THE FILE
SYSTEM
UNIX SYSTEM FILES

A *file* is a named place to store information

- **regular or ordinary files**: store text (e.g. letters, memos), data, programs
- **directory files**: store the names of other files, act as table of contents, used to group related files
- **special files**: used for devices (e.g. printers, terminals)
RELATIVE PATH NAMES

- Never begin with /
- Search begins at current directory
- Current directory .
- Parent directory ..

$ cat calendar

$ ls ..

$ ls misc/medical

$ cat project/notes/form
2.1 The basics of files

- A file is a sequence of bytes

- No structure is imposed

- No meaning attached to contents

- od -c command.
  
  - c char - b octal - x hex

- read system call

- No special EOF

- Ctrl-d : send what was typed to prgrm reading terminal

2.2 What's in a file?

- file command

- magic number

- no distinction between files

- drawback sort/ed/grep bin file
cassatt% cat egfile
abc
def
gi
ijklm

nopoq
cassatt% od egfile
0000000 060542 061412 062145 063012 063550 005151 065153 066155
0000020 005156 067560 070412
0000025
cassatt% od -d egfile
0000000 24930 25354 25701 26122 26472 02665 27243 27757
0000020 02670 28528 28938
0000026
cassatt% od -x egfile
0000000 6162 630a 6465 660a 6768 0a69 6a6b 6c6d
0000020 0a6e 6f70 710a
0000026
cassatt% wc egfile
  5    5   22 egfile
cassatt%
$ od -b egfile
0000000 141 142 143 012 144 145 146 012 147 150 012 151 152 153 154 155
0000020 012 156 157 160 161 012
0000026
$
$ od -c egfile
0000000  a  b  c  \n  d  e  f  \n  g  h  \n  i  j  k  l  m
0000020  \n  n  o  p  q  \n
0000026
$
$ od -bc egfile
0000000  141 142 143 012 144 145 146 012 147 150 012 151 152 153 154 155.
          a  b  c  \n  d  e  f  \n  g  h  \n  i  j  k  l  m
0000020  012 156 157 160 161 012
             \n  n  o  p  q  \n
0000026
$
$
NAME
dir_ufs, dir - format of ufs directories

SYNOPSIS
#include <sys/param.h>
#include <sys/types.h>
#include <sys/fs/ufs_fsdir.h>

DESCRIPTION
A directory consists of some number of blocks of DIRBLKSIZE bytes, where
DIRBLKSIZE is chosen such that it can be transferred to disk in a single atomic
operation (for example, 512 bytes on most machines).

Each DIRBLKSIZE-byte block contains some number of directory
entry structures, which are of variable length. Each directory
entry has a struct direct at the front of it, containing
its inode number, the length of the entry, and the
length of the name contained in the entry. These entries
are followed by the name padded to a 4 byte boundary with
null bytes. All names are guaranteed null-terminated. The
maximum length of a name in a directory is MAXNAMLEN.

#define DIRBLKSIZE DEV_BSIZE
#define MAXNAMLEN 256

struct direct {
    u_long d_ino;       /* inode number of entry */
    u_short d_reclen;   /* length of this record */
    u_short d_namlen;   /* length of string in d_name */
    char d_name[MAXNAMLEN + 1];
    /* typically shorter */
};

SEE ALSO
fs_ufs(4)
The command `du` (disc usage) was written to tell how much disc space is consumed by the files in a directory, including all its subdirectories.

```
$ du
6   ./recipes/pie
4   ./recipes/cookie
11  ./recipes
13  
$  
```

The filenames are obvious; the numbers are the number of disc blocks — typically 512 or 1024 bytes each — of storage for each file. The value for a directory indicates how many blocks are consumed by all the files in that directory and its subdirectories, including the directory itself.

