NAME

du — assess disk usage, file size, or count

SYNOPSIS

du [**-P**|**-HD**|**-L**] [**-cxlS0**] [**-a**|**-s**] [**-d** *depth*]
[**-i**|[**-A**] {[**-bkmg**|**-B** *block-size*] [**-,**]}|[**-h**|**--si**]] [**-t** [**-**]*threshold*]
[**--exclude**=*pattern*]... [**-X** *pattern-file*]
[**--time**[={**atime**|**access**|**use**}|{**ctime**|**status**}]
[**--time-style**=**iso**|**long-iso**|**full-iso**|**+***date-format*]]
[*file*]...|**--files0-from**=*files*

DESCRIPTION

Writes the space allocated to, apparent size of, or file count under *file*s (or "**.**", the default) in blocks, and their path, separated by a tab, one per line, to the standard output stream. With **--files0-from**, the contents of *files* (standard input stream if "**-**") are used as a NUL-separated list of filenames instead.

The first output column is the size:

- by default, the space allocated, according to the *st_blocks* *struct stat* field — most likely less than the apparent size due to holes, filesystem compression, &c.;
- with **-A**, apparent/"total" size of the i-node: this is meaningful for regular files and symbolic links, but not for device nodes &a., and corresponds to the *st_size* field;
- with **-i**, **1** for each processed file.

A directory's size is the sum of its and all of its descendants' size, and the max of its and its descendants' time, unless **-S**. If that size exceeds **16E**, it's considered infinite.

By default, all traversed directories are written. **-a** writes files of all types, and **-sd** limit the directory depth for which files are written, but *not* which ones are processed.

If any *pattern* glob(7)-matches a path, or any of its **/**-delimited tails, (as strings (fnmatch(3) mode **0**), not pathnames), that path, including its children, is excluded from processing. Unless **-l**, each file is processed only once.

-c adds another line for a total file, whose size is the sum of all processed files, and time the max of all processed files'. If no files were processed, its time is **never**.

With **-h** or **--si** sizes are output in a human-readable 3.2T-style. Otherwise, sizes are output in rounded-up blocks of **-B**, the first valid of the DU_BLOCK_SIZE, BLOCK_SIZE, BLOCKSIZE environment variables, or **512** bytes. **-i** is always output as **-B** 1 (a simple count), overriding all unit specifiers.

-B, the block size environment variables, and **-t** are in the case-insensitive format:
[*base*][**KMGTPEZY**][**B**] (with at least one of {*base*, **KMGTPEZY**, **B**})
Where *base* is an optionally-floating-point number of bytes, defaulting to **1**, which is then optionally multiplied by the relevant unit. **B** sets the unit multiplier to **1000** (from **1024**). The block size is equal to $base \cdot unit^{mult}$, if any, or *base*.

With **--time**, the time format is the one specified by **--time-style**, the TIME_STYLE environment variable, or **long-iso**. If a time is unrepresentable in the current time-zone, it's written as-if via %s.%N and a diagnostic is issued to the standard error stream.

OPTIONS

-P, **--no-dereference**	Never follow symbolic links. This is the default.
-H, **-D**, **--dereference-args**	Only follow *file*s, but not any of their descendants.
-L, **--dereference**	Follow all symbolic links.
-c, **--total**	Write an additional line with the sum (max) of all processed files.

-x, **--one-file-system** Do not process or enter filesystems (mount-points) different than their corresponding *file*.

-l, **--count-links** Process every file each time it's encountered, instead of only the first time.

-S, **--separate-dirs** Consider each directory to contain only non-directories for size (time) purposes.

-0, **--null** End output lines with a NUL instead of a newline.

-a, **--all** Write all file types, not just directories (top-level *file*s are always written).

-s, **--summari**[**sz**]**e** Write only the top-level *file*s. Excludes **-a** and **-d** >**0**.

-d, **--max-depth**=*depth* Do not write files below *depth*, with the top-level *file* being at level **0**, its children at level **1**, &c. **-d** 0 is equivalent to **-s**, but may be used together wtih **-a**.

-i, **--inodes** Count **1** for every file processed. Overrides **-A**, and all block size specifiers with **-B** 1.

-A, **--apparent-size** Count the apparent size, rather than actual space taken.

-B, **--block-size**=*block-size*
Set block size for size output.

-b, **--bytes** Equivalent to **-AB** 1.

-k Equivalent to **-B** 1k.

-m Equivalent to **-B** 1M.

-g Equivalent to **-B** 1G.

