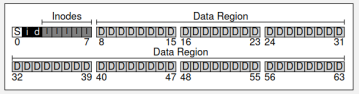
File System Concepts

- file systems are purely software
  - not supported by hardware features
- many different file systems exist
- key aspects of file systems
  - how the data and metadata is organized (*data structures*)
  - how those data structures are accessed to implement system calls (*access methods*)

A Simple File System – Organization

- the disk partition is viewed as a series of blocks
  - numbered 0 to N-1



- the space allocated for the *inode table* determines the maximum number of files the file system can hold
- *inode bitmap* (i) tracks usage of inodes
- *data bitmap* (d) tracks usage of data blocks
  - 1 bit per inode/block – must allocate enough space for the number of inodes and data blocks in the file system
- *superblock* (S) stores file system information
  - number of inodes, data blocks
  - location of the inode table
  - magic number
  - …

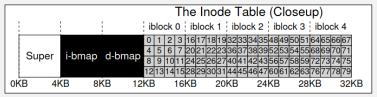
Inodes

- inodes store the metadata for files
  - type (file, directory, etc)
  - size
  - number of blocks
  - owner and access permissions
  - time created, modified, last accessed
  - location of data blocks

simplified ext2 inode
– ext2 replaced the first Linux filesystem (ext)
– now deprecated and replaced by ext4

Size	Name	What is this inode field for?
2	mode	can this file be read/written/executed?
2	uid	who owns this file?
4	size	how many bytes are in this file?
4	time	what time was this file last accessed?
4	ctime	what time was this file's inode changed?
4	mtime	what time was this file last modified?
4	dtime	what time was this inode deleted?
2	gid	which group does this file belong to?
2	links_count	how many hard links are there to this file?
4	blocks	how many blocks have been allocated to this file?
4	flags	how should ext2 use this inode?
4	osd1	an OS-dependent field
60	block	a set of disk pointers (15 total)
4	generation	file version (used by NFS)
4	file_acl	a new permissions model beyond mode bits
4	dir_acl	called access control lists

- in simple file systems, the inode number can be used to index into the inode table to find the file's inode



Inodes – Referencing Data Blocks

Options –

- *direct pointers*
  - inode stores a pointer to each data block
  - only supports very small files
- *multilevel index* classic Unix filesystem, also used by ext2/ext3
  - inode has a fixed number of direct pointers and one *indirect pointer* to a block containing pointers to data blocks
  - for larger files, inode also includes double and triple indirect pointers
  - imbalanced tree design reflects that most files are small, and having more levels than needed wastes space
- *extents* used by ext4, the current default Linux file system, and NTFS (Windows)
  - store a collection of pointers + lengths
  - less flexible but more compact
  - work well when files have only a few extents
- *linked list* used by FAT/FAT12/FAT16/FAT32 – MS-DOS (1981-2000), Windows 9x (1995-2000); still used on mobile devices, embedded systems, and for data exchange
  - inode stores only a pointer to the first data block of the file
  - augment with an in-memory table of next pointers indexed by the data block address to allow efficient random access