`du` has an option `-a`, for “all,” that causes it to print out all the files in a directory. If one of those is a directory, `du` processes that as well:

```
$ du -a
2   ./recipes/pie/apple
3   ./recipes/pie/crust
6   ./recipes/pie
3   ./recipes/cookie/choc.chip
4   ./recipes/cookie
11  ./recipes
13  ./junk
13  
$  
```

The output of `du -a` can be piped through `grep` to look for specific files:

```
$ du -a | grep choc
3   ./recipes/cookie/choc.chip
$  
```

Recall from Chapter 1 that the name `.` is a directory entry that refers to the directory itself; it permits access to a directory without having to know the full
18544 -rw------- 1 goelman 22 Jun 8 1994 eqfile
19299 -rw------- 1 goelman 17 Feb 23 1993 file2
18809 -rw------- 1 goelman 17 Feb 2 1993 file3
18521 -rw------- 1 goelman 10 Jan 11 1994 file4
25389 drwx------- 2 goelman 512 Feb 21 1994 oddir
NAME
dirent - file system independent directory entry

SYNOPSIS
#include <dirent.h>

DESCRIPTION
Different file system types may have different directory entries. The dirent structure defines a file system independent directory entry, which contains information common to directory entries in different file system types. A set of these structures is returned by the getdents(2) system call.

The dirent structure is defined:

```c
struct dirent {
    ino_t     d_ino;
    off_t     d_off;
    unsigned short d_reclen;
    char      d_name[1];
};
```

The d_ino is a number which is unique for each file in the file system. The d_off entry contains a value which is interpretable only by the filesystem that generated it. It may be supplied as an offset to lseek(2) to find the entry following the current one in a directory. The field d_name is the beginning of the character array giving the name of the directory entry. This name is null terminated and may have at most MAXNAMLEN characters. This results in file system independent directory entries being variable length entities. The value of d_reclen is the record length of this entry. This length is defined to be the number of bytes between the current entry and the next one, so that the next structure will be suitably aligned.

SEE ALSO
getdents(2), lseek(2)

unOS 5.7 Last change: 6 Jan 1998
2.3 Directories and filenames

- current/working directory
- pwd, du
- . (dot) and .. (dot-dot)
- closer look: $16\text{bytes} = 2(\text{inode}) + 14(\text{name})$ NOT!
- directory = ordinary file (not for Create/Write)

2.4 Permissions

- super-user
- crypt command
- owner/group/others
- permissions: read/write/execute
- set-uid progrm (ex. passwd)

- permissions for a directory
  - $r$: can read direct.
  - $w$: create/delete files in direct.
$ grep you /etc/passwd
you: gkmbCTRj04COM:604:1:Y.O.A.People:/usr/you:
$

The fields in the password file are separated by colons and are laid out like this (as seen in passwd(5)):

```
```

The file is ordinary text, but the field definitions and separator are a convention agreed upon by the programs that use the information in the file.

The shell field is often empty, implying that you use the default shell, /bin/sh. The miscellany field may contain anything; often, it has your name and address or phone number.

$ ls -l /etc/passwd
-rw-r--r-- 1 root 515 Aug 30 10:40 /etc/passwd
$ ls -l /etc/passwd
-rw-r--r-- 1 root 515 Aug 30 10:40 /etc/passwd
$

$ ls -l /bin/passwd
-rw-r--r-- 1 root 8454 Jan 4 1983 /bin/passwd
$
x : can search direct.

- $ ls -l filename
- $ chmod -w filename
- $ chmod 777 filename.

2.5 Inodes

- file : name, data, admin info
- inode: permissions, time
- time: last modified, last used, last inode changed
- closer look to rm, ln, cp

2.6 The directory hierarchy

2.7 Devices
The i-node contains all the information about a file except for its name, which is kept in a directory. An i-node is 64 bytes long, so there are 8 i-nodes to a physical block. There is no set number of blocks occupied by the i-node list; it depends on how many i-nodes are specified at the time the file system is created. An i-node contains:

