NAME

pv - monitor the progress of data through a pipe

SYNOPSIS

pv [*OPTION*] [*FILE*]...
pv [-*h*|-*V*]

DESCRIPTION

pv shows the progress of data through a pipeline by giving information such as time elapsed, percentage completed (with progress bar), current throughput rate, total data transferred, and ETA.

To use it, insert it in a pipeline between two processes, with the appropriate options. Its standard input will be passed through to its standard output and progress will be shown on standard error.

pv will copy each supplied *FILE* in turn to standard output (- means standard input), or if no *FILE*s are specified just standard input is copied. This is the same behaviour as **cat**(1).

A simple example to watch how quickly a file is transferred using **nc**(1):

```
pv file | nc -w 1 somewhere.com 3000
```

A similar example, transferring a file from another process and passing the expected size to **pv**:

```
cat file | pv -s 12345 | nc -w 1 somewhere.com 3000
```

A more complicated example using numeric output to feed into the **dialog**(1) program for a full-screen progress display:

```
(tar cf - . \
 | pv -n -s $(du -sb . | awk ’{print $1}’) \
 | gzip -9 > out.tgz) 2>&1 \
| dialog --gauge ’Progress’ 7 70
```

Taking an image of a disk, skipping errors:

```
pv -EE /dev/your/disk/device > disk-image.img
```

Writing an image back to a disk:

```
pv disk-image.img > /dev/your/disk/device
```

Zeroing a disk:

```
pv < /dev/zero > /dev/your/disk/device
```

Note that if the input size cannot be calculated, and the output is a block device, then the size of the block device will be used and **pv** will automatically stop at that size as if **-S** had been given.

(Linux only): Watching file descriptor 3 opened by another process 1234:

```
pv -d 1234:3
```

(Linux only): Watching all file descriptors used by process 1234:

```
pv -d 1234
```

OPTIONS

pv takes many options, which are divided into display switches, output modifiers, and general options.

DISPLAY SWITCHES

If no display switches are specified, **pv** behaves as if **-p**, **-t**, **-e**, **-r**, and **-b** had been given (i.e. everything except average rate is switched on). Otherwise, only those display types that are explicitly switched on will be shown.

-p, --progress

Turn the progress bar on. If any inputs are not files, or are unreadable, and no size was explicitly given (with the **-s** modifier), the progress bar cannot indicate how close to completion the transfer is, so it will just move left and right to indicate that data is moving.

-t, --timer

Turn the timer on. This will display the total elapsed time that **pv** has been running for.

-e, --eta

Turn the ETA timer on. This will attempt to guess, based on current transfer rates and the total data size, how long it will be before completion. This option will have no effect if the total data size cannot be determined.

-I, --fineta
Turn the ETA timer on, but display the estimated local time of arrival instead of time left. When the estimated time is more than 6 hours in the future, the date is shown as well.

-r, --rate
Turn the rate counter on. This will display the current rate of data transfer.

-a, --average-rate
Turn the average rate counter on. This will display the current average rate of data transfer (default: last 30s, see **-m**).

-b, --bytes
Turn the total byte counter on. This will display the total amount of data transferred so far.

-8, --bits
Display the total bits instead of the total bytes. The output suffix will be "b" instead of "B".

-k, --si
Display and interpret suffixes as multiples of 1000 rather than the default of 1024. Note that this only takes effect on options after this one, so for consistency, specify this option first.

-T, --buffer-percent
Turn on the transfer buffer percentage display. This will show the percentage of the transfer buffer in use **-** but see the caveat under **%T** in the **FORMATTING** section below. Implies **-C**.

-A *NUM*, **--last-written** *NUM*
Show the last *NUM* bytes written **-** but see the caveat under **%nA** in the **FORMATTING** section below. Implies **-C**.

-F *FORMAT*, **--format** *FORMAT*
Ignore the options **-p**, **-t**, **-e**, **-r**, **-a**, **-b**, **-T**, and **-A**, and instead use the format string *FORMAT* to determine the output format. See the **FORMATTING** section below.

