

**NAME**

**tmpfs** - in-memory file system

**SYNOPSIS**

To compile this driver into the kernel, place the following line in your kernel configuration file:

```
options TMPFS
```

Alternatively, to load the driver as a module at boot time, place the following line in loader.conf(5):

```
tmpfs_load="YES"
```

**DESCRIPTION**

The **tmpfs** driver implements an in-memory, or **tmpfs** file system. The filesystem stores both file metadata and data in main memory. This allows very fast and low latency accesses to the data. The data is volatile. An amount or system reboot invalidates it. These properties make the filesystem's mounts suitable for fast scratch storage, like */tmp*.

If the system becomes low on memory and swap is configured (see `swapon(8)`), the system can transfer file data to swap space, freeing memory for other needs. Metadata, including the directory content, is never swapped out by the current implementation. Keep this in mind when planning the mount limits, especially when expecting to place many small files on a tmpfs mount.

When `mmap(2)` is used on a file from a tmpfs mount, the swap VM object managing the file pages is used to implement mapping and avoid double-copying of the file data. This quirk causes process inspection tools, like `procstat(1)`, to report anonymous memory mappings instead of file mappings.

**OPTIONS**

The following options are available when mounting **tmpfs** file systems:

- |             |  |
|-------------|--|
| <b>gid</b>  | Specifies the group ID of the root inode of the file system. Defaults to the mount point's GID.                  |
| <b>uid</b>  | Specifies the user ID of the root inode of the file system. Defaults to the mount point's UID.                   |
| <b>mode</b> | Specifies the mode (in octal notation) of the root inode of the file system. Defaults to the mount point's mode. |
| <b>none</b> | Do not use namecache to resolve names to files for the created mount. This saves                                 |

memory, but currently might impair scalability for highly used mounts on large machines.

- inodes** Specifies the maximum number of nodes available to the file system. If not specified, the file system chooses a reasonable maximum based on the file system size, which can be limited with the **size** option.
- size** Specifies the total file system size in bytes, unless suffixed with one of k, m, g, t, or p, which denote byte, kilobyte, megabyte, gigabyte, terabyte and petabyte respectively. If zero (the default) or a value larger than `SIZE_MAX - PAGE_SIZE` is given, the available amount of memory (including main memory and swap space) will be used.
- maxfilesize** Specifies the maximum file size in bytes. Defaults to the maximum possible value.
- easize** Specifies the maximum memory size used by extended attributes in bytes. Defaults to 16 megabytes.

## EXAMPLES

Mount a **tmpfs** memory file system:

```
mount -t tmpfs tmpfs /tmp
```

Configure a **tmpfs** mount via `fstab(5)`:

```
tmpfs /tmp tmpfs rw 0 0
```

## SEE ALSO

`procstat(1)`, `mmap(2)`, `nmount(2)`, `unmount(2)`, `fstab(5)`, `mdmfs(8)`, `mount(8)`, `swapinfo(8)`, `swapon(8)`

## HISTORY

The **tmpfs** driver first appeared in FreeBSD 7.0.

## AUTHORS

The **tmpfs** kernel implementation was written by Julio M. Merino Vidal <[jmmv@NetBSD.org](mailto:jmmv@NetBSD.org)> as a Google Summer of Code project.

Rohit Jalan and others ported it from NetBSD to FreeBSD.

This manual page was written by Xin LI <[delphij@FreeBSD.org](mailto:delphij@FreeBSD.org)>.