- the type and mode of file: type is regular (-), directory (d), block (b), character (c), or FIFO, also known as named pipe, (p); mode is the set of read-write-execute permissions
- the number of links to the file
- the owner's user-id number
- the group-id number to which the file belongs
- the number of bytes in the file
- an array of 13 disk block addresses
- the date and time last accessed
- the date and time last modified
- the date and time created
This comes from the file /usr/include/sys/fs/ufs_inode.h:

```
struct icommon {
    o_mode_t ic_smode; /* 0: mode and type of file */
    short ic_nlink; /* 2: number of links to file */
    o_uid_t ic_suid; /* 4: owner's user id */
    o_gid_t ic_sgid; /* 6: owner's group id */
    u_offset_t ic_lsize; /* 8: number of bytes in file */
#ifdef _KERNEL
    struct timeval32 ic_atime; /* 16: time last accessed */
    struct timeval32 ic_mtime; /* 24: time last modified */
    struct timeval32 ic_ctime; /* 32: last time inode changed */
#else
    time32_t ic_atime; /* 16: time last accessed */
    int32_t ic_atspare;
    time32_t ic_mtime; /* 24: time last modified */
    int32_t ic_mtspare;
    time32_t ic_ctime; /* 32: last time inode changed */
    int32_t ic_ctspare;
#endif
    daddr32_t ic_dib[NDBADDR]; /* 40: disk block addresses */
    daddr32_t ic_ib[NIADDR]; /* 88: indirect blocks */
    int32_t ic_flags; /* 100: status, currently unused */
    int32_t ic_blocks; /* 104: blocks actually held */
    int32_t ic_gen; /* 108: generation number */
    int32_t ic_shadow; /* 112: shadow inode */
    uid_t ic_uid; /* 116: long EFT version of uid */
    gid_t ic_gid; /* 120: long EFT version of gid */
    uint32_t ic_oeflflag; /* 124: reserved */
};
```
Table 2.1: Interesting Directories (see also hier(7))

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>root of the file system</td>
</tr>
<tr>
<td>/bin</td>
<td>essential programs in executable form (&quot;binaries&quot;)</td>
</tr>
<tr>
<td>/dev</td>
<td>device files</td>
</tr>
<tr>
<td>/etc</td>
<td>system miscellany</td>
</tr>
<tr>
<td>/etc/motd</td>
<td>login message of the day</td>
</tr>
<tr>
<td>/etc/passwd</td>
<td>password file</td>
</tr>
<tr>
<td>/lib</td>
<td>essential libraries, etc.</td>
</tr>
<tr>
<td>/tmp</td>
<td>temporary files; cleaned when system is restarted</td>
</tr>
<tr>
<td>/unix</td>
<td>executable form of the operating system</td>
</tr>
<tr>
<td>/usr</td>
<td>user file system</td>
</tr>
<tr>
<td>/usr/adm</td>
<td>system administration: accounting info., etc.</td>
</tr>
<tr>
<td>/usr/bin</td>
<td>user binaries: troff, etc.</td>
</tr>
<tr>
<td>/usr/dict</td>
<td>dictionary (words) and support for spell(1)</td>
</tr>
<tr>
<td>/usr/games</td>
<td>game programs</td>
</tr>
<tr>
<td>/usr/include</td>
<td>header files for C programs, e.g. math.h</td>
</tr>
<tr>
<td>/usr/include/sys</td>
<td>system header files for C programs, e.g. inode.h</td>
</tr>
<tr>
<td>/usr/lib</td>
<td>libraries for C, FORTRAN, etc.</td>
</tr>
<tr>
<td>/usr/man</td>
<td>on-line manual</td>
</tr>
<tr>
<td>/usr/man/man1</td>
<td>manual pages for section 1 of manual</td>
</tr>
<tr>
<td>/usr/dec</td>
<td>hardware diagnostics, bootstrap programs, etc.</td>
</tr>
<tr>
<td>/usr/news</td>
<td>community service messages</td>
</tr>
<tr>
<td>/usr/pub</td>
<td>public oddments: see ascii(7) and eqnchar(7)</td>
</tr>
<tr>
<td>/usr/src</td>
<td>source code for utilities and libraries</td>
</tr>
<tr>
<td>/usr/src/cmd</td>
<td>source for commands in /bin and /usr/bin</td>
</tr>
<tr>
<td>/usr/src/lib</td>
<td>source code for subroutine libraries</td>
</tr>
<tr>
<td>/usr/spool</td>
<td>working directories for communications programs</td>
</tr>
<tr>
<td>/usr/spool/lpd</td>
<td>line printer temporary directory</td>
</tr>
<tr>
<td>/usr/spool/mail</td>
<td>mail in-boxes</td>
</tr>
<tr>
<td>/usr/spool/uucp</td>
<td>working directory for the uucp programs</td>
</tr>
<tr>
<td>/usr/sys</td>
<td>source for the operating system kernel</td>
</tr>
<tr>
<td>/usr/tmp</td>
<td>alternate temporary directory (little used)</td>
</tr>
<tr>
<td>/usr/you</td>
<td>your login directory</td>
</tr>
<tr>
<td>/usr/you/bin</td>
<td>your personal programs</td>
</tr>
</tbody>
</table>
```bash
$ ls -l /dev
crw--w--w- 1 root 0, 0 Sep 27 23:09 console
crw--r--r-- 1 root 3, 1 Sep 27 14:37 kmem
crw--r--r-- 1 root 3, 0 May 6 1981 mem
brw-rw-rw- 1 root 1, 64 Aug 24 17:41 mt0
crw-rw-rw- 1 root 3, 2 Sep 28 02:03 null
crw-rw-rw- 1 root 4, 64 Sep 9 15:42 rmt0
brw-r----- 1 root 2, 0 Sep 8 08:07 rp00
brw-r----- 1 root 2, 1 Sep 27 23:09 rp01
crw-r----- 1 root 13, 0 Apr 12 1983 rrp00
crw-r----- 1 root 13, 1 Jul 28 15:18 rrp01
crw-rw-rw- 1 root 2, 0 Jul 5 08:04 tty
crw--w--w- 1 you 1, 0 Sep 28 02:38 tty0
crw--w--w- 1 root 1, 1 Sep 27 23:09 tty1
crw--w--w- 1 root 1, 2 Sep 27 17:33 tty2
crw--w--w- 1 root 1, 3 Sep 27 18:48 tty3
```
## DEVICE DESIGNATIONS

