ReiserFS

Nodes of a BTree each cover many consecutive blocks

Files' contents are only stored in leaf nodes

Search keys (only D,F,O have to be used everywhere)

D = directory number – unique id for dir this file is in,

F = file number – unique id for this file

O = fragment offset

S = fragment size

L = total length of whole file

Many small files share a single leaf node

Big files are fragmented and spread over many nodes.

e.g. if node can contain up to 10,000 bytes of file content

and we have a 27,300 byte file

Maybe first 3,100 bytes fit at the end of one node

next 10,000 bytes (O = 3,100) fill the next node

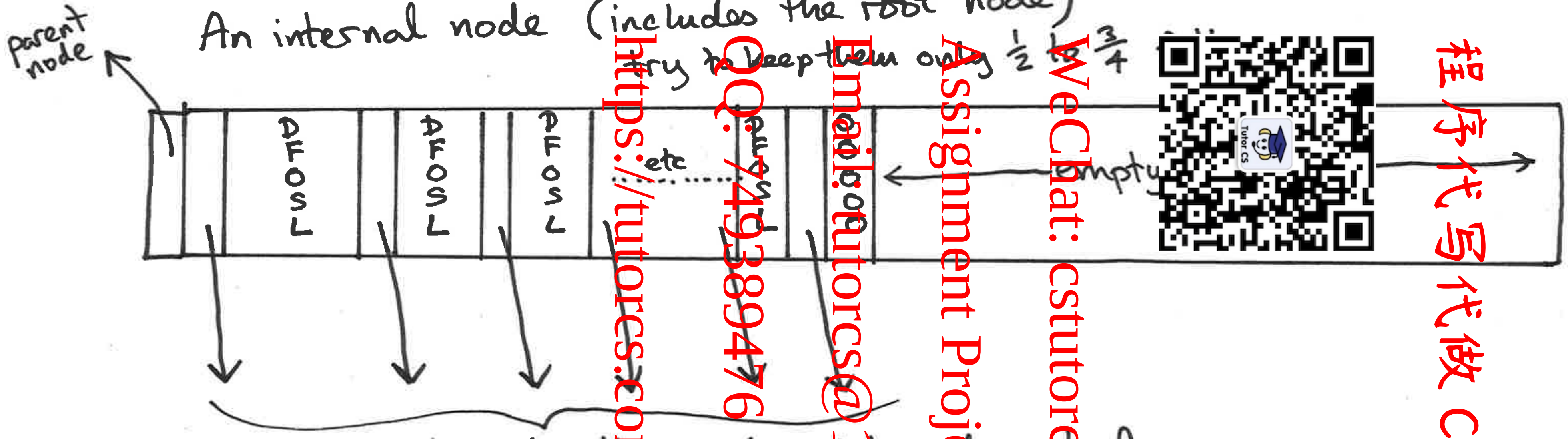
next 10,000 bytes (O = 13,100) fill the next node

remaining 4,200 bytes (O = 23,100) share the next node
with the next file

Search keys are (D, F, O)

Tree is ordered primarily on increasing D,
secondarily on increasing F,
thirdly on increasing O.

An internal node (includes the root node)

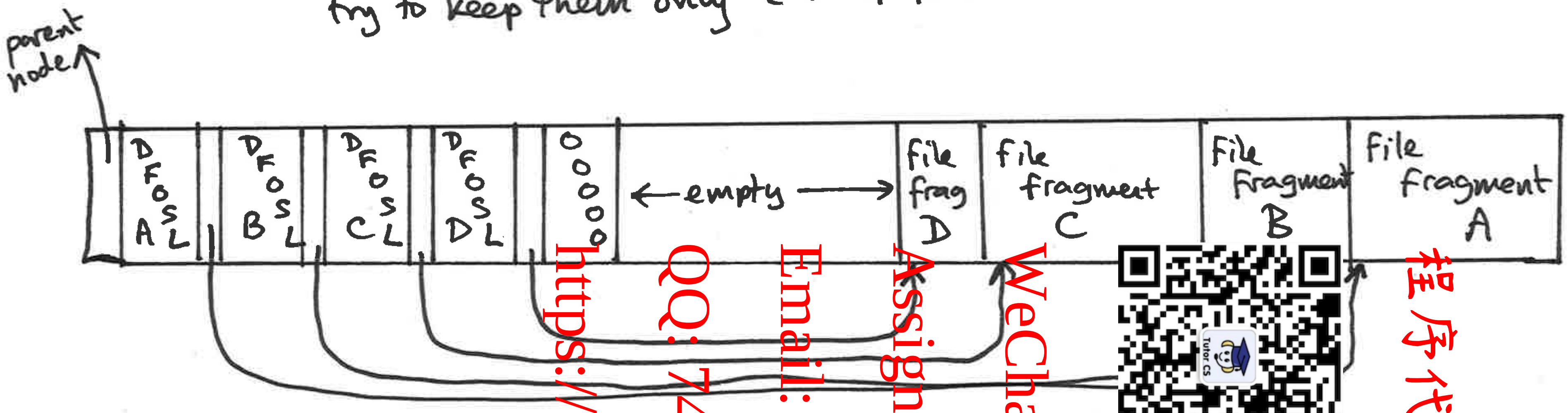


pointers to other nodes, internal or leaf.
remember its all on disk, nodes are just big clusters,
so these pointers are just cluster (block) numbers.

parent pointers are not really necessary.

A leaf node

try to keep them only ½ to ¾ full



these pointers are just positions within th

fragments of the same file that are in the same node are always merged into a single fragment.

so we know that in this picture

A B C & D are fragments of different files, and

B & C are entire files

All leaf nodes automatically have the same depth in the tree.

Think of the 2-3-tree operations, this is just an extension.