-, Format all numbers with thousands-separators (where the locale places them, so not necessarily every thousand). For example, this may turn a total size of **893458** blocks (around 436 kibibytes by default) into **893,458**.

-h, **--human-readable** Fold all sizes into a human readable *1024*-based 3.2T style. Overrides **-B**. Supersedes **-,**.

--si Likewise, but *1000*. Supersedes **-,**.

-t, **--threshold=** *threshold* Do not write files smaller than *threshold*.

-t, **--threshold=-***threshold* Do not write files bigger than *threshold*.

--exclude=*pattern* Do not process files whose any path tail **/** segment matches *pattern*.

-X, **--exclude-from**=*pattern-file*
Use exclusion patterns from newline-delimited *pattern-file* (standard input stream if "**-**"); if it contains NULs, the patterns for those lines are terminated at these points. A union is taken of the patterns in *pattern-file* and ones specified via **--exclude**.

--time Insert a column containing the modification time (*st_mtim*) after the size.

--time=atime|access|use Likewise, but the access time (*st_atim*).

--time=ctime|status Likewise, but the i-node status change time (*st_ctim*).

--time-style=iso Equivalent to **--time-style=+"**%010F**"** (*YYYY-MM-DD* — the ISO 8601 date format).

--time-style=long-iso Equivalent to **--time-style=+"**%010F %R**"** (*YYYY-MM-DD HH:MM*); this is the default.

--time-style=full-iso Equivalent to **--time-style=+"**%010F %T.%N %z**"** (*YYYY-MM-DD HH:MM:SS.NSNSNSNSN ±TZTZ*).

--time-style=+*date-format* Format via date(1)-compatible *date-format* format, truncated before the first newline.
A "**posix-**" prefix, if present, is removed from **--time-style**.

All **--time** and non-**+ --time-style** values are prefix-matched (**--time**=*c* **--time-style**=*posix-f* is equivalent to **--time**=**ctime** **--time-style**=**full-iso**, &c.).

ENVIRONMENT

DU_BLOCK_SIZE, BLOCK_SIZE, BLOCKSIZE
: The first valid of these variables sets the default block size, instead of **512**.

TIME_STYLE
: Used as the time format if **--time** and no **--time-style**, instead of the default **long-iso**.

TZ
: Override timezone for formatting **--time**s, cf. tzset(3).

EXIT STATUS

1 if *files* or *pattern-file* couldn't be accessed or read, or *file*s or any of their descendants couldn't.

EXAMPLES

Assess the on-disk sizes of user homes, singling out a known delinquent:

```
# du -hcs /home/cicada /home /root
1.9G    /home/cicada
193M    /home
13M     /root
2.1G    total
```

Compare the actual and apparent sizes for (sparse, compressed) images bigger than 20MiB, in MiB units, and note the last modification date of each:

```
$ printf '%s\n' '*.sh' 'initrd*' |
  du -mad1 -X- -t20M --time --time-style=iso
882     2022-05-10      ./sr.ht-alpine
54      2022-06-05      ./43bsd
464     2022-05-05      ./debian-hurd-20210812.img
15484   2021-05-10      ./tzpfmest
219     2021-09-05      ./ultrix
1338    2022-06-28      ./fedora-server
641     2017-12-05      ./debian-unofficial-kfreebsd-amd64-NETINST-1.iso
19130   2022-07-14      .

$ printf '%s\n' '*.sh' 'initrd*' |
  du -mad1 -X- -t20M --time --time-style=iso -A
873     2022-06-30      ./42bsd
3073    2022-05-10      ./sr.ht-alpine
542     2022-06-05      ./43bsd
5001    2022-05-05      ./debian-hurd-20210812.img
16897   2021-05-10      ./tzpfmest
1327    2021-09-05      ./ultrix
40960   2022-06-28      ./fedora-server
61      2021-08-05      ./rt11
648     2017-12-05      ./debian-unofficial-kfreebsd-amd64-NETINST-1.iso
69416   2022-07-14      .
```

SEE ALSO

date(1), df(1), ls(1), lstat(2), stat(2), fnmatch(3), glob(7), inode(7)

STANDARDS

Conforms to IEEE Std 1003.1-2024 ("POSIX.1") — only **-HLxkas** are standard. BLOCKSIZE and **-P** are extensions, originating from 4.4BSD. The **-D** and DU_BLOCK_SIZE, BLOCK_SIZE spellings originate from the GNU system. **-clS0**, **-d**, **--inodes**, **--apparent-size**, **-bmgBh**,

--si, **-t**, **--exclude**, **-X**, **--time**, **--time-style** (TIME_STYLE) are extensions, originating from the GNU system (though **-cdmh** are widely available on modern BSD). The GNU system considers a dangling symbolic link that would be traversed (in **-H** or **-P** mode) an error; this implementation, like 4.4BSD, counts the symbolic link itself. The **-i** spelling is compatible with NetBSD. The **-A** spelling is compatible with the illumos gate and FreeBSD. The **--summarise** spelling is an extension. **-,** is an extension.