-n, --numeric
Numeric output. Instead of giving a visual indication of progress, **pv** will give an integer percentage, one per line, on standard error, suitable for piping (via convoluted redirection) into **dialog**(1). Note that **-f** is not required if **-n** is being used.

Note that if **--numeric** is in use, then adding **--bytes** will cause the number of bytes processed so far to be output instead of a percentage; if **--line-mode** is also in use as well as **--bytes** and **--numeric**, then instead of bytes or a percentage, the number of lines so far is output. And finally, if **--timer** is added to **--numeric**, then each output line is prefixed with the elapsed time so far, as a decimal number of seconds.

-q, --quiet

No output. Useful if the **-L** option is being used on its own to just limit the transfer rate of a pipe.

OUTPUT MODIFIERS

-W, --wait

Wait until the first byte has been transferred before showing any progress information or calculating any ETAs. Useful if the program you are piping to or from requires extra information before it starts, eg piping data into **gpg**(1) or **mcrypt**(1) which require a passphrase before data can be processed.

-D *SEC*, **--delay-start** *SEC*

Wait until *SEC* seconds have passed before showing any progress information, for example in a script where you only want to show a progress bar if it starts taking a long time. Note that this can be a decimal such as 0.5.

-s *SIZE*, **--size** *SIZE*

Assume the total amount of data to be transferred is *SIZE* bytes when calculating percentages and ETAs. The same suffixes of "k", "m" etc can be used as with **-L**.

If *SIZE* starts with **@**, the size of file whose name follows the **@** will be used.

Note that **--size** has no effect if used with **-d** *PID* to watch all file descriptors of a process, but will work with **-d PID:FD**.

-l, --line-mode

Instead of counting bytes, count lines (newline characters). The progress bar will only move when a new line is found, and the value passed to the **-s** option will be interpreted as a line count.

If this option is used without **-s**, the "total size" (in this case, total line count) is calculated by reading through all input files once before transfer starts. If any inputs are pipes or non-regular files, or are unreadable, the total size will not be calculated.

-0, --null

Count lines as terminated with a zero byte instead of with a newline. This option implies --line-mode.

-i *SEC*, **--interval** *SEC*

Wait *SEC* seconds between updates. The default is to update every second. Note that this can be a decimal such as 0.1.

-m *SEC*, **--average-rate-window** *SEC*

Compute current average rate over a *SEC* seconds window for average rate and ETA calculations (default 30 seconds).

-w *WIDTH*, **--width** *WIDTH*

Assume the terminal is *WIDTH* characters wide, instead of trying to work it out (or assuming 80 if it cannot be guessed). If this option is used, the output width will not be adjusted if the width of the terminal changes while the transfer is running.

-H *HEIGHT*, **--height** *HEIGHT*

Assume the terminal is *HEIGHT* rows high, instead of trying to work it out (or assuming 25 if it cannot be guessed). If this option is used, the output height will not be adjusted if the height of the terminal changes while the transfer is running.

-N *NAME*, **--name** *NAME*

Prefix the output information with *NAME*. Useful in conjunction with **-c** if you have a complicated pipeline and you want to be able to tell different parts of it apart.

-f, --force

Force output. Normally, **pv** will not output any visual display if standard error is not a terminal. This option forces it to do so.

-c, --cursor

Use cursor positioning escape sequences instead of just using carriage returns. This is useful in conjunction with **-N** (name) if you are using multiple **pv** invocations in a single, long, pipeline.

DATA TRANSFER MODIFIERS

-o *FILE*, **--output** *FILE*

Write data to *FILE* rather than standard output. If the file already exists, it will be truncated.

-L *RATE*, **--rate-limit** *RATE*

Limit the transfer to a maximum of *RATE* bytes per second. A suffix of "K", "M", "G", or "T" can be added to denote kibibytes (*1024), mebibytes, and so on. If **--si** was also passed, suffixes will denote kilobytes (*1000), megabytes, etc. Note the caveat about the positioning of **--si .**

-B *BYTES*, **--buffer-size** *BYTES*

Use a transfer buffer size of *BYTES* bytes. A suffix of "K", "M", "G", or "T" can be added to denote kibibytes (*1024), mebibytes, and so on. The default buffer size is the block size of the input file's filesystem multiplied by 32 (512KiB max), or 400KiB if the block size cannot be determined. This can be useful on platforms like MacOS which perform better in pipelines with specific buffer sizes such as 1024. Implies **-C**.

