



Linux Ubuntu 26.04 Manual Pages on command 'pipesz.1'

\$ man pipesz.1

PIPESZ(1) User Commands PIPESZ(1)

NAME

pipesz - set or examine pipe and FIFO buffer sizes

SYNOPSIS

pipesz [options] [--set size] [--] [command [argument] ...]

pipesz [options] --get

DESCRIPTION

Pipes and FIFOs maintain an internal buffer used to transfer data between the read end and the write end. In some cases, the default size of this internal buffer may not be appropriate. This program provides facilities to set and examine the size of these buffers.

The --set operation sets pipe buffer sizes. If it is specified, it must be specified with an explicit size. Otherwise, it is implied and the size is read from

/proc/sys/fs/pipe-max-size. The kernel may adjust size as described in fcntl(2). To

determine the actual buffer sizes set, use the --verbose option. If neither --file nor --fd are specified, --set acts on standard output.

The --set operation permits an optional command to execute after setting the pipe buffer sizes. This command is executed with the adjusted pipes.

The --get operation outputs data in a tabular format. The first column is the name of the pipe as passed to pipesz. File descriptors are named as "fd N". The second column is the size, in bytes, of the pipe's internal buffer. The third column is the number of unread bytes currently in the pipe. The columns are separated by tabs ('\t', ASCII 09h). If

--verbose is specified, a descriptive header is also emitted. If neither --file nor --fd are specified, --get acts on standard input.

Unless the `--check` option is specified, `pipesz` does not exit if it encounters an error while manipulating a file or file descriptor. This allows `pipesz` to be used generically without fear of disrupting the execution of pipelines should the type of certain files be later changed. For minimal disruption, the `--quiet` option prevents warnings from being emitted in these cases.

The kernel imposes limits on the amount of pipe buffer space unprivileged processes can use, though see [BUGS](#) below. The kernel will also refuse to shrink a pipe buffer if this would cause a loss of buffered data. See [pipe\(7\)](#) for additional details.

`pipesz` supports specifying multiple short options consecutively, in the usual `getopt(3)` fashion. The first non-option argument is interpreted as command. If command might begin with '-', use '--' to separate it from arguments to `pipesz`. In shell scripts, it is good practice to use '--' when parameter expansion is involved. `pipesz` itself does not read from standard input and does not write to standard output unless `--get`, `--help`, or `--version` are specified.

OPTIONS

`-g, --get`

Report the size of pipe buffers to standard output and exit. As a special behavior, if neither `--file` nor `--fd` are specified, standard input is examined. It is an error to specify this option in combination with `--set`.

`-s, --set size`

Set the size of the pipe buffers, in bytes. This option may be suffixed by K, M, G, KiB, MiB, or GiB to indicate multiples of 1024. Fractional values are supported in this case. Additional suffixes are supported but are unlikely to be useful. If this option is not specified, a default value is used, as described above. If this option is specified multiple times, a warning is emitted and only the last-specified size is used. As a special behavior, if neither `--file` nor `--fd` are specified, standard output is adjusted. It is an error to specify this option in combination with `--get`.

`-f, --file path`

Set the buffer size of the FIFO or pipe at path, relative to the current working directory. You may specify this option multiple times to affect different files, and you may do so in combination with `--fd`. Generally, this option is used with FIFOs, but it will also operate on anonymous pipes such as those found in `/proc/PID/fd`. Changes to the buffer size of FIFOs are not preserved across system restarts.

-n, --fd fd

Set the buffer size of the pipe or FIFO passed to pipesz as the specified file descriptor number. You may specify this option multiple times to affect different file descriptors, and you may do so in combination with --file. Shorthand options are provided for the common cases of fd 0 (stdin), fd 1 (stdout), and fd 2 (stderr). These should suffice in most cases.

-i, --stdin

Shorthand for --fd 0.

-o, --stdout

Shorthand for --fd 1.

-e, --stderr

Shorthand for --fd 2.

-c, --check

Exit, without executing command, in case of any error while manipulating a file or file descriptor. The default behavior if this is not specified is to emit a warning to standard error and continue.

-q, --quiet

Do not diagnose non-fatal errors to standard error. This option does not affect the normal output of --get, --verbose, --help, or --version.

-v, --verbose

If specified with --get, pipesz will emit a descriptive header above the table.

Otherwise, if specified, pipesz will print the actual buffer sizes set by the kernel to standard error.

-h, --help

Display help text and exit.

-V, --version

Display version and exit.

EXAMPLES

```
pipesz dd if=file bs=1M | ...
```

Runs dd(1) with an expanded standard output pipe, allowing it to avoid context switches when piping around large blocks.

```
pipesz -s1M -cf /run/my-service.fifo
```

Sets the pipe buffer size of a service FIFO to 1,048,576 bytes. If the buffer size could

not be set, pipesz exits with an error.

```
echo hello | pipesz -g
```

Prints the size of pipe used by the shell to pass input to pipesz. Since pipesz does not read standard input, it may also report 6 unread bytes in the pipe, depending on relative timings.

```
find /proc/PID/fd -exec pipesz -gqf '{}' ';'
```

Prints the size and number of unread bytes of all pipes in use by PID. If some pipes are routinely full, pipesz might be able to mitigate a processing bottleneck.

NOTES

Linux supports adjusting the size of pipe buffers since kernel 2.6.35. This release also introduced `/proc/sys/fs/pipe-max-size`.

This program uses `fcntl(2)` `F_GETPIPE_SZ`/`F_SETPIPE_SZ` to get and set pipe buffer sizes.

This program uses `ioctl(2)` `FIONREAD` to report the amount of unread data in pipes. If for some reason this fails, the amount of unread data is reported as 0.

BUGS

Before Linux 4.9, some bugs affect how certain resource limits are enforced when setting pipe buffer sizes. See `pipe(7)` for details.

AUTHORS

Nathan Sharp <nwsharp@live.com>

SEE ALSO

`pipe(7)`

REPORTING BUGS

For bug reports, use the issue tracker <<https://github.com/util-linux/util-linux/issues>>.

AVAILABILITY

The pipesz command is part of the util-linux package which can be downloaded from Linux Kernel Archive <<https://www.kernel.org/pub/linux/utils/util-linux/>>.

util-linux 2.41.3

2025-12-15

PIPESZ(1)