NAME

tmpfs – memory based filesystem

SYNOPSIS

#include <sys/mount.h>
mount (*special*, *directory*, **MS_DATA, "tmpfs", NULL, 0);**

DESCRIPTION

tmpfs is a memory based filesystem which uses kernel resources relating to the VM system and page cache as a filesystem. Once mounted, a **tmpfs** filesystem provides standard file operations and semantics. **tmpfs** is so named because files and directories are not preserved across reboot or unmounts, all files residing on a **tmpfs** filesystem that is unmounted will be lost.

tmpfs filesystems can be mounted with the command:

mount –F tmpfs swap *directory*

Alternatively, to mount a **tmpfs** filesystem on **/tmp** at multi-user startup time (and maximizing possible performance improvements), add the following line to **/etc/vfstab**:

swap – /tmp tmpfs – yes –

tmpfs is designed as a performance enhancement which is achieved by caching the writes to files residing on a **tmpfs** filesystem. Performance improvements are most noticeable when a large number of short lived files are written and accessed on a **tmpfs** filesystem. Large compilations with **tmpfs** mounted on **/tmp** are a good example of this.

Users of **tmpfs** should be aware of some constraints involved in mounting a **tmpfs** filesystem. The resources used by **tmpfs** are the same as those used when commands are executed (for example, swap space allocation). This means that large sized **tmpfs** files can affect the amount of space left over for programs to execute. Likewise, programs requiring large amounts of memory use up the space available to **tmpfs**. Users running into this constraint (for example, running out of space on **tmpfs**) can allocate more swap space by using the **swap**(1M) command.

Another constraint is that the number of files available in a **tmpfs** filesystem is calculated based on the physical memory of the machine and not the size of the swap device/partition. If you have too many files, **tmpfs** will print a warning message and you will be unable to create new files. You cannot increase this limit by adding swap space.

Normal filesystem writes are scheduled to be written to a permanent storage medium along with all control information associated with the file (for example, modification time, file permissions). **tmpfs** control information resides only in memory and never needs to be written to permanent storage. File data remains in core until memory demands are sufficient to cause pages associated with **tmpfs** to be reused at which time they are copied out to swap.

An additional mount option can be specified to control the size of an individual **tmpfs** filesystem.

SEE ALSO

df(1M), **mount**(1M), **mount_tmpfs**(1M), **swap**(1M), **mmap**(2), **mount**(2), **umount**(2), **vfstab**(4)

System Administration Guide, Volume I

DIAGNOSTICS

If **tmpfs** runs out of space, one of the following messages will be printed to the console.

directory: **File system full, swap space limit exceeded**

This message is printed because a page could not be allocated while writing to a file. This can occur if **tmpfs** is attempting to write more than it is allowed, or if currently executing programs are using a lot of memory. To make more space available, remove unnecessary files, exit from some programs, or allocate more swap space using **swap**(1M).

directory: **File system full, memory allocation failed.**

tmpfs ran out of physical memory while attempting to create a new file or directory. Remove unnecessary files or directories or install more physical memory.

WARNINGS

Files and directories on a **tmpfs** filesystem are not preserved across reboots or unmounts. Command scripts or programs which count on this will not work as expected.

NOTES

Compilers do not necessarily use **/tmp** to write intermediate files therefore missing some significant performance benefits. This can be remedied by setting the environment variable **TMPDIR** to **/tmp**. Compilers use the value in this environment variable as the name of the directory to store intermediate files.

swap to a **tmpfs** file is not supported.

df(1M) output is of limited accuracy since a **tmpfs** filesystem size is not static and the space available to **tmpfs** is dependent on the swap space demands of the entire system.