The GNU system brokenly defaults to **-k** unless the POSIXLY_CORRECT environment variable is set.

The GNU system disallows **-t***0*, block sizes with **B** but without a multiplier (**-B**[*base*]**B**), as well as lower-case **B** (**-B***base*[*mult*]**b**), and only supports integer *base*s; it also suffers from numerous time handling bugs, esp. given timestamps before the epoch and **-c** with no processed files, and writes unrepresentable times without the sub-second component. The *pattern-file* is read before *files*, compatibly with the GNU system.

HISTORY

Research UNIX

Appears in the first edition of the UNIX Programmer's Manual as du(I):

```
NAME        du  --  summarize disk usage
SYNOPSIS    du [ -s ] [ -a ] [ name ... ]
```

With a present-day output format and **-as** (though, rather than being exclusive, each pair cancels itself out). Lacking a *st_blocks* equivalent, block counts for each file are derived from the *st_size* equivalent (though still **512** and rounded up). The two **BUGS** are that top-level file *name*s aren't written without **-a** (though they're written in both the **-a** or **-s** modes), and that the visited i-node cache is not discriminated by the filesystem — **du** / does not descend into /usr, since the root i-node number is, expectedly, the same for both. Conversely, **du** / /usr walks the entirety of both filesystems, since the seen i-node cache is emptied for each *name*. For non-directories, repeats are written, but with a size of **0**. This is, of course, described succinctly as

> A file which has two links to it is only counted once.

Version 3 AT&T UNIX adds a *st_major*, *st_minor* equivalent, unused in **du**. Since at most Version 5 AT&T UNIX, the i-node cache is grown on the fly rather than containing just **500** (appx. twice the file count in an entire Version 1 AT&T UNIX system) entries.

Version 7 AT&T UNIX sees a rewrite in C: only one of **-as**, as the first argument, is recognised, and I/O errors are noted to the standard error stream; the second **BUGS** entry becomes just

> If there are too many distinct linked files, *du* counts the excess files multiply.

This limit is **1000** {*st_dev*, *st_ino*} pairs, now across the entire program lifetime, but counted only if the file has more than one link and isn't a directory: **du -a** /unix / // would count /unix for all three *name*s, and only single-link files for the third one. As a fresh addition, the trailing /, if any, is trimmed for output, regardless of the path at hand: the above invocation would write lines for /unix, , and /.

The BSD

4.1cBSD:

- with the advent of symbolic links, elects to not follow them,
- coalesces the sizes into blocks of **1024** (with the corresponding manual paragraph updated to note "kilobytes" instead of "blocks"; this is also the only substantive change),
- accepts any amount of **-as**es starting the argument list — naturally, if both are specified, all non-directories are written, regardless of depth, and top-level non-directories are written twice,
- fork(2)s for all but the final *name* (there doesn't seem to be a good reason for this, except to avoid a **getwd**()/**chdir**()?),
- and hence deduplicates files across each *name* separately.

4.2BSD sees the advent of *st_blocks* and uses it instead of scaling *st_size* for each file.

4.3BSD–Reno sees a **SYNOPSIS** of

du [**-aksx**] [*pathname* **. . .**]

in consort with a rewrite:

- top-level *pathname*s are always written, as present-day,
- **-s** forces output, even for repeat *pathname*s,
- **-sa** (in this order) writes just the non-directory files; the block size and **-kx** are as present-day, and
- the seen-file cache is global across the program lifespan, growing as needed, but only tracks files with more than one link.

This means that **du -s** /unix /unix / / writes /unix twice, assesses / the first time, and writes size **0** for it the second time. If a directory can be executed but not read, **du** fails to return to its parent, but continues processing, to the obvious side-effect of being very broken for the rest of the *pathname* (but subsequent *pathname*s work as expected).

4.4BSD sees another rewrite, this time in terms of fts(3), with a **SYNOPSIS** of

du [**-H** | **-L** | **-P**] [**-a** | **-s**] [**-x**] [*file* **. . .**]

and $BLOCKSIZE handling via getbsize(3) — case-insensitive

[*base*][**KMG**]

with an integer *base*, units in powers of **1024**, clamped to [**512**, **1G**] with a warning, and a default of **512**. This is largely as present-day, except file tracking is unchanged, and **-x** still processes (but does not descend down) mounted directories — this matches the FTS_XDEV behaviour directly.

