

Announcements

- Reading Chapter 11 (8th ed)

Free Space Management

- How do we find a disk block to allocate?
- Bit Vectors
 - array of bits (one per block) that indicates if a block is free
 - compact so can keep in memory
 - 100 GB disk, 4K blocks -> 6MB per disk (0.003%)
 - easy to find long runs of free blocks
- Linked lists
 - each disk block contains the pointer to the next free block
 - pointer to first free block is keep in a special location on disk
- Run length encoding (called counting in book)
 - pointer to first free block is keep in a special location on disk
 - each free block also includes a count of the number of consecutive blocks that are free

DOS Directories

- **Root directory**
 - immediately follows the FAT
- **Directory is a table of 32 byte entries**
 - 8 byte file name, 3 byte filename extension
 - size of file, data and time stamp, starting cluster number of the file, file attribute codes
 - Fixed size and capacity
- **Subdirectory**
 - This is just a file
 - Record of where the subdirectory is located is stored in the FAT

Implementing Directories

- **Linear List**

- array of names for files
- must search entire list to find or allocate a filename
- sorting can improve search performance, but adds complexity

- **Hash table**

- use hash function to find filenames in directory
- needs a good hash function
- need to resolve collisions
- must keep table small and expand on demand since many directories are mostly empty

Unix Directories

- Space for directories are allocated in units called *chunks*
 - Size of a chunk is chosen so that each allocation can be transferred to disk in a single operation
 - Chunks are broken into variable-length directory entries to allow filenames of arbitrary length
 - No directory entry can span more than one chunk
 - Directory entry contains
 - pointer to inode (file data-structure)
 - size of entry
 - length of filename contained in entry (up to 255)
 - remainder of entry is variable length - contains file name

inodes

- File index node
- Contains:
 - Pointers to blocks in a file (direct, single indirect, double indirect, triple indirect)
 - Type and access mode
 - File's owner
 - Number of references to file
 - Size of file
 - Number of physical blocks