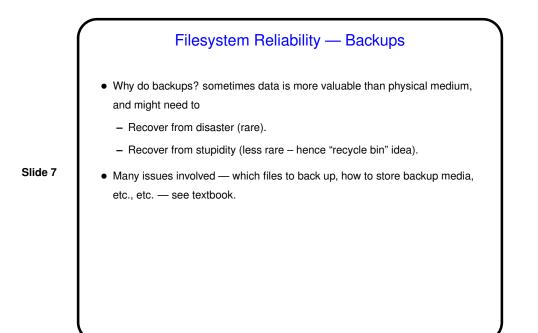
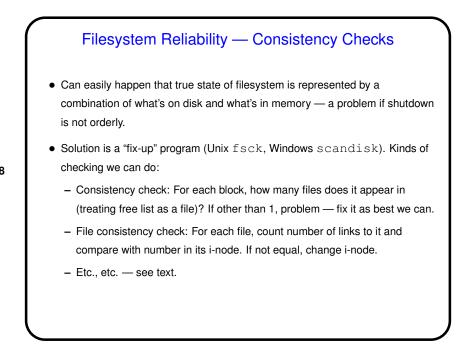


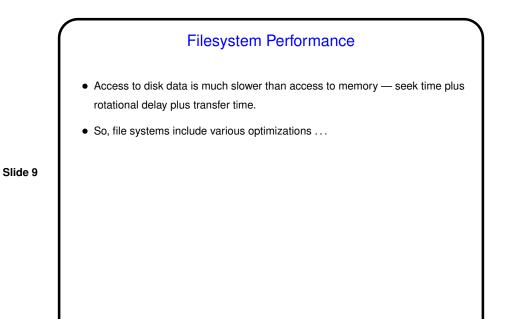
Managing Free Space — Bitmap
Another way to track which blocks are free — "bitmap" with one bit for each block on disk, also kept on disk.
How this works:

Keep one block of map in memory.
Modify entries as for free list.

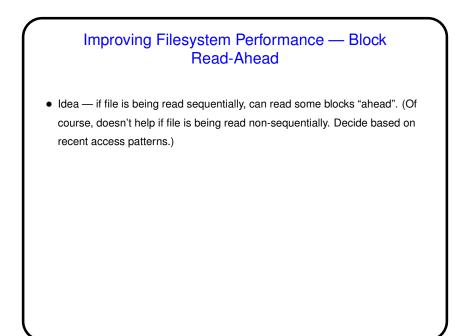
Usually requires less space.







Improving Filesystem Performance — Caching
Idea — keep some disk blocks in memory; keep track of which ones are there using hash table (base hash code on device and disk address).
When cache is full and we must load a new block, which one to replace? Could use algorithms based on page replacement algorithms, could even do LRU accurately — though that might be wrong (e.g., want to keep data blocks being filled).
When should blocks be written out?
If block is needed for file system consistency, could write out right away. If block hasn't been written out in a while, also could write out, to avoid data loss in long-running program.
Two approaches: "Write-through cache" (Windows) — always write out modified blocks right away. Periodic "sync" to write out (Unix).

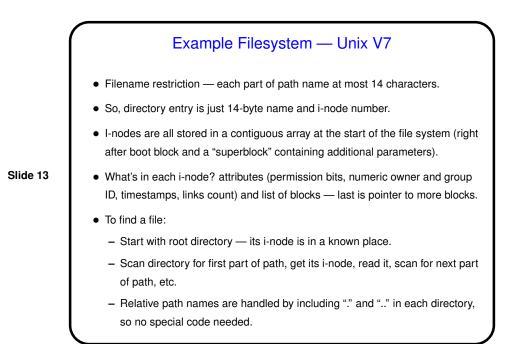


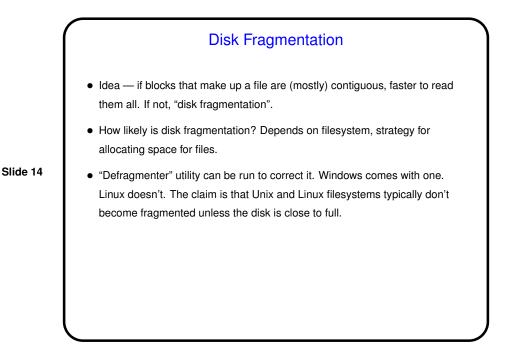
Improving Filesystem Performance — Reducing Disk Arm Motion

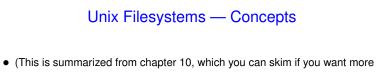
 Group blocks for each file together — easier if bitmap is used to keep track of free space. If not grouped together — "disk fragmentation" may affect performance.

Slide 12

• Place i-nodes so they're fast to get to (and so maybe we can read an i-node and associated file block together).







- Single type of file sequence of bytes. lseek allows random access.
- Single root directory. mount allows access to multiple physical devices.

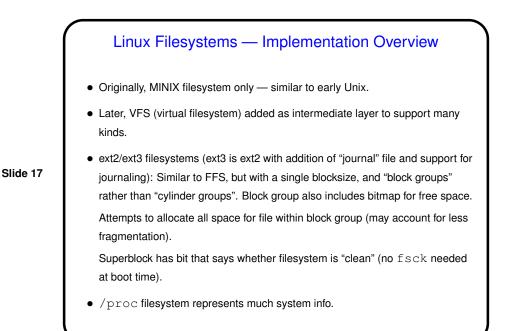
- Links, hard or symbolic, to allow non-tree directory structure.
- Locks to control access to files/records.
- File descriptors for open files.

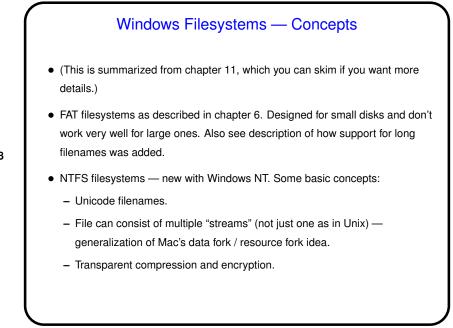
details.)

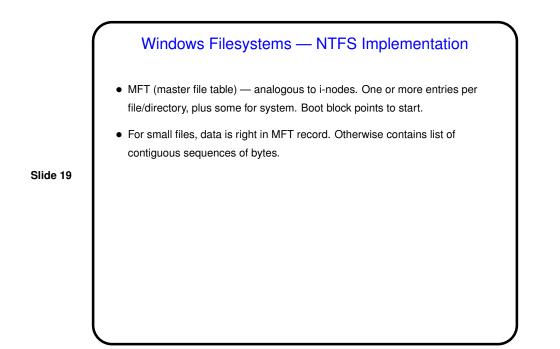
• "Pipes" — pseudofiles for connecting processes.



- Superblock contains critical info how many i-nodes, location of free list, how many blocks, etc.
- After that? In early implementations, all i-nodes followed by all data blocks.
 Later implementations (Berkeley FFS) use "cylinder groups", each containing superblock copy, i-nodes, and data blocks for better performance, reliability.
- Slide 16
- Directory entries fixed-size in early implementations, varying-size in later ones.
- I-nodes contain file attributes, link count, list of (some) blocks, pointers to indirect blocks.
- In memory table of i-nodes for open files, table of file descriptors (containing, e.g., info about position within file).
- NFS allows access to other systems' disks.







None — sign in.