<table>
<thead>
<tr>
<th>Access</th>
<th>User</th>
<th>Group</th>
<th>Name</th>
<th>Date</th>
<th>Time</th>
<th>File</th>
</tr>
</thead>
<tbody>
<tr>
<td>crw-w-w-</td>
<td>emily</td>
<td>other</td>
<td>1</td>
<td>0 Nov</td>
<td>08:39</td>
<td>/dev/tty11</td>
</tr>
<tr>
<td>crw-w-w-</td>
<td>root</td>
<td>nuucp</td>
<td>1</td>
<td>1 Sep</td>
<td>13:06</td>
<td>/dev/tty12</td>
</tr>
<tr>
<td>crw-w-w-</td>
<td>dick</td>
<td>other</td>
<td>1</td>
<td>2 Nov</td>
<td>08:12</td>
<td>/dev/tty13</td>
</tr>
<tr>
<td>crw-w-w-</td>
<td>uucp</td>
<td>other</td>
<td>1</td>
<td>3 Nov</td>
<td>08:22</td>
<td>/dev/tty14</td>
</tr>
<tr>
<td>crw-rw-rw-</td>
<td>root</td>
<td>root</td>
<td>1</td>
<td>4 Jul</td>
<td>08:53</td>
<td>/dev/tty15</td>
</tr>
<tr>
<td>crw-w-w-</td>
<td>root</td>
<td>nuucp</td>
<td>2</td>
<td>0 Nov</td>
<td>07:54</td>
<td>/dev/tty21</td>
</tr>
<tr>
<td>crw-rw-rw-</td>
<td>root</td>
<td>root</td>
<td>2</td>
<td>1 Jul</td>
<td>08:53</td>
<td>/dev/tty22</td>
</tr>
</tbody>
</table>

### TTY
/ on /dev/dsk/c0t0d0s0 read/write/setuid on Wed Nov 26 14:45:35
/usr on /dev/dsk/c0t0d0s6 read/write/setuid on Wed Nov 26 14:45
/proc on /proc read/write/setuid on Wed Nov 26 14:45:35 1997
/dev/fd on fd read/write/setuid on Wed Nov 26 14:45:35 1997
/var on /dev/dsk/c0t0d0s3 read/write/setuid on Wed Nov 26 14:45
/tmp on swap read/write on Wed Nov 26 14:45:38 1997
/mnt on /dev/dsk/c0t0d0s7 quota/setuid/read/write on Wed Nov 26
/mnt3 on /dev/dsk/c0t1d0s4 setuid/read/write on Wed Nov 26 14:4
/opt on /dev/dsk/c0t0d0s5 setuid/read/write on Wed Nov 26 14:45
/usr/openwin on /dev/dsk/c0t0d0s4 setuid/read/write on Wed Nov
/usr4 on /dev/dsk/c0t1d0s0 setuid/read/write on Wed Nov 26 14:4
/usr1 on /dev/dsk/c0t1d0s3 setuid/read/write on Wed Nov 26 14:4
/opt-gnu on /dev/dsk/c0t2d0s7 setuid/read/write on Wed Nov 26 1
/opt/local on /dev/dsk/c0t2d0s6 setuid/read/write on Wed Nov 26
/home on /dev/dsk/c0t3d0s1 setuid/read/write on Wed Nov 26 14:4
/usr3 on /dev/dsk/c0t3d0s7 setuid/read/write on Wed Nov 26 14:4
/mnt2 on /dev/dsk/c0t5d0s3 quota/setuid/read/write on Wed Nov 2
/usr2 on /dev/dsk/c0t5d0s4 setuid/read/write on Wed Nov 26 14:4
/csc on /dev/dsk/c0t5d0s6 quota/setuid/read/write on Wed Nov 26
/cdrom/solaris_2_5_1_desktop_1_1 on /vol/dev/dsk/c0t6d0/solaris
/etc/name_to_major

