pv Pipe progress viewer

START HERE

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

`pv -N <name> -c`
Label a bar

`<cmd> | pv | <cmd>`
Watch a pipe

`pv -d <pid>`
Watch a running process

`pv -s <size> ...`
Give total for ETA

`pv -q -L <rate>`
Throttle silently

`pv -L 1m`
Cap speed at 1 MiB/s

`pv -F '%t %b'`
Custom display

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

CONCEPTS

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

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

Display goes to stderr, so redirecting stdout never hides it. It is suppressed if stderr is not a terminal unless you use -f.

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
Truncates an existing file.

`pv -V`
Show version

DISPLAY

`pv -p`
Progress bar only

`pv -t`
Elapsed time only
Handy as a stopwatch: sleep 10m | pv -t

`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

`pv -8`
Show bits instead of bytes

DISPLAY (cont.)

`pv -k`
Use powers of 1000 (SI)
Place it before the options it should affect.

`pv -v`
Print rate statistics at the end

`pv -T`
Show transfer buffer fill

`pv -u <style>`
Bar style: plain, block, granular, shaded

`pv -w <cols>`
Force display width

`pv -i <sec>`
Update interval (default 1)

`pv -m <sec>`
Window for average rate and ETA

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
With -l, -s is a line count.

`pv -0`
Count NUL-terminated records

`pv -S -s <size>`
Stop after size bytes

`pv -g`
Gauge for unknown size

`pv -W`
Wait for first byte before timing
Useful after an ssh password prompt.

`pv -D <sec>`
Show nothing for the first seconds

RATE AND TRANSFER

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

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

RATE AND TRANSFER (cont.)

`pv -q`
No output

`pv -B <bytes>`
Transfer buffer size

`pv -C`
Do not use splice
Needed for -A, %L and similar buffer-based displays.

`pv -E`
Skip read errors, pad with NULs

`pv -EE`
Same, report each file's error once

`pv -Z <bytes>`
Skip fixed blocks after an error

`pv -O`
Write sparse output

`pv -X`
Read and discard input

`pv -Y`
Sync to disk after each write

`pv -U <file>`
Store input first, then forward

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

`pv -x window`
Also show progress in window title

`pv -x process`
Also show progress in process title

FORMAT AND SCRIPTING

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

`pv -F '%b %t %r %p %e'`
Default format spelled out

`pv -n`
Integer percentages on stderr
For dialog --gauge and similar front ends.

FORMAT AND SCRIPTING (cont.)

`pv -nbt`
Numeric with bytes and time

`pv -F '%{progress-amount-only}'`
Percent only

Format sequences

Sequence	Shows
%p	Progress bar
%t	Elapsed time
%e	ETA
%I	Finish time
%r	Current rate
%a	Average rate
%b	Bytes
%T	Buffer fill
%N	Name
%%	Literal percent

OTHER PROCESSES

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

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

`pv -d =<name>`
Find the process by name

`pv -d @<listfile>`
Read targets from a file

`pv -Q <pid>`
Show progress of another pv

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

`pv -M both -- <command>`
Monitor a command's input and output

`pv -P <pidfile>`
Write pid to a file

RECIPES

Archive a directory with progress

```
tar cf - <dir>/ | pv -s $(du -sb <dir>/ | cut -f1) | gzip -9 > <dir>.tar.gz
```

Extract with progress

```
pv <archive>.tar.gz | tar xzf -
```

Image a disk, skip bad sectors

```
pv -EE /dev/<disk> > <disk>.img
pv <disk>.img > /dev/<disk>
```

Throttled copy over ssh

```
pv -L 2m <file> | ssh <host> 'cat > <file>'
# watch an already running gzip
pv -d $(pidof gzip)
```