## old file system

drawbacks(low throughput):

* long seek from inode to data, although the data might be consecutive, the inodes might not be consecutive, which leads to large time consuming.
* small block size, limited read-ahead.
* after adding and removing from files, the files become randomly split on the disk.

## new file system

disk partition -> one or more cylinder groups -> consecutive cylinder on a disk

\tip:

global allocator - to make files in a directory in a cylinder group

local allocator - try to find the optimal block in a cylinder for one file.

\tip:’

the benefits for delayed writing:

* schedule the writing (take the example of elevator)
* integrate writing to one same file

each cylinder group -> bookkeeping information, superblock, space for inodes, a bit map for available blocks, summary information for usage in the cylinder group.

the bookkeeping information is located sparsely to avoid hardware crashing influence.

(I think) the cylinder group is the file system, which contain s a superblock

superblock -> contains rotational layout tables to help find out the optimal next block.

problem: large block size -> more waste on the space.

solution: block is further divided into fragments, which has a low bound of the sector size.

=> the result is that the wast space is about the same with the old system with the fragment size.

space allocation -> when there is a write system call.

paramaterize in the new file system:

parameters: speed of the processor, hardware support, characteristics of the mass storage devices(number of blocks per track and the disk spin rate)

Using the parameters -> calculate the rotationally optimal next block.

this kind of paramaterization could be done after the disk is relocated to a new processor with new parameters.

the layout: balance between 1) localizing data->too to create the old file system 2) spreading out unrelated data

Allocation of inode:

* For files

The files in the same directory are supposed to be in the same cylinder group.

* For directories

Allocate to the cylinder groups which contain above-average free blocks.

Allocation for data blocks:

Should be allocated to the same cylinder group, unless it reaches to a maximum usage of the cylinder group.

## file systems functional enhancement

* long file names support
* advisory lock(shared or exclusive)
* symbolic links
* rename
* quota with soft limits and hard limits

\tip:

journaling file system(JFS):

when write to a fileA, temporarily write to some placeB, and tell A it has an update at B; and then writes back to A at some time.