le 40
cfgfour 86
cgeight-p4 87
cgeight 80
xdc 88
xyc 89
kstate 90
vol 91
xbox 92
bootbus 93
sbi 94
mcp 95
mcpsa 96
mcpp 97
sx 98
cgfourteen 99
sx_cmem 100
stc 101
isp 102
gec 103
qge 104
tl 105
be 106
llcl 107
audio 108
pci 111
tnf 114
pci_pci 115
soc 116
pln 117
sed 118
central 119
fhc 120
environ 121
sysctl 122
ac 123
sram 124
simstat 125
ebus 126

cn 0
rootnex 1
pseudo 2
ip 3
logindmux 4
icmp 5
fas 6
hme 7
sp 10
clone 11
sad 12
um 13
iwc 14
wc 15
conskbd 16
consms 17
dump 19
se 20
log 21
sy 22
ptm 23
pts 24
ptc 25
ptal 26
bwtwo 27
audio 28
ze 29
cgtprime 30
cgtwo 31
sd 32
st 33
xy 34
xd 35
fd 36
su 37
openpence 38
cgsix 39
REVIEW OF /dev DIRECTORY FILES.

The purpose of the typical /dev directory files are shown in the chart below.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>acw</td>
<td>Networking</td>
</tr>
<tr>
<td>boot</td>
<td>Boot driver</td>
</tr>
<tr>
<td>console, syscon</td>
<td>Console terminal</td>
</tr>
<tr>
<td>diskette</td>
<td>Floppy disk</td>
</tr>
<tr>
<td>disk, rdisk</td>
<td>Hard disk</td>
</tr>
<tr>
<td>kmem, mem</td>
<td>Core UNIX</td>
</tr>
<tr>
<td>lp</td>
<td>Printers</td>
</tr>
<tr>
<td>ml, mml</td>
<td>Cartridge or 9-track tape</td>
</tr>
<tr>
<td>null</td>
<td>Bit bucket</td>
</tr>
<tr>
<td>swap</td>
<td>Swap driver</td>
</tr>
<tr>
<td>sxt</td>
<td>Layering</td>
</tr>
<tr>
<td>tty</td>
<td>Terminal ports</td>
</tr>
<tr>
<td>User</td>
<td>Mode</td>
</tr>
<tr>
<td>-------</td>
<td>----------------</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>teresa</td>
<td>crw-</td>
</tr>
<tr>
<td></td>
<td>r</td>
</tr>
<tr>
<td></td>
<td>r</td>
</tr>
<tr>
<td></td>
<td>r</td>
</tr>
<tr>
<td></td>
<td>r</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>root</td>
<td>crw-rw-rw</td>
</tr>
<tr>
<td>Miscellaneous Information</td>
<td></td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>- logical/physical disk sizes</td>
<td></td>
</tr>
<tr>
<td>- superblock modified flag</td>
<td></td>
</tr>
<tr>
<td>- read-only system flag</td>
<td></td>
</tr>
<tr>
<td>- current date of last update</td>
<td></td>
</tr>
<tr>
<td>- label or name</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>I-node Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>- total number of i-nodes</td>
</tr>
<tr>
<td>- total free i-nodes</td>
</tr>
<tr>
<td>- array of free i-node numbers</td>
</tr>
<tr>
<td>- index into inumber array</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Block Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>- total number of free blocks</td>
</tr>
<tr>
<td>- array of free block numbers</td>
</tr>
<tr>
<td>- index into array of free block numbers</td>
</tr>
</tbody>
</table>

Figure 5-4: The Super-Block