 1 or more numbers or inclusive ranges separated by a commas.

cut -f 2,5-7 some_file.txt

will display the 2nd, 5th, 6th, and 7th field for each source line

Ranges can extend to the end of the row by excluding the second number

cut -f 3- some_file.txt

will display the 3rd field and all fields after for each source line

The first number of a range can be excluded, and this is effectively the same as using 1 as the first number: it causes the range to begin at the first column. Ranges can also display a single column

cut -f 1,3-5 some_file.txt

will display the 1st, 3rd, 4th, and 5th field for each source line

The --complement option, when used, inverts the effect of the sequence

cut --complement -f 4-6 some_file.txt

will display the every field but the 4th, 5th, and 6th

Specifying a data source

If no sourcefile arguments are specified, stdin is used as the source of lines to print

If sourcefile arguments are specified, stdin is ignored and all files are read in consecu?

tively if a sourcefile is not successfully read, a warning will print to stderr, and the

eventual status code will be 1, but cut will continue to read through proceeding sourcefiles

To print columns from both STDIN and a file argument, use - (dash) as a sourcefile argument

to represent stdin.

Field Mode options

The fields in each line are identified by a delimiter (separator)

Set the delimiter

Set the delimiter which separates fields in the file using the --delimiter (-d) option. Set?

ting the delimiter is optional. If not set, a default delimiter of Tab will be used.

If the -w option is provided, fields will be separated by any number of whitespace charac?

ters (Space and Tab). The output delimiter will be a Tab unless explicitly specified. Only

one of -d or -w option can be specified. This is an extension adopted from FreeBSD.

Optionally Filter based on delimiter

If the --only-delimited (-s) flag is provided, only lines which contain the delimiter will

be printed

Replace the delimiter

If the --output-delimiter option is provided, the argument used for it will replace the de?

limiter character in each line printed. This is useful for transforming tabular data - e.g.

to convert a CSV to a TSV (tab-separated file)

Line endings

When the --zero-terminated (-z) option is used, cut sees \0 (null) as the 'line ending'

character (both for the purposes of reading lines and separating printed lines) instead of

\\n (newline). This is useful for tabular data where some of the cells may contain newlines

echo 'ab\0cd' | cut -z -c 1

will result in 'a\0c\0'

VERSION

v(uutils coreutils) 0.8.0