NAME

   mdmfs, **mount_mfs** - configure and mount an in-memory file system using the md(4) driver or the tmpfs(5) filesystem

SYNOPSIS

   mdmfs [**-DLlMNnPStTUX**] [**-a** *maxcontig*] [**-b** *block-size*] [**-c** *blocks-per-cylinder-group*]
           [**-d** *max-extent-size*] [**-E** *path-mdconfig*] [**-e** *maxbpg*] [**-F** *file*] [**-f** *frag-size*] [**-i** *bytes*] [**-k** *skel*]
           [**-m** *percent-free*] [**-O** *optimization*] [**-o** *mount-options*] [**-p** *permissions*] [**-s** *size*] [**-T** *fstype*]
           [**-v** *version*] [**-w** *user*:*group*] *md-device mount-point*

DESCRIPTION

   The **mdmfs** utility is designed to be a work-alike and look-alike of the deprecated **mount_mfs**. The end result is essentially the same, but is accomplished in a completely different way. Based on *md-device*, the **mdmfs** utility either creates a tmpfs(5) filesystem, or it configures an md(4) disk using mdconfig(8), puts a UFS file system on it (unless **-P** was specified) using newfs(8), and mounts it using mount(8). It can handle geom_uzip(4) compressed disk images, as long as the kernel supports this GEOM class. All the command line options are passed to the appropriate program at the appropriate stage in order to achieve the desired effect.

   When *md-device* is 'auto', **mdmfs** uses tmpfs(5) if it is present in the kernel or can be loaded as a module, otherwise it falls back to using md(4) auto-unit as if 'md' had been specified.

   When *md-device* is 'tmpfs', **mdmfs** mounts a tmpfs(5) filesystem, translating the **-s** size option, if present, into a '-o size=' mount option. Any **-o** options on the command line are passed through to the tmpfs(5) mount. Options specific to mdconfig(8) or newfs(8) are ignored.

   When *md-device* does not result in tmpfs(5) being used, then an md(4) device is configured instead. By default, **mdmfs** creates a swap-based (MD_SWAP) disk with soft-updates enabled and mounts it on *mount-point*. It uses the md(4) device specified by *md-device*. If *md-device* is 'md' (no unit number), it will use md(4)'s auto-unit feature to automatically select an unused device. Unless otherwise specified with one of the options below, it uses the default arguments to all the helper programs.

   The following options are available. Where possible, the option letter matches the one used by **mount_mfs** for the same thing.

   -a *maxcontig*
           Specify the maximum number of contiguous blocks that will be laid out before forcing a rotational delay (see the **-d** option).

   -b *block-size*

    The block size of the file system, in bytes.

-c *blocks-per-cylinder-group*
    The number of blocks per cylinder group in the file system.

-D   If not using auto-unit, do not run mdconfig(8) to try to detach the unit before attaching it.

-d *max-extent-size*
    The file system may choose to store large files using extents. This parameter specifies the largest extent size that may be used. It is presently limited to its default value which is 16 times the file system blocksize.

-E *path-mdconfig*
    Use *path-mdconfig* as a location of the mdconfig(8) utility.

-e *maxbpg*
    Indicate the maximum number of blocks any single file can allocate out of a cylinder group before it is forced to begin allocating blocks from another cylinder group.

-F *file* Create a vnode-backed (MD_VNODE) memory disk backed by *file*.

-f *frag-size*
    The fragment size of the file system in bytes.

-i *bytes*
    Number of bytes per inode.

-k *skel*
    Copy the content of directory *skel* into *mount-point*.

-l   Enable multilabel MAC on the new file system.

-L   Show the output of the helper programs. By default, it is sent to */dev/null*.

-M   Create a malloc(9) backed disk (MD_MALLOC) instead of a swap-backed disk.

-m *percent-free*
    The percentage of space reserved for the superuser.

-N   Do not actually run the helper programs. This is most useful in conjunction with **-X**.

-n   Do not create a *.snap* directory on the new file system.

