Announcements

- Reading Chapters 11 (11.6) and 12 (12.1-12.7)
  - suggested problems: 12.6 and 12.8
- Project #4 is available from the Web page

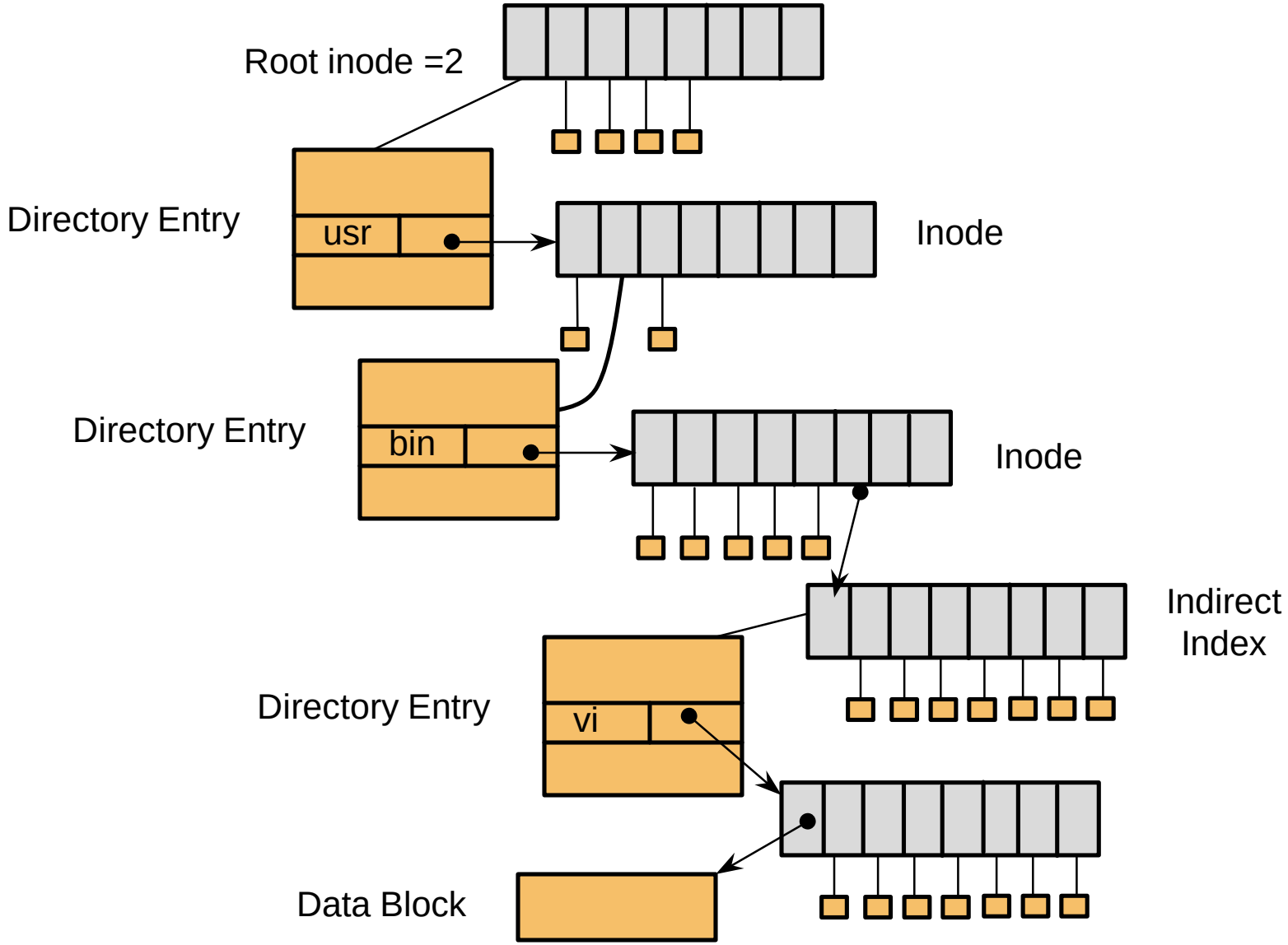
copyright 1996-8 Jeffrey K. Hollingsworth

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

copyright 1996-8 Jeffrey K. Hollingsworth

File Lookup (/usr/bin/vi)



copyright 1996-8 Jeffrey K. Hollingsworth

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

copyright 1996-8 Jeffrey K. Hollingsworth

Unix directories - links

- Each file has unique inode but it may have multiple directory entries in the same filesystem to reference inode
- Each directory entry creates a hard link of a filename to the file's inode
  - Number of links to file are kept in reference count variable in inode
  - If links are removed, file is deleted when number of links becomes zero
- Symbolic or soft link
  - Implemented as a file that contains a pathname
  - Symbolic links do not have an effect on inode reference count

