

## NAME

filesystem — format of system volume

## DESCRIPTION

Every file system storage volume (e.g. RP04 disk) has a common format for certain vital information. Every such volume is divided into a certain number of 256 word (512 byte) blocks. Block 0 is unused and is available to contain a bootstrap program or other information.

Block 1 is the *super block*. Starting from its first word, the format of a super-block is:

```

/*          @(#)filsys.h    3.1          */
/*
 * Definition of the unix super block.
 * The root super block is allocated and
 * read in iinit/alloc.c. Subsequently
 * a super block is allocated and read
 * with each mount (smount/sys3.c) and
 * released with unmount (sumount/sys3.c).
 * A disk block is ripped off for storage.
 * See alloc.c for general alloc/free
 * routines for free list and I list.
 */
struct      filsys
{
    char      *s_ysize;      /* size in blocks of I list */
    char      *s_fsize;     /* size in blocks of entire volume */
    int       s_nfree;      /* number of in core free blocks (0-100) */
    int       s_free[100];  /* in core free blocks */
    int       s_ninode;     /* number of in core I nodes (0-100) */
    int       s_inode[100]; /* in core free I nodes */
    char      s_flock;      /* lock during free list manipulation */
    char      s_ilock;      /* lock during I list manipulation */
    char      s_fmod;       /* super block modified flag */
    char      s_ronly;      /* mounted read-only flag */
    long      s_time;       /* current date of last update */
    int       pad[40];
    int       s_tfree;      /* Total free, for subsystem examination */
    int       s_tinode;     /* Free inodes, for subsystem examination */
    char      s_fname[6];   /* File system name */
    char      s_fpack[6];   /* File system pack name */
};

```

*Isize* is the number of blocks devoted to the i-list, which starts just after the super-block, in block 2. *Fsize* is the first block not potentially available for allocation to a file. These numbers are used by the system to check for bad block numbers; if an 'impossible' block number is allocated from the free list or is freed, a diagnostic is written on the on-line console. Moreover, the free array is cleared, so as to prevent further allocation from a presumably corrupted free list.

The free list for each volume is maintained as follows. The *free* array contains, in *free[1]*, ... , *free[nfree-1]*, up to 49 numbers of free blocks. *free[0]* is the block number of the head of a chain of blocks constituting the free list. The first long in each free-chain block is the number (up to 50) of free-block numbers listed in the next 50 longs of this chain member. The first of these 50 blocks is the link to the next member of the chain. To allocate a block: decrement *nfree*, and the new block is *free[nfree]*. If the new block number is 0, there are no blocks left, so give an error. If *nfree* became 0, read in the block named by the new block number, replace *nfree* by its first word, and copy the block numbers in the next 50 longs into the *free* array. To free a block, check if *nfree* is 50; if so, copy *nfree* and the *free* array into it, write it out, and set *nfree* to 0. In any event set *free[nfree]* to the freed block's number and increment *nfree*.

*Tfree* is the total free blocks available in the file system.

*Ninode* is the number of free i-numbers in the *inode* array. To allocate an i-node: if *ninode* is greater than 0, decrement it and return *inode[ninode]*. If it was 0, read the i-list and place the numbers of all free inodes (up to 100) into the *inode* array, then try again. To free an i-node, provided *ninode* is less than 100, place its number into *inode[ninode]* and increment *ninode*. If *ninode* is already 100, do not bother to enter the freed i-node into any table. This list of i-nodes is only to speed up the allocation process; the information as to whether the inode is really free or not is maintained in the inode itself.

*Tinode* is the total free inodes available in the file system.

*Flock* and *ilock* are flags maintained in the core copy of the file system while it is mounted and their values on disk are immaterial. The value of *fmod* on disk is likewise immaterial; it is used as a flag to indicate that the super-block has changed and should be copied to the disk during the next periodic update of file system information.

*Ronly* is a read-only flag to indicate write-protection.

*Time* is the last time the super-block of the file system was changed, and is a double-precision representation of the number of seconds that have elapsed since 0000 Jan. 1, 1970 (GMT). During a reboot, the *time* of the super-block for the root file system is used to set the system's idea of the time.

*Fname* is the name of the file system and *fpack* is the name of the pack.

I-numbers begin at 1, and the storage for i-nodes begins in block 2. Also, i-nodes are 64 bytes long, so 8 of them fit into a block. Therefore, i-node *i* is located in block  $(i + 15) / 8$ , and begins  $64 * ((i + 15) \text{ mod } 8)$  bytes from its start. I-node 1 is reserved for future use. I-node 2 is reserved for the root directory of the file system, but no other i-number has a built-in meaning. Each i-node represents one file. For the format of an inode and its flags, see *inode(5)*.

#### FILES

/usr/include/sys/filsys.h  
/usr/include/sys/stat.h  
/usr/include/sys/types.h  
/usr/include/sys/param.h

#### SEE ALSO

*inode(1)*, *mkfs(1M)*, *stat(2)*, *stat:o(2)*, *inode(5)*