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 'du.1p' command

$ man du.1p

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

du ? estimate file space usage

SYNOPSIS

du [-a|-s] [-kx] [-H|-L] [file...]

DESCRIPTION

By default, the du utility shall write to standard output the size of the file space allocated to, and the size of the file space allocated to each subdirectory of, the file hierarchy rooted in each of the spec? ified files. By default, when a symbolic link is encountered on the command line or in the file hierarchy, du shall count the size of the symbolic link (rather than the file referenced by the link), and shall not follow the link to another portion of the file hierarchy. The size of the file space allocated to a file of type directory shall be de? fined as the sum total of space allocated to all files in the file hi? erarchy rooted in the directory plus the space allocated to the direc? tory itself.

When du cannot stat() files or stat() or read directories, it shall re?

port an error condition and the final exit status is affected. A file that occurs multiple times under one file operand and that has a link count greater than 1 shall be counted and written for only one entry. It is implementation-defined whether a file that has a link count no greater than 1 is counted and written just once, or is counted and written for each occurrence. It is implementation-defined whether a file that occurs under one file operand is counted for other file operands. The directory entry that is selected in the report is unspecified. By default, file sizes shall be written in 512-byte units, rounded up to the next 512-byte unit.

OPTIONS

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

The following options shall be supported:

-a In addition to the default output, report the size of each file not of type directory in the file hierarchy rooted in the specified file. The -a option shall not affect whether non-directories given as file operands are listed.

-H If a symbolic link is specified on the command line, du shall count the size of the file or file hierarchy referenced by the link.

-k Write the files sizes in units of 1024 bytes, rather than the default 512-byte units.

-L If a symbolic link is specified on the command line or encountered during the traversal of a file hierarchy, du shall count the size of the file or file hierarchy referenced by the link.

-s Instead of the default output, report only the total sum for each of the specified files.

-x When evaluating file sizes, evaluate only those files that have the same device as the file specified by the file operand.

Specifying more than one of the mutually-exclusive options -H and -L

shall not be considered an error. The last option specified shall de?
termine the behavior of the utility.

OPERANDS

The following operand shall be supported:

file The pathname of a file whose size is to be written. If no file is specified, the current directory shall be used.

STDIN

Not used.

INPUT FILES

None.

ENVIRONMENT VARIABLES

The following environment variables shall affect the execution of du:

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).

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 output from du shall consist of the amount of space allocated to a file and the name of the file, in the following format:

"%d %s\n", <size>, <pathname>

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 Successful completion.

>0 An error occurred.

CONSEQUENCES OF ERRORS

Default.

The following sections are informative.

APPLICATION USAGE

None.

EXAMPLES

None.

RATIONALE

The use of 512-byte units is historical practice and maintains compati? bility with ls and other utilities in this volume of POSIX.1?2017. This does not mandate that the file system itself be based on 512-byte blocks. The -k option was added as a compromise measure. It was agreed by the standard developers that 512 bytes was the best default unit be? cause of its complete historical consistency on System V (versus the mixed 512/1024-byte usage on BSD systems), and that a -k option to switch to 1024-byte units was a good compromise. Users who prefer the 1024-byte quantity can easily alias du to du -k without breaking the many historical scripts relying on the 512-byte units.

The -b option was added to an early proposal to provide a resolution to the situation where System V and BSD systems give figures for file sizes in blocks, which is an implementation-defined concept. (In common usage, the block size is 512 bytes for System V and 1024 bytes for BSD

systems.) However, -b was later deleted, since the default was eventu? ally decided as 512-byte units.

Historical file systems provided no way to obtain exact figures for the space allocation given to files. There are two known areas of inaccura? cies in historical file systems: cases of indirect blocks being used by the file system or sparse files yielding incorrectly high values. An indirect block is space used by the file system in the storage of the file, but that need not be counted in the space allocated to the file.

A sparse file is one in which an lseek() call has been made to a posi? tion beyond the end of the file and data has subsequently been written at that point. A file system need not allocate all the intervening zero-filled blocks to such a file. It is up to the implementation to define exactly how accurate its methods are.

The -a and -s options were mutually-exclusive in the original version of du. The POSIX Shell and Utilities description is implied by the language in the SVID where -s is described as causing ``only the grand total'' to be reported. Some systems may produce output for -sa, but a Strictly Conforming POSIX Shell and Utilities Application cannot use that combination.

The -a and -s options were adopted from the SVID except that the System V behavior of not listing non-directories explicitly given as operands, unless the -a option is specified, was considered a bug; the BSD-based behavior (report for all operands) is mandated. The default behavior of du in the SVID with regard to reporting the failure to read files (it produces no messages) was considered counter-intuitive, and thus it was specified that the POSIX Shell and Utilities default behavior shall be to produce such messages. These messages can be turned off with shell redirection to achieve the System V behavior.

The -x option is historical practice on recent BSD systems. It has been adopted by this volume of POSIX.1?2017 because there was no other his? torical method of limiting the du search to a single file hierarchy.

This limitation of the search is necessary to make it possible to ob? tain file space usage information about a file system on which other

file systems are mounted, without having to resort to a lengthy find and awk script.

FUTURE DIRECTIONS

A future version of this standard may require that a file that occurs multiple times shall be counted and written for only one entry, even if the occurrences are under different file operands.

SEE ALSO

ls

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

The System Interfaces volume of POSIX.1?2017, fstatat()

COPYRIGHT