1 OPERATING SYSTEMS LABORATORY XIII UNIX FILE MANAGEMENT

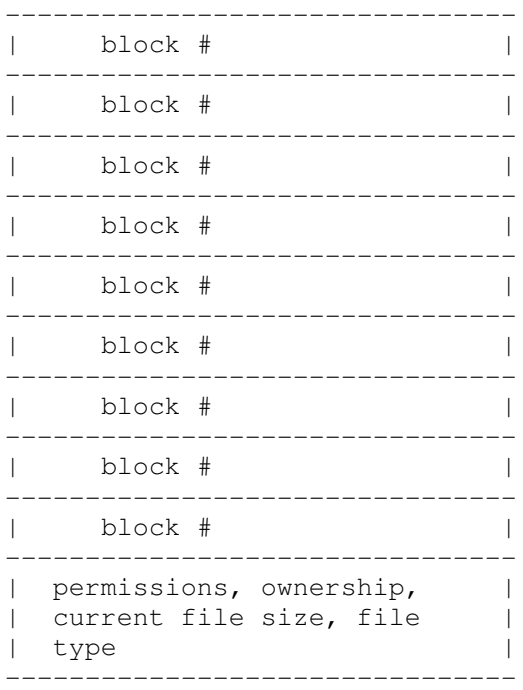
- In UNIX, each file is represented by an inode

- The inode

-- maps individual byte addresses relative to the beginning of the file to logical block numbers for a particular disk

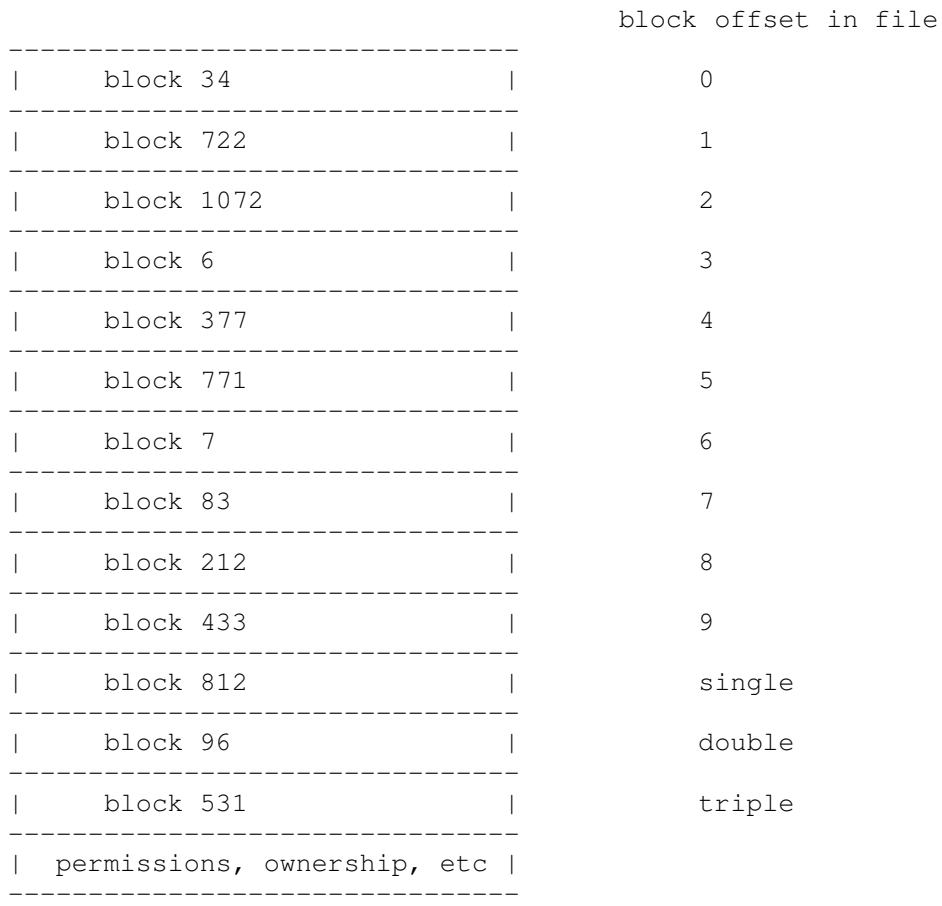
-- holds permission information for the file, whether the file is a regular file, a directory, a special file, and the current file size

- An example inode



- blocks are a single, fixed size,
- table index corresponds to logical position in the file

- For example,



defines a file. If blocks are 512 bytes long, the 1033rd byte in the file is found by first calculating the logical offset (in blocks) from the beginning of the file as:

1033 / 512 = 2

At table entry 2 (block offset 2) we find logical disk block number 1072. The byte offset within disk block 1072 is:

1033 % 512 = 9

So given the inode shown above, the 1033rd byte of the file is byte 9 of block 1072 on the disk from whence this inode is allocated.

-- latter table slots (single, double, triple in the figure) refer to single-, double-, and triple-levels of indirection.

-- the number of direct, single, double, and triple slots in each inode are implementation specific. However, given these numbers, the block size, and the size of a block number, it is possible to calculate how large the largest file that can be represented is.

For example, assume

block number = 4 bytes
block size = 512 bytes
direct blocks = 10
single indirect blocks =1
double indirect blocks =1
triple indirect blocks =1

implies

(10) directly accessible blocks +
(512 / 4) single indirectly addressable blocks +
(512 / 4)^2 double indirectly addressable blocks +
(512 / 4)^3 triple indirectly addressable blocks =

2113674 addressable blocks * 512 bytes/block =

1,082,201,088 addressable bytes within a single file

- $ ls –i (execute this command inside your home directory)

- Compile and run the following program
- Analyze the output

```
/* structuremembers.c
Print structure members of files
st_mode the type and mode of the file
st_ino
st_dev
st_rdev
st_nlink
st_uid
```

```
st_gid
st_size
st_atime
st_mtime
st_ctime
*/
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>

int main()
{
        struct stat status;
        int i;
        for (i=1 ; i < argc ; i++)
        {
                if(stat(argv[i], &status))
                        fprintf(stderr, "Cannot stat %s \n", argv[i]);
                else
                        printf("%15s %4.4o\n", argv[i], status.st_mode & 07777);
                        //printf("%15s %14d\n", argv[i], status.st_ino);

        return 0;
}
```