NAME

dtt types a “top ten” ranking of object collections found on a file−structured storage device such as a Unix or Windows volume. In other words, it floats the really big chunks to the top.

SYNOPSIS

dtt [*options*] [*files*]

DESCRIPTION

Object measurements are aggregated into collections by directory name, or optionally by filename suffix. A grand total is reported, followed by the top ten collections in order, largest to smallest.

The measurements available for ranking are:

- physical size,
- virtual (apparent) size,
- directory entries, and
- inodes.

When using the directory name aggregation method, dtt defines a directory’s measurement as its own measurement plus the sum of the measurements of its contents. To avoid double−counting, if any directory would be in the top ten list along with one or more of its descendant directories or objects, only the descendants are reported, not the directory.

When using the filename suffix aggregation method, dtt defines a suffix’s measurement as the sum of the measurements of all files having that suffix. Objects having no suffix are aggregated using an empty (blank) suffix.

OPTIONS

Options, given in the argument list, alter the behavior of dtt. Giving an unknown option is a fatal error: dtt types a diagnostic message on stderr and stops.

MEASUREMENT CONTROL OPTIONS

Physical size

When no measurement control options are given, dtt measures physical size. This is useful when the focus of the investigation is on disk blocks in use on the local volume.

Virtual size

The −−apparent−size option measures virtual (apparent) size. This is useful when the focus of the investigation is large objects, regardless of disk space currently in use. Virtual size is often less than physical size, because file lengths do not generally extend exactly to their block boundaries. But it can also be much more than physical size, in the case of sparse or offline files. The −−apparent−size, −−dirent, and −−inodes options conflict with one another.

Directory entries or inodes

The −−dirent (−d) option counts directory entries instead of file sizes. The −−inodes option counts inodes instead of file sizes. A file with multiple hard links will be counted each time by −−dirent, but will be counted once by −−inodes. The −−apparent−size, −−dirent, and −−inodes options conflict with one another.

When −−dirent or −−inodes are used, dtt operates as if −−block−size=1 were given as the first argument. In these modes, environment variables DTT_BLOCK_SIZE, BLOCK_SIZE, and BLOCKSIZE are ignored.

Other values of −−block−size|B can be given, to scale a −−dirent or −−inodes report. For example, −−dirent −−block−size=KB would report the number of directory entries scaled to 1000 files per unit. −−dirent −−block−size=K would scale the report to 1024 files per unit, so that 65536 entries would be reported as 64. (If doing that is confusing, consider not doing that.)

SCALING CONTROL OPTIONS

Scaling to a given unit of measure

The −−block−size=s (−B s) option scales measurements to a given unit of measure. The option value can be n, or it can be an integer having an optional suffix. The −−human−readable and −−si options conflict

with this option.

The option value n means “native blocksize”. It scales measurements to the st_blksize property of the file system being reported on, such as 4096.

Optional suffixes used with an integer correspond to those supported by du and df: K or KiB for customary (binary) kilobytes (1024 bytes), KB for metric kilobytes (1000 bytes), and so on for M (mega), G (giga), T (tera), P (peta), E (exa), Z (zetta), Y (yotta), R (ronna), and Q (quetta). Scaled values are rounded to whole numbers and typed with no suffix.

When −−block−size is not present, dtt operates as if −−block−size were given as the first argument. It takes the option value from the environment. dtt searches for the variables DTT_BLOCK_SIZE, BLOCK_SIZE, and BLOCKSIZE in the environment. Searching in this order, the first value found determines the runtime value. If none of these variables are found, the value 512 is used. (But see −−dirent and −−inodes for a special case change to this behavior.)

The option value n is only compatible with measurements of multiple filesystems when they agree on st_blksize. Multiple blocksizes cause dtt to operate as if −−block−size is not present. If the option value n is set by an environment variable, the value 512 is used.

Scaling to human−readable binary units

The −−human−readable (−h) option scales sizes individually to customary binary units of measure. Any count below 2^(10) (1024) is scaled to bytes, and typed as a counting number. Larger values are scaled to customary kilobytes or “kibibytes” (K) through quettabytes (Q), rounded, and typed as a number with a letter suffix. For example, a count of 2^(10) or more but less than 2^(20) is scaled to customary kilobytes, rounded, and typed with a K suffix. Rounded values below 10 are rounded to 1 decimal place. Rounded values of 10 or more are rounded to whole numbers. The −−block−size and −−si options conflict with this option.

Scaling to human−readable SI (metric) units

The −−si option scales sizes individually to metric units of measure. It operates like −−human−readable, using multiples of 1000 instead of 1024. The −−block−size and −−human−readable options conflict with this option.

AGGREGATION CONTROL OPTIONS

Aggregating by file and directory names

When no aggregation control options are given, dtt aggregates by file and directory names.

Aggregating by filename suffix

The −−suffix (−s) option aggregates by filename suffix. This is useful when the focus of the investigation is particular file types. The suffixes recognized are:

* a final dot in the filename, optionally followed by other characters; or,
* a final tilde.

MISCELLANEOUS OPTIONS

The −−one−file−system (−x) option limits the dtt investigation to one filesystem. This makes dtt behave like find −xdev or du −x. When dtt detects a filesystem change, it types a message on stderr, skips the object, and continues.

The −−help (−h) option types the command synopsis on stdout. No measurements are taken. The −−version option conflicts with this option.