-C, --no-splice

Never use **splice**(2), even if it would normally be possible. The **splice**(2) system call is a more efficient way of transferring data from or to a pipe than regular **read**(2) and **write**(2), but means that the transfer buffer may not be used. This prevents **-A** and **-T** from working, cannot work with **-X**, and makes **-B** redundant, so using **-A**, **-T**, **-X**, or **-B** automatically switches on **-C**. Switching on **-C** results in a small loss of transfer efficiency. (This option has no effect on systems where **splice**(2) is unavailable).

-E, --skip-errors

Ignore read errors by attempting to skip past the offending sections. The corresponding parts of the output will be null bytes. At first only a few bytes will be skipped, but if there are many errors in a row then the skips will move up to chunks of 512. This is intended to be similar to **dd conv=sync,noerror** but has not been as thoroughly tested.

Specify **-E** twice to only report a read error once per file, instead of reporting each byte range skipped.

-Z *BYTES*, **--error-skip-block** *BYTES*

When ignoring read errors with **-E**, instead of trying to adaptively skip by reading small amounts and skipping progressively larger sections until a read succeeds, move to the next file block of *BYTES* bytes as soon as an error occurs. There may still be some shorter skips where the block being skipped coincides with the end of the transfer buffer.

This option can only be used with **-E** and is intended for use when reading from a block device, such as **-E -Z 4K** to skip in 4 kibibyte blocks. This will speed up reads from faulty media, at the expense of potentially losing more data.

-S, --stop-at-size

If a size was specified with **-s**, stop transferring data once that many bytes have been written, instead of continuing to the end of input.

-Y, --sync

After every write operation, synchronise the buffer caches to disk - see **fdatasync**(2). This has no effect when the output is a pipe. Using **-Y** may improve the accuracy of the progress bar when writing to a slow disk.

-K, --direct-io

Set the **O_DIRECT** flag on all inputs and outputs, if it is available. This will minimise the effect of caches, at the cost of performance. Due to memory alignment requirements, it also may cause read or write failures with an error of "Invalid argument", especially if reading and writing files across a variety of filesystems in a single **pv** call. Use this option with caution.

-X, --discard

Instead of transferring input data to standard output, discard it. This is equivalent to redirecting standard output to */dev/null*, except that **write**(2) is never called. Implies **-C**.

-d *PID*[**:***FD*], **--watchfd** *PID*[**:***FD*]

Instead of transferring data, watch file descriptor *FD* of process *PID*, and show its progress. The **pv** process will exit when *FD* either changes to a different file, changes read/write mode, or is closed; other data transfer modifiers - and remote control - may not be used with this option.

If only a *PID* is specified, then that process will be watched, and all regular files and block devices it opens will be shown with a progress bar. The **pv** process will exit when process *PID* exits.

-R *PID*, **--remote** *PID*

If *PID* is an instance of **pv** that is already running, **-R** *PID* will cause that instance to act as though it had been given this instance's command line instead. For example, if **pv -L 123K** is running with process ID 9876, then running **pv -R 9876 -L 321K** will cause it to start using a rate limit of 321KiB instead of 123KiB. Note that some options cannot be changed while running, such as **-c**, **-l**, **-f**, **-D**, **-E**, and **-S**.

GENERAL OPTIONS

-P *FILE*, **--pidfile** *FILE*

Save the process ID of **pv** in *FILE*. The file will be replaced if it already exists, and will be removed when **pv** exits. While **pv** is running, it will contain a single number - the process ID of **pv** - followed by a newline.

-h, --help

Print a usage message on standard output and exit successfully.

-V, --version

Print version information on standard output and exit successfully.

