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 'df.1p' command

$ man df.1p

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

df ? report free disk space

SYNOPSIS

df [-k] [-P|-t] [file...]

DESCRIPTION

The df utility shall write the amount of available space and file slots for file systems on which the invoking user has appropriate read ac? cess. File systems shall be specified by the file operands; when none are specified, information shall be written for all file systems. The format of the default output from df is unspecified, but all space fig? ures are reported in 512-byte units, unless the -k option is specified. This output shall contain at least the file system names, amount of available space on each of these file systems, and, if no options other than -t are specified, the number of free file slots, or inodes, avail? able; when -t is specified, the output shall contain the total allo? cated space as well.

OPTIONS

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

The following options shall be supported:

-k Use 1024-byte units, instead of the default 512-byte units, when writing space figures.

-P Produce output in the format described in the STDOUT section.

-t Include total allocated-space figures in the output.

OPERANDS

The following operand shall be supported:

file A pathname of a file within the hierarchy of the desired file system. If a file other than a FIFO, a regular file, a di? rectory, or a special file representing the device containing the file system (for example, /dev/dsk/0s1) is specified, the results are unspecified. If the file operand names a file other than a special file containing a file system, df shall write the amount of free space in the file system containing the specified file operand. Otherwise, df shall write the amount of free space in that file system.

STDIN

Not used.

INPUT FILES

None.

ENVIRONMENT VARIABLES

The following environment variables shall affect the execution of df:

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 error and informative messages written to standard output.

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

ASYNCHRONOUS EVENTS

Default.

STDOUT

When both the -k and -P options are specified, the following header line shall be written (in the POSIX locale):

"Filesystem 1024-blocks Used Available Capacity Mounted on\n"

When the -P option is specified without the -k option, the following header line shall be written (in the POSIX locale):

"Filesystem 512-blocks Used Available Capacity Mounted on\n"

The implementation may adjust the spacing of the header line and the individual data lines so that the information is presented in orderly columns.

The remaining output with -P shall consist of one line of information for each specified file system. These lines shall be formatted as fol? lows:

"%s %d %d %d %d%% %s\n", <file system name>, <total space>, <space used>, <space free>, <percentage used>, <file system root>

In the following list, all quantities expressed in 512-byte units (1024-byte when -k is specified) shall be rounded up to the next higher unit. The fields are:

<file system name>

The name of the file system, in an implementation-defined format.

<total space>

The total size of the file system in 512-byte units. The ex?

act meaning of this figure is implementation-defined, but should include <space used>, <space free>, plus any space re? served by the system not normally available to a user.

<space used>

The total amount of space allocated to existing files in the file system, in 512-byte units.

<space free>

The total amount of space available within the file system for the creation of new files by unprivileged users, in 512-byte units. When this figure is less than or equal to zero, it shall not be possible to create any new files on the file system without first deleting others, unless the process has appropriate privileges. The figure written may be less than zero.

<percentage used>

The percentage of the normally available space that is cur? rently allocated to all files on the file system. This shall be calculated using the fraction:

<space used>/(<space used>+ <space free>)

expressed as a percentage. This percentage may be greater than 100 if <space free> is less than zero. The percentage value shall be expressed as a positive integer, with any fractional result causing it to be rounded to the next high? est integer.

<file system root>

The directory below which the file system hierarchy appears.

The output format is unspecified when -t is used.

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

On most systems, the ``name of the file system, in an implementation-defined format'' is the special file on which the file system is mounted.

On large file systems, the calculation specified for percentage used can create huge rounding errors.

EXAMPLES

1. The following example writes portable information about the /usr file system:

```
df -P /usr
```

2. Assuming that /usr/src is part of the /usr file system, the follow?ing produces the same output as the previous example:

```
df -P /usr/src
```

RATIONALE

The behavior of df with the -P option is the default action of the 4.2 BSD df utility. The uppercase -P was selected to avoid collision with a known industry extension using -p.

Historical df implementations vary considerably in their default out?put. It was therefore necessary to describe the default output in a loose manner to accommodate all known historical implementations and to add a portable option (-P) to provide information in a portable format.

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 more logical 1024-byte quantity can easily alias df to df -k without breaking many historical scripts relying on the 512-byte units.

It was suggested that df and the various related utilities be modified to access a BLOCKSIZE environment variable to achieve consistency and user acceptance. Since this is not historical practice on any system, it is left as a possible area for system extensions and will be re-evaluated in a future version if it is widely implemented.

FUTURE DIRECTIONS

None.

SEE ALSO

find

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

COPYRIGHT