NAME

newfs – construct a new UFS file system

SYNOPSIS

newfs [–**Nv**] [*mkfs-options*] *raw-device*

DESCRIPTION

newfs is a "friendly" front-end to the **mkfs**(1M) program for making UFS file systems on disk partitions. **newfs** calculates the appropriate parameters to use and calls **mkfs**.

raw-device is the name of a raw special device residing in **/dev/rdsk** (for example, **/dev/rdsk/c0t0d0s6**) on which to create the file system.

If you run **newfs** interactively (that is, standard input is a tty), **newfs** will prompt for confirmation before making the file system.

If the –**N** option is not specified and the inodes of the device are not randomized, **newfs** will call **fsirand**(1M).

Note: You must be super-user to use this command.

OPTIONS

–**N** Print out the file system parameters that would be used in creating the file system without actually creating the file system. **fsirand**(1M) is not called here.

–**v** Verbose. **newfs** prints out its actions, including the parameters passed to **mkfs**.

mkfs-options
Options that override the default parameters are:

–**a** *apc* The number of alternate blocks per cylinder (SCSI devices only) to reserve for bad block replacement. The default is **0**.

–**b** *bsize* The logical block size of the file system in bytes (either 4096 or 8192). The default is **8192**.

–**c** *cgize* The number of cylinders per cylinder group (ranging from 1 to 32). The default is **16**.

–**d** *gap* Rotational delay. The expected time (in milliseconds) to service a transfer completion interrupt and initiate a new transfer on the same disk. It is used to decide how much rotational spacing to place between successive blocks in a file. This parameter can be subsequently changed using the **tunefs**(1M) command. The default is disk-type dependent.

–**f** *fragsize* The smallest amount of disk space in bytes to allocate to a file. The values must be a power of two selected from the range 512 to the logical block size. If logical block size is 4096, legal values are 512, 1024, 2048 and 4096; if logical block size is 8192, 8192 is also a legal value. The default is **1024**.

–**i** *nbpi* The number of bytes per inode. This specifies the density of inodes in the file system. The number is divided into the total size of the file system to determine the fixed number of inodes to create. It should reflect the expected average size of files in the file system. If fewer inodes are desired, a larger number should be used; to create more inodes a smaller number should be given. The default is **2048**.

–m *free* The minimum percentage of free space to maintain in the file system. This space is off-limits to normal users. Once the file system is filled to this threshold, only the super-user can continue writing to the file system. This parameter can be subsequently changed using the **tunefs**(1M) command. The default is **10%**.

–n *nrpos* The number of different rotational positions in which to divide a cylinder group. The default is **8**.

–o *opt* (**space** or **time**). The file system can either be instructed to try to minimize the time spent allocating blocks, or to try to minimize the space fragmentation on the disk. If the minimum free space threshold (as specified by the **–m** option) is less than 10%, space optimization will be used.

–r *rpm* The speed of the disk in revolutions per minute. The default is **3600**.

–s *size* The size of the file system in sectors. The default is to use the entire partition.

–t *ntrack* The number of tracks per cylinders on the disk. The default is taken from the disk label.

–C *maxcontig* The maximum number of blocks, belonging to one file, that will be allocated contiguously before inserting a rotational delay. For a 4K file system, the default is **14**; for an 8K file system, the default is **7**. This parameter can be subsequently changed using the **tunefs**(1M) command.

Note: This parameter also controls clustering. Regardless of the value of *gap*, clustering is enabled only when *maxcontig* is greater than 1. Clustering allows higher I/O rates for sequential I/O and is described in **tunefs**(1M).

EXAMPLES

The following example verbosely displays the parameters for the raw special device, **c0t0d0s6**, but does not actually create a new file system:

```
example# newfs –Nv /dev/rdsk/c0t0d0s6
mkfs –F ufs –o N /dev/rdsk/c0t0d0s6 1112940 54 15 8192 1024 16 10 60 2048 t 0 –1 8
/dev/rdsk/c0t0d0s6:    1112940 sectors in 1374 cylinders of 15 tracks, 54 sectors
       569.8MB in 86 cyl groups (16 c/g, 6.64MB/g, 3072 i/g)
super-block backups (for fsck –b #) at:
 32, 13056, 26080, 39104, 52128, 65152, 78176, 91200, 104224,
 ...
example#
```

SEE ALSO

fsck(1M), **fsck_ufs**(1M), **fsirand**(1M), **mkfs**(1M), **mkfs_ufs**(1M), **tunefs**(1M), **fs_ufs**(4)

DIAGNOSTICS

newfs: *special* **No such file or directory**
The device specified does not exist, or a disk partition was not specified.

*special***: cannot open**
You must be super-user to use this command.