FORMATTING

If the **-F** option is given, then the output format is determined by the given format string. Within that string, the following sequences can be used:

%p Progress bar. Expands to fill the remaining space. Should only be specified once. Equivalent to **-p**.

%t Elapsed time. Equivalent to **-t**.

%e ETA as time remaining. Equivalent to **-e**.

%I ETA as local time of completion. Equivalent to **-I**.

%r Current data transfer rate. Equivalent to **-r**.

%a Average data transfer rate. Equivalent to **-a**.

%b Bytes transferred so far (or lines if **-l** was specified). Equivalent to **-b**. If **--bits** was specified, **%b** shows the bits transferred so far, not bytes.

%T Percentage of the transfer buffer in use. Equivalent to **-T**. Shows "{----}" if the transfer is being done with **splice**(2), since splicing to or from pipes does not use the buffer.

%nA

Show the last **n** bytes written (e.g. **%16A** for the last 16 bytes). Shows only dots if the transfer is being done with **splice**(2), since splicing to or from pipes does not use the buffer.

%N

Name prefix given by **-N**. Padded to 9 characters with spaces, and suffixed with :.

%%

A single %.

The format string equivalent of turning on all display switches is **‘%N %b %T %t %r %a %p %e’**.

COMMON SWITCHES

Some suggested common switch combinations:

pv -ptebar

Show a progress bar, elapsed time, estimated completion time, byte counter, average rate, and current rate.

pv -betlap

Show a progress bar, elapsed time, estimated completion time, line counter, and average rate, counting lines instead of bytes.

pv -t

Show only the elapsed time - useful as a simple timer, e.g. **sleep 10m | pv -t**.

pv -pterb

The default behaviour: progress bar, elapsed time, estimated completion time, current rate, and byte counter.

On MacOS, it may be useful to specify **-B 1024** in a pipeline, as this may improve performance.

EXIT STATUS

An exit status of 1 indicates a problem with the **-R** or **-P** options.

Any other exit status is a bitmask of the following:

2 One or more files could not be accessed, **stat**(2)ed, or opened.

4 An input file was the same as the output file.

8 Internal error with closing a file or moving to the next file.

16 There was an error while transferring data from one or more input files.

32 A signal was caught that caused an early exit.

64 Memory allocation failed.

A zero exit status indicates no problems.

ENVIRONMENT

The following environment variables may affect **pv**:

HOME

The current user's home directory. This may be used by the remote control mechanism (the **--remote** option) to exchange messages between **pv** instances: if the */run/user/UID/* directory does not exist (where *UID* is the current user ID), then *$HOME/.pv/* will be used instead.

TMPDIR, TMP

The directory to create per-tty lock files for the terminal when using the **--cursor** option. If **TMPDIR** is set to a non-empty value, it is the directory under which lock files are created. Otherwise, if **TMP** is set, then it is used; and if neither are set, then */tmp* is used.

AUTHOR

Written by Andrew Wood, with patches submitted by various other people. Please see the package's ACKNOWLEDGEMENTS file for a complete list of contributors.

KNOWN PROBLEMS

The following problems are known to exist in **pv**:

- In some versions of **bash**(1) and **zsh**(1), the construct **<(pv filename)** will not output any progress to the terminal when run from an interactive shell, due to the subprocess being run in a separate process group from the one that owns the terminal. In these cases, use **--force**.

- The **-c** option does not work properly on Cygwin without **cygserver** running, if started near the bottom of the screen (IPC is needed to handle the terminal scrolling). To fix this, start **cygserver** before using **pv -c**.

- The **-R** option requires that either **/run/user/<uid>/** or **$HOME/** can be written to, for inter-process communication.

If you find any other problems, please report them.

REPORTING BUGS

Please report any bugs to **pv@ivarch.com**.

Alternatively, use the issue tracker linked from the **pv** home page:
<https://www.ivarch.com/programs/pv.shtml>

SEE ALSO

cat(1), **dialog**(1), **splice**(2), **open**(2) (for **O_DIRECT**)

COPYRIGHT