NAME

mkfs − make a file system

SYNOPSIS

mkfs [**−Ldot**] [**−i** *inodes*] [**−b** *blocks*] *special prototype*

OPTIONS

−L Make a listing on standard output

−d Use mod time of *mkfs* binary for all files

−o Use a drive other than 0 or 1 (safety precaution)

−t Do not test if file system fits on the medium

−1 Make a version 1 file system (for backward compatibility)

−i Number of i-nodes (files)

−b Filesystem size

EXAMPLES

mkfs /dev/fd1 proto # Make a file system on */dev/fd1*

mkfs -b 360 /dev/fd1 # Make empty 360 block file system

mkfs /dev/fd1 360 # Alternate way to specify the size

DESCRIPTION

Mkfs builds a file system and copies specified files to it. The prototype file tells which directories and files to copy to it. If the prototype file cannot be opened, and its name is just a string of digits, an empty file system will be made with the specified number of blocks. A sample prototype file follows. The text following the # sign in the example below is comment. In real prototype files, comments are not allowed.

```
boot                                             # boot block file (ignored)
360 63                                           # blocks and i-nodes
d--755 1 1                                       # root directory
  bin    d--755  2 1                             # bin dir: mode (755), uid (2), gid (1)
         sh      ---755 2 1 /user/bin/shell      # shell has mode rwxr-xr-x
         mv      -u-755 2 1 /user/bin/mv         # u = SETUID bit
         login   -ug755 2 1 /user/bin/login      # SETUID and SETGID
  $                                              # end of /bin
  dev    d--755 2 1                              # special files: tty (char), fd0 (block)
         tty     c--777 2 1 4 0                  # uid=2, gid=1, major=4, minor=0
         fd0     b--644 2 1 2 0 360              # uid, gid, major, minor, blocks
  $                                              # end of /dev
  user   d--755 12 1                             # user dir: mode (755), uid (12), gid (1)
         ast     d--755 12 1                     # /user/ast
         $                                       # /user/ast is empty
  $                                              # end of /user
$                                                # end of root directory
```

The first entry on each line (except the first 3 and the $ lines, which terminate directories) is the name the file or directory will get on the new file system. Next comes its mode, with the first character being **−dbc** for regular files, directories, block special files and character special files, respectively. The next two characters are used to specify the SETUID and SETGID bits, as shown above. The last three characters of the mode are the *rwx* protection bits.

Following the mode are the uid and gid. For special files, the major and minor devices are needed. The size in blocks must also be specified for block special files (the MINIX block size is 1K; this can only be changed by changing *BLOCK_SIZE* and then recompiling the operating system).

The maximum size of a file system is 1 Gb for a version 2 file system, and 64 Mb for a version 1 file system. Alas the 8086 *fsck* runs out of memory on a V2 file system larger than 128 Mb, so for the 8086 version of MINIX you have to limit yourself to file systems of that size.

SEE ALSO

mkproto(1), **fsck**(1), **mount**(1).