Implementation lecture 3

Disks continued

What are they used for?

Structure of file systems - ch 11

What do we use our disks for?

At least 3 more or less separate things

- File storage
- Virtual memory - 12.4
- Spooling space (along with virtual memory could be stored in standard files)
- System areas - e.g. bootstrap code, information about the drive - size, name, partition maps, super blocks, inodes, blocks in use, blocks free

There must be some conventions e.g. the OS knows where to find information about the disk from the disk.

e.g. Track 0

Why is it usually not a good idea to implement virtual memory in an ordinary file?

Virtual memory needs to be as fast as possible.

The extra overhead in dealing with ordinary files.

contiguous allocation is faster than scattered allocation

can then use simpler operations than normal system reads and writes

Some companies claim no performance degradation using swap files.

- but the file is not accessed using the standard read and write calls
- routines optimised to deal with virtual memory chunks

Similar arguments can be made for treating spooling areas as special

- they are always used sequentially
- they are transitory

<u>How do we find things?</u>

Almost always the file structure is a tree.

Nodes correspond to directories

- except the first - which may be the volume identifier
- and the last - which is probably an ordinary file.

Is it sensible to store directories as files?

Advantages

- Expandability - as long as you can allocate a new file, you can allocate a new directory
- Expandability - easy to allow unlimited numbers of files in the directory. The directory grows like an ordinary file.
- Uniform. No special treatment for directories.

Disadvantages

Directories occur all over the disk.

Slower to find things. Long pathnames lead to lots of seeking and opening.

If no special treatment is needed for directories why does UNIX require superuser privileges to make a directory? - the mkdir command is suid to superuser.

It is a good idea to keep information about each file with the file.

Helps recovery if the directories are damaged.

How else could it be done?

Store all directories together in a special directory area.

Fast access to directory information (no long seeks)

What about loading the entire directory into memory?

Where are the blocks for each file?

Don't usually have contiguous allocation of files.

Advantages of speed (especially a reduced number of file system accesses).

But a lack of flexibility. Works best with more than enough disk space.

Chaining

Directory points to the first block of the file

Each block points to the next one

Free list of blocks maintained on the disk

Bad for random access

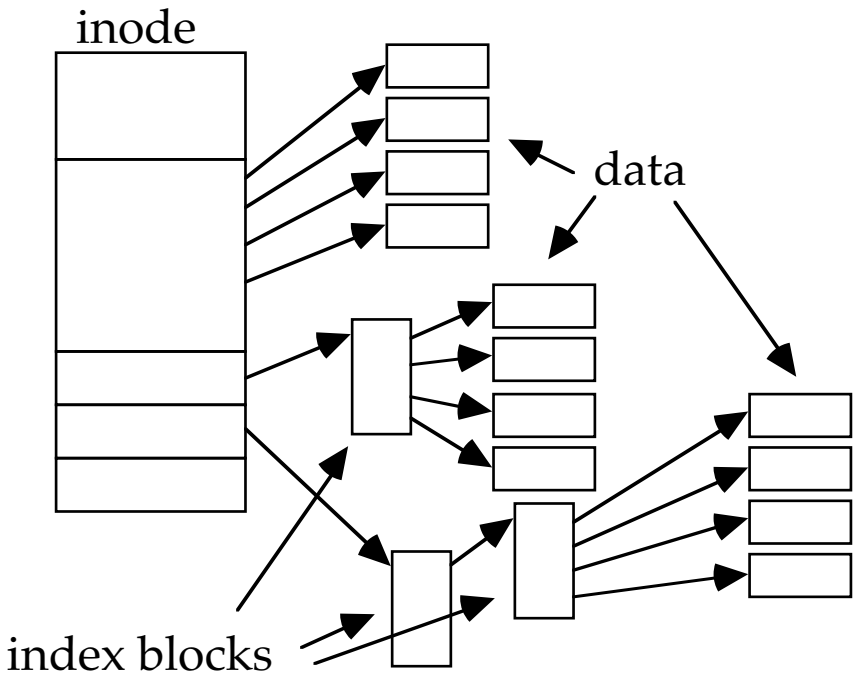
Can lose large chunks of a file with one bad block

Doubly-linked circular list helps

List of blocks (or chunks) in the directory entries

Can grow very large

Extra levels of indirection may be required



Free list of blocks

or bitmap - array with used blocks shown

MS-DOS FAT

Links, but all in one table.

Directory holds the number of the first block.

File blocks linked together.

How many reads do we need to do to read a particular byte?

(contiguous allocation wins this one)

Different names for the same file

Either maintain a copy of the file data

or a copy of the directory entry data.

Each copy needs to know about the other copies.

Provide a level of indirection.

Directory entries don't directly hold information about files.

They point to something which does - UNIX inodes

Maintain the number of links.

UNIX soft links

Just text files with the real name stored there.

File is flagged as a link.

Observed behaviour of Win95 shortcuts

Shortcuts made from shortcuts point to the original data.

Moving or renaming the original data updates all the shortcuts.

Deleting the original data leads to file not found.

Explain what the underlying mechanism must be.

What about Mac aliases?

Partitioning disks

Sometimes we want one disk to look like several.

Why?

Because the lousy file system demands it

- limitations on number of files
- on size of disk drives
- forces too much space wasted due to number of logical blocks or clusters

Good reasons

- some disk management can be easier with smaller drives
- safer - a bad partition shouldn't affect files on the other partitions
- can maintain and check one partition while others are still in use
- back up entire partitions e.g. onto a Zip or Syquest drive
- want different file systems on the same device

Need

Some sort of record which specifies which parts of the device correspond to which logical disk.