Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'tmpfs.5' command

$ man tmpfs.5

TMPFS(5) Linux Programmer's Manual TMPFS(5)

NAME

tmpfs - a virtual memory filesystem

DESCRIPTION

The tmpfs facility allows the creation of filesystems whose contents

reside in virtual memory. Since the files on such filesystems typi?

cally reside in RAM, file access is extremely fast.

The filesystem is automatically created when mounting a filesystem with

the type tmpfs via a command such as the following:

```
$ sudo mount -t tmpfs -o size=10M tmpfs /mnt/mytmpfs
```

A tmpfs filesystem has the following properties:

* The filesystem can employ swap space when physical memory pressure

demands it.

* The filesystem consumes only as much physical memory and swap space

as is required to store the current contents of the filesystem.

* During a remount operation (mount -o remount), the filesystem size

can be changed (without losing the existing contents of the filesys?

tem).

If a tmpfs filesystem is unmounted, its contents are discarded (lost).

Mount options

The tmpfs filesystem supports the following mount options:

size=bytes

Specify an upper limit on the size of the filesystem. The size is given in bytes, and rounded up to entire pages.

The size may have a k, m, or g suffix for Ki, Mi, Gi (binary kilo (kibi), binary mega (mebi) and binary giga (gibi)).

The size may also have a % suffix to limit this instance to a percentage of physical RAM.

The default, when neither size nor nr_blocks is specified, is size=50%.

nr_blocks=blocks

The same as size, but in blocks of PAGE_CACHE_SIZE.

Blocks may be specified with k, m, or g suffixes like size, but not a % suffix.

nr_inodes=inodes

The maximum number of inodes for this instance. The default is half of the number of your physical RAM pages, or (on a machine with highmem) the number of lowmem RAM pages, whichever is smaller.

Inodes may be specified with k, m, or g suffixes like size, but not a % suffix.

mode=mode

Set initial permissions of the root directory.

gid=gid (since Linux 2.5.7)

Set the initial group ID of the root directory.

uid=uid (since Linux 2.5.7)

Set the initial user ID of the root directory.

huge=huge_option (since Linux 4.7.0)

Set the huge table memory allocation policy for all files in this instance (if CONFIG_TRANSPARENT_HUGE_PAGECACHE is enabled).

The huge_option value is one of the following:

never Do not allocate huge pages. This is the default.

always Attempt to allocate huge pages every time a new page is needed.

within_size

Only allocate huge page if it will be fully within i_size. Also respect fadvise(2)/madvise(2) hints

advise Only allocate huge pages if requested with fadvise(2)/madvise(2).

deny For use in emergencies, to force the huge option off from all mounts.

force Force the huge option on for all mounts; useful for testing.

mpol=mpol_option (since Linux 2.6.15)

Set the NUMA memory allocation policy for all files in this instance (if CONFIG_NUMA is enabled).

The mpol_option value is one of the following:

default

Use the process allocation policy (see set_mempolicy(2)).

prefer:node

Preferably allocate memory from the given node.

bind:nodelist

Allocate memory only from nodes in nodelist.

interleave

Allocate from each node in turn.

interleave:nodelist

Allocate from each node of in turn.

local Preferably allocate memory from the local node.

In the above, nodelist is a comma-separated list of decimal numbers and ranges that specify NUMA nodes. A range is a pair of hyphen-separated decimal numbers, the smallest and largest node numbers in the range. For example, mpol=bind:0-3,5,7,9-15.

VERSIONS

The tmpfs facility was added in Linux 2.4, as a successor to the older ramfs facility, which did not provide limit checking or allow for the use of swap space.

NOTES

In order for user-space tools and applications to create tmpfs filesystems, the kernel must be configured with the CONFIG_TMPFS option.

The tmpfs filesystem supports extended attributes (see xattr(7)), but user extended attributes are not permitted.

An internal shared memory filesystem is used for System V shared memory (shmget(2)) and shared anonymous mappings (mmap(2) with the MAP_SHARED and MAP_ANONYMOUS flags). This filesystem is available regardless of whether the kernel was configured with the CONFIG_TMPFS option.

A tmpfs filesystem mounted at /dev/shm is used for the implementation of POSIX shared memory (shm_overview(7)) and POSIX semaphores (sem_overview(7)).

The amount of memory consumed by all tmpfs filesystems is shown in the Shmem field of /proc/meminfo and in the shared field displayed by free(1).

The tmpfs facility was formerly called shmfs.

SEE ALSO

df(1), du(1), memfd_create(2), mmap(2), set_mempolicy(2), shm_open(3), mount(8)

The kernel source files Documentation/filesystems/tmpfs.txt and Docu?
mentation/admin-guide/mm/transhuge.rst.

COLOPHON

This page is part of release 5.10 of the Linux man-pages project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at https://www.kernel.org/doc/man-pages/.