The −−verbose (−v) option types more information on stderr as the count takes place. Currently that information consists of warnings when duplicate inodes are ignored, and warnings when a change in st_blksize is found.

The −−version option types version information on stdout. No measurements are taken. The −−help option conflicts with this option.

OPTION CONFLICT RESOLUTION

When the operator combines options, conflicts are resolved according to the following rules.

* Options are processed from left to right.
* When conflicting options are given, or the same option is given multiple times with different values, the rightmost option processed takes precedence.

OBJECT LIST ARGUMENTS

After options, the operator can supply names of files and directories as arguments. These are the objects to be measured. If the operator defaults (supplies no names), the objects in the current working directory will be measured, including hidden objects, but excluding the . and .. entries.

PHASES OF OPERATION

The program operation has three phases:

1. measure,
2. reduce to a top−ten list, and
3. type the report.

MEASURE PHASE

The Measure Phase measures each object in accord with the selected Measurement Control option. When an object is a directory, its descendants are also measured. Hidden descendants are measured, except the . and .. entries. Measurements are aggregated in accord with the selected Aggregation Control option. Encountering an unreadable or nonexistent object is not a fatal error. dtt types a diagnostic message on stderr and continues with the next object to be measured.

REDUCE PHASE

The Reduce Phase iterates over the accumulated objects in order by decreasing measurement, saving the top ten. As noted above, when using the directory aggregation method, it discards any directory object which would be in the top ten with one of its child objects. The Reduce Phase also adds the “grand total” object that shows the total of all measurements.

REPORT PHASE

The aggregated objects and measurements are typed on stdout, ordered by decreasing size and scaled in accord with the selected Scaling Control option.

AUTHOR

Edward K. McGuire

COPYRIGHT AND LICENSE

Copyright © 2011, 2022, 2023, 2024 Edward K. McGuire, Fort Worth, Texas. All rights reserved. Redistribution and use of this software, with or without modification, is permitted, provided that the following conditions are met:

1. Redistribution of this software must retain the copyright notice above, this list of conditions, and the disclaimer below.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCY OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NOTES

DETAILS OF OPERATION

dtt is careful of multiple entries in the directory structure which catalog the same file (duplicate inodes). Storage allocated to an inode is not counted more than once.

The output will only be accurate if the operator's credentials permit the program to collect the information. The standard error output should be monitored for any errors indicating that the program did not have permission to scan a directory.

The native dtt unit of measure is the 512−byte block. This is for compatibility with Unix dt/du. If the system has Gnu dt/du, the program installer can configure dtt to use 1024−byte blocks using an environment variable.

LANGUAGE PROCESSOR

The language of the program is strict Perl 5. It is syntax−compatible with Perl 5.6.0, but requires 5.6.1 or newer to work properly. Every release of the program is tested against multiple Perl 5 releases on the Perl stable release track:

* the Perl 5 prepackaged with the test environment
* the latest patch levels of all newer Perl 5 stable releases
* the oldest Perl that still works with it

The program is not tested against the Perl 5 experimental release track.

This release of dtt has been tested with:

* Perl 5.40.0 (stable release)
* Perl 5.38.2 (stable release)
* Perl 5.36.3 (stable release)
* Perl 5.34.3 (stable release packaged with dev environment)
* Perl 5.6.1 (oldest release that works)

ACKNOWLEDGEMENTS

dtt was inspired by du in standard Unix. In fact, du is used to test dtt. What du doesn't do, and dtt does well, is aggregation by whole directory subtree or suffix with no redundant reporting.

I am indebted to N. Ramsey for *Noweb −− A Simple, Extensible Tool for Literate Programming*, used to prepare dtt and its documentation for public release. Migrating dtt to *Noweb* has made dtt better software.

T. Oetiker, et al., *The Not So Short Introduction to LaTeX*, was an essential reference.

P. Volkerding, et al., *Slackware 15.0*, provided an ideal test platform for development.

D. Book, *Perl::MinimumVersion*, and K. Liu, *App::perlbrew*, made possible the support of dtt with older Perl releases.

BUGS

Gnu du seems to round up, but dtt rounds to nearest number. This is arguably a bug in Gnu du.

Gnu du does not count apparent size of files other than regular files and symbolic links, beginning with Gnu coreutils release 9.2. dtt has followed suit starting in release 2.3, to make its output verifiable against Gnu du. This is arguably a bug in both Gnu du and dtt.

The −−suffix results should be tested against a real−world filesystem … but it's not clear how. Also, The three measurement types are well−tested, but the many new output scaling options, not so much.

dtt is careful never to read file blocks. On a sparse volume, such as a OneDrive or Google Drive File Stream, reading file blocks causes those blocks to be loaded over the network to the local cache. The cost of such an operation is heavy network traffic and loss of free blocks on the local disk. However, dtt must read directories. This unavoidably causes directory files to be cached, and incurs the network traffic cost to cache them.

A conflict between multiple blocksizes and −−blocksize=n should maybe fall back to a local system default scale factor, instead of always 512.

ROADMAP

dtt should be packaged for more platforms than Slackware 15.0. When packaged for Slackware 15.0, the default unit of measure should be customized to 1024 to match the behavior of Gnu df/du on the target platform; the same for other platforms having Gnu df/du.

How to handle an overlayfs is an interesting question.

Consider implementing −n|−−number or something like it to change top “ten” to top “some other number”. Consider adding −i for −−inodes. Suggested by pan64 at LinuxQuestions.org.

Consider whether dtt is capable of supporting btrfs.