pv Pipe progress viewer

CONCEPTS

pv copies stdin (or files) to stdout unchanged and draws progress on stderr, so it can sit anywhere in a pipe.

producer cat, tar, dd
↓ stdout
pv counts and throttles
↓ stdout
consumer gzip, ssh, file

Without -s, pv cannot know the total: no percentage and no ETA, only a moving bar. Pass the size when you can.

BASICS

`pv <file> > <copy>`
Copy a file with progress bar

`pv <file> | <command>`
Feed a file to a command with progress

`<command> | pv > <file>`
Watch output of a command

`pv < <in> | <command>`
Same, reading from stdin

`pv -o <file>`
Write to a file instead of stdout

DISPLAY

`pv -p`
Progress bar only

`pv -t`
Elapsed time only

`pv -e`
Show ETA (needs size)

`pv -r`
Show current rate

`pv -a`
Show average rate

`pv -b`
Show bytes transferred

`pv -ptebar`
Everything at once

`pv -I`
Show estimated finish time

SIZE AND COUNTING

`pv -s <size>`
Expected total (K, M, G, T suffix)

`pv -s @<path>`
Take size from a file or directory

`pv -s $(du -sb <dir> | cut -f1)`
Size of a directory in bytes

`pv -s $(stat -c %s <file>)`
Size of a file for piped input

`pv -l`
Count lines, not bytes

RATE AND TRANSFER

`pv -L <rate>`
Limit throughput (bytes/s, suffixes)

`pv -q -L <rate>`
Throttle with no display

`pv -q`
No output

NAMES AND MULTIPLE BARS

`pv -N <name>`
Prefix the bar with a name

`pv -c`
Cursor positioning for several bars

`pv -cN in | gzip -d | pv -cN out`
Two labelled bars in one pipe

`pv -f`
Force display when stderr is no tty

FORMAT AND SCRIPTING

`pv -F '<format>'`
Custom display format

`pv -n`
Integer percentages on stderr

OTHER PROCESSES

`pv -d <pid>`
Watch all files a process has open

`pv -d <pid>:<fd>`
Watch one file descriptor

`pv -R <pid> -L <rate>`
Change a running pv's options