

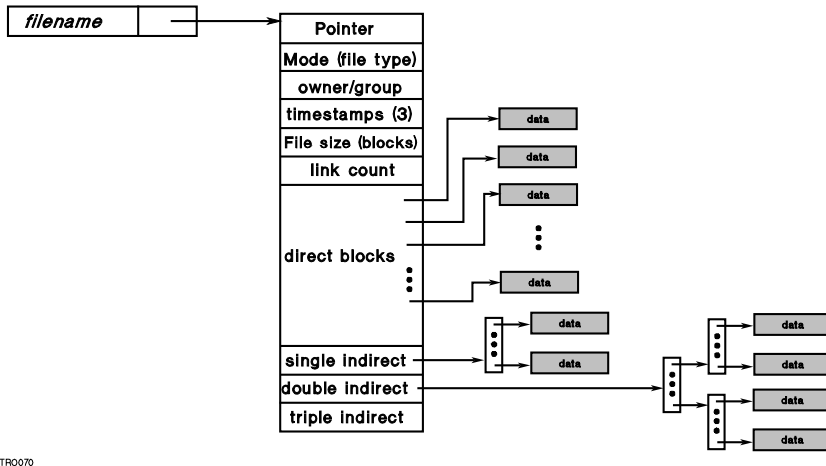
Disk Allocation Methods

- ▶ **Contiguous**
 - Simple and efficient, but
 - Not very flexible, but
 - Other formats strive for it
- ▶ **Linked List**
 - Great for sequential access, but
 - Not so good for random access.
 - File Allocation Table (FAT) – links in separate table
- ▶ **Indexed**
 - Good for both random and sequential access, but
 - Large files require lots of indices.
 - Unix uses indirect blocks

SLIDE001

UNIX i-node

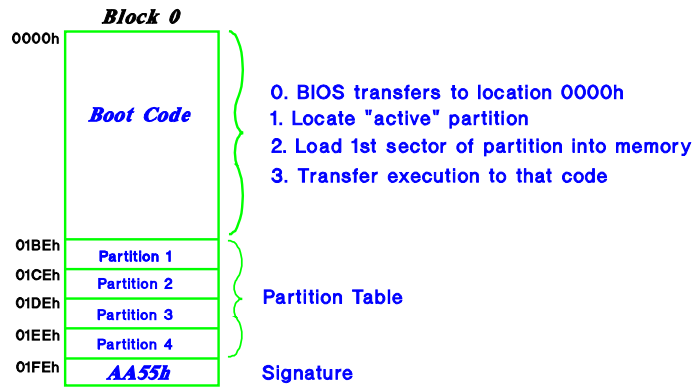
Directory Entry



INTR0070

Master Boot Record

Block zero on the disk

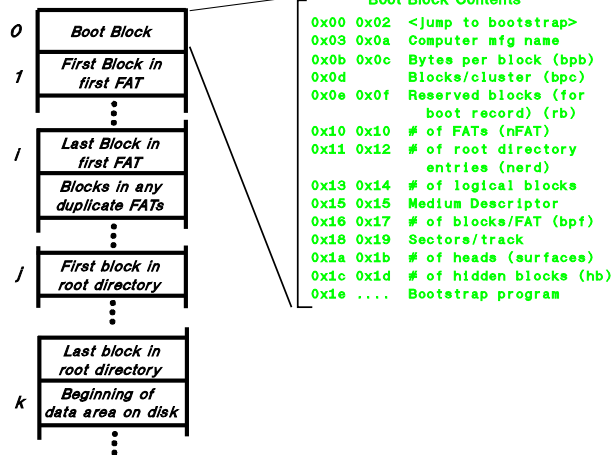


DISK0010

University of Idaho

DOS Partition