-O *optimization*

      Select the optimization preference; valid choices are **space** and **time**, which will optimize for minimum space fragmentation and minimum time spent allocating blocks, respectively.

-o *mount-options*

      Specify the mount options with which to mount the file system. See mount(8) for more information.

-P   Preserve the existing file system; do not run newfs(8). This only makes sense if **-F** is specified to create a vnode-backed disk.

-p *permissions*

      Set the file (directory) permissions of the mount point *mount-point* to *permissions*. The *permissions* argument can be in any of the mode formats recognized by chmod(1). If symbolic permissions are specified, the operation characters "+" and "-" are interpreted relative to the initial permissions of "a=rwx".

-S   Do not enable soft-updates on the file system.

-s *size*

      Specify the size of the disk to create. This only makes sense if **-F** is *not* specified. That is, this will work when the backing storage is some form of memory, as opposed to a fixed-size file. The size may include the usual SI suffixes (k, m, g, t, p). A number without a suffix is interpreted as a count of 512-byte sectors.

-t   Turn on the TRIM enable flag for newfs(8). When used with a file system that issue BIO_DELETE bio requests, md(4) returns deleted blocks to the system memory pool.

-T *fstype*

      Specify a file system type for a vnode-backed memory disk. Any file system supported by mount(8) command can be specified. This option only makes sense when **-F** and **-P** are used.

-U   Enable soft-updates on the file system. This is the default, and is accepted only for compatibility. It is only really useful to negate the **-S** flag, should such a need occur.

-v *version*

      Specify the UFS version number for use on the file system; it may be either 1 or 2. The default is derived from the default of the newfs(8) command.

-w *user*:*group*

   Set the owner and group to *user* and *group*, respectively. The arguments have the same
   semantics as with chown(8), but specifying just a *user* or just a *group* is not supported.

-X   Print what command will be run before running it, and other assorted debugging information.

The **-F** and **-s** options are passed to mdconfig(8) as **-f** and **-s**, respectively. The **-a**, **-b**, **-c**, **-d**, **-e**, **-f**, **-i**, **-m**
and **-n** options are passed to newfs(8) with the same letter. The **-O** option is passed to newfs(8) as **-o**.
The **-o** option is passed to mount(8) with the same letter. The **-T** option is passed to mount(8) as **-t**. For
information on semantics, refer to the documentation of the programs that the options are passed to.

EXAMPLES

Create and mount a 32 megabyte swap-backed file system on */tmp*:

   mdmfs -s 32m md /tmp

The same file system created as an entry in */etc/fstab*:

   md /tmp mfs rw,-s32m 2 0

Create and mount a 16 megabyte malloc-backed file system on */tmp* using the */dev/md1* device;
furthermore, do not use soft-updates on it and mount it **async**:

   mdmfs -M -S -o async -s 16m md1 /tmp

Create and mount a geom_uzip(4) based compressed disk image:

   mdmfs -P -F foo.uzip -oro md.uzip /tmp/

Mount the same image, specifying the */dev/md1* device:

   mdmfs -P -F foo.uzip -oro md1.uzip /tmp/

Configure a vnode-backed file system and mount its first partition, using automatic device numbering:

   mdmfs -P -F foo.img mds1a /tmp/

Mount a vnode-backed cd9660 file system using automatic device numbering:

   mdmfs -T cd9660 -P -F foo.iso md /tmp

COMPATIBILITY

The **mdmfs** utility, while designed to be compatible with **mount_mfs**, can be useful by itself. Since **mount_mfs** had some silly defaults, a "compatibility" mode is provided for the case where bug-to-bug compatibility is desired.

Compatibility is enabled by starting **mdmfs** with the name mount_mfs or mfs (as returned by getprogname(3)). In this mode, the following behavior, as done by **mount_mfs**, is duplicated:

- The file mode of *mount-point* is set by default to 01777 as if **-p** *1777* was given on the command line.

SEE ALSO

md(4), fstab(5), tmpfs(5), mdconfig(8), mount(8), newfs(8)

HISTORY

The **mdmfs** utility appeared in FreeBSD 5.0.

AUTHORS

Dima Dorfman