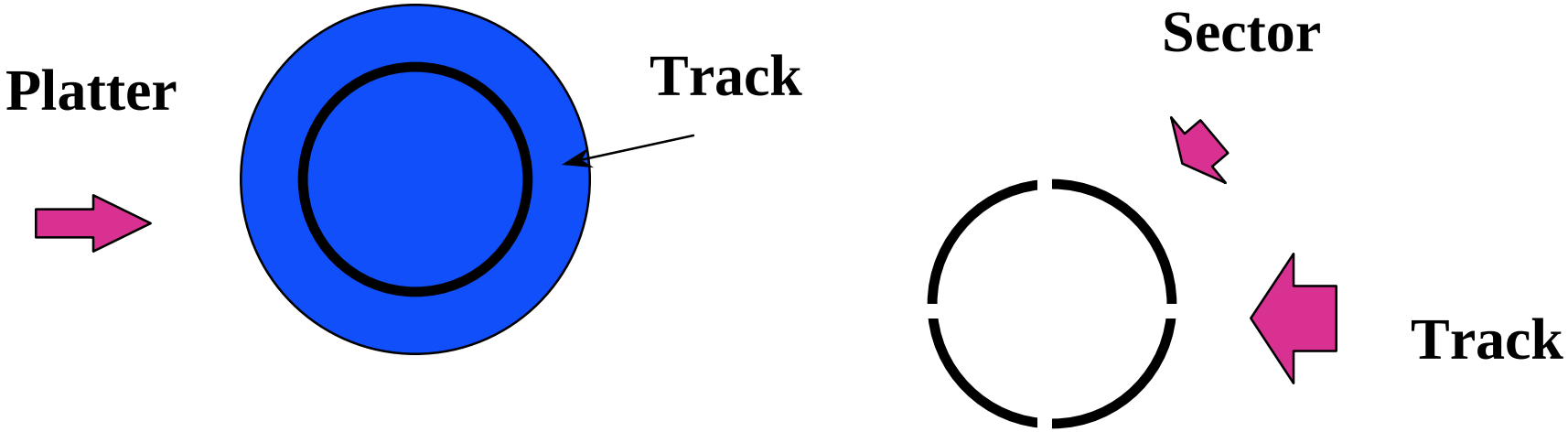
copyright 1996-8 Jeffrey K. Hollingsworth

Using UNIX filesystem data structures

- Example: find /usr/bin/vi
  - from Leffler, McKusick, Karels and Quarterman
  - Search root directory of filesystem to find /usr
    - root directory inode is, by convention, stored in inode #2
    - inode shows *where data blocks are* for root directory - *these blocks* (not the inode itself) *must* be retrieved and searched for entry user
    - we discover that the directory user's inode is inode #4
  - Search user for bin
    - access blocks pointed to by inode #4 and search contents of blocks for entry that gives us bin's inode
    - we discover that bin's inode is inode #7
  - Search bin for vi
    - access blocks pointed to by inode #7 and search contents of block for an entry that gives us vi's inode
    - we discover that vi's inode is inode #7
  - Access inode #7 - this is vi's inode

copyright 1996-8 Jeffrey K. Hollingsworth

Magnetic Disks



Collection of platters (1-20)
Rotate at 3600-7200 RPM
Size - usually 2.5-3.5 inch
Usually 500-2500 tracks per platter
Track consists of around 64 sectors
zones: vary number of tracks/sector based on distance from center

copyright 1996-8 Jeffrey K. Hollingsworth

Access Times

- Seek: Move disk arm over appropriate track
  - Seek times vary depending on locality - seek times are order of milliseconds
- Rotational delay: Wait time until desired information is under disk arm
  - A disk that rotates at 7200 RPM will take 8.3 ms to complete a full rotation
- Transfer time: time taken to transfer a block of bits (usually a sector)
  - Depends on recording density of track, rotation speed, block size
  - Achieved transfer rate for many blocks can also be influenced by other system bottlenecks (software, hardware)
  - Rates range from 2 to 8 MB per second

copyright 1996-8 Jeffrey K. Hollingsworth

Disk Scheduling

- First come, first served
  - ordering may lead to lots of disk head movement
  - i.e. 1, 190, 3, 170, 4, 160 etc.
  - total number of tracks traversed : 863
- Shortest seek time first: select request with the minimum seek time from current head position
  - move head to closest track
  - i.e. 1,3,4,160,190
  - total number of tracks traversed: 189
  - potential problem with distant tracks not getting service for an indefinite period

copyright 1996-8 Jeffrey K. Hollingsworth

Disk Scheduling

- Scan scheduling - read-write head starts at one end of the disk, moves to the other, servicing requests as it reaches each track
  - Consider example: 1, 190, 3, 170, 4, 160
  - If head starts at track 64 and moves towards 0, the ordering would be 4,3,1,160,170,190
  - Total distance 265
- C-Scan (circular scan)
  - disk head sweeps in only one direction
  - when the disk head reaches one end, it returns to the other
  - Consider example: 1, 190, 3, 170, 4, 160
  - If head starts at track 64 and moves towards 0, the ordering would be 4,3,1,190,170,160
  - Total distance 282

copyright 1996-8 Jeffrey K. Hollingsworth

Disk Cache

- Buffer in main memory for disk sectors
- Cache contains copy of some of the sectors on a disk. When I/O request is made for a sector, a check is made to find out if sector is in the disk cache
- Replacement strategy:
  - Least recently used: block that has been in the cache longest with no reference gets replaced
  - Least frequently used: block that experiences fewest references gets replaced

copyright 1996-8 Jeffrey K. Hollingsworth