System V

Programmer's Workbench 2.0 (PWB/UNIX 2.0) sees a variation on the Version 7 AT&T UNIX **du** without final / trimming, chdir(2)ing to walk the tree, and a shorter (**100**-entry) seen i-node cache, which, if it were to overflow, instead of happily going off the end, considers all multiply-linked non-directories to have been seen, and all error output commented out.

AT&T System III UNIX sees a **500**-entry cache with a reasonable overflow mechanism, instead, considering them to not have been seen, and a **SYNOPSIS** of

du [**-ars**] [names]

with **-r** enabling error output, which remains unchanged from Version 7 AT&T UNIX. Blocks are counted as previously, with the addition of i-node/metadata block counts: if the size-based block count works out to more than the number of direct blocks (i.e. ones stored "for free" in the i-node), additional blocks are added for the number of indirect blocks (i.e. ones containing just data pointers) needed to store the pointers to the data, and likewise for doubly-indirect ones. This manual is the first to note that

Files with holes in them will get an incorrect block count.

AT&T System V Release 1 UNIX calculates block counts in increments of the filesystem block size (BSIZE, which depends on the filesystem configured when building — **512** for the "original" (and when built with dual-filesystem support) and **1024** for the new one).

AT&T System V Release 2 UNIX exits **1** when the seen i-node cache overflows and always uses a "logical" block size of **512**, but continues to calculate the additional indirect blocks (with capacities now also dependent on the configured filesystem) based off that figure — this means that on the "new" (**1024**-block) filesystem, those are overestimated two-fold.

AT&T System V Release 4 UNIX includes 4.2BSD **du** in /usr/ucb, but exits (and bubbles from subprocesses) **1** for I/O errors and rounds *st_size* up to the nearest kilobyte before processing. Additionally, that includes a 135-word comment waxing poetic about the many downfalls of not using *st_blocks* and various vendors' many solutions (yes, despite 4.2BSD coming with *st_blocks*).

Its own **du** is derived from the /usr/ucb one (but, notably, missing the Regents of the University of California licensing notice), using getopt(3) with **-r** guarding error writes, and writing *st_blocks* directly, with its advent.

Standards

X/Open Portability Guide Issue 2 ("XPG2") specifies

du [−ars] [file ...]

marked OF ("Output format incompletely specified" – it isn't at all) PI ("The behaviour cannot be guaranteed to be consistent"), mostly standardising AT&T System III UNIX behaviour: proclaiming **512**-byte units but noting that some systems don't report in such, laconically declaring that

> A file with two or more links is only counted once.

with no lifetime requirements, warning that

> Files with holes in them may get an incorrect (high) estimate.

and inventing a broken **−a** spec that means nothing for good measure. In a UN ("Possibly unsupportable feature")-marked block, **−r** is described as:

> Some implementations of *du* are silent about directories that cannot be read, files that cannot be opened, etc. If this is the case, the *−r* option will cause *du* to generate messages in such instances.

Which is a classic standards moment.

X/Open Portability Guide Issue 4 ("XPG4") (quoting alignment with the IEEE Std 1003.2a-1992 ("POSIX.2") (User Portability Extension) supplement):

- renders **−as** exclusive,
- adds **−x**, as present-day,
- fixes non-directories at the top level not being listed without **−a**,
- requires normal error reporting and makes **−r** an OB(solete) no-op, as well as
- requiring **1024**-byte output unit with the new **−k** flag, the rounding-up behaviour, and defining the output format, all as present-day.

Version 2 of the Single UNIX Specification ("SUSv2") marks **du** LEGACY.

IEEE Std 1003.1-2001 ("POSIX.1"):

- "reinstates" it (by unmarking it LEGACY, noting that to have been incorrect),
- makes it part of the User Portability Utilities feature group,
- removes **−r**, and
- adds **−HL** (and in general opines on symlink traversal behaviour — imported from IEEE Std 1003.2b ("POSIX.2b": Shell and Utilities — Amendment) draft),

both as present-day.

IEEE Std 1003.1-2008 ("POSIX.1") moves **du** to the base spec, since its User Portability Utilities are exclusively interactive.

IEEE Std 1003.1-2024 ("POSIX.1") tightens the file deduplication requirements to present-day: each file is only ever processed *once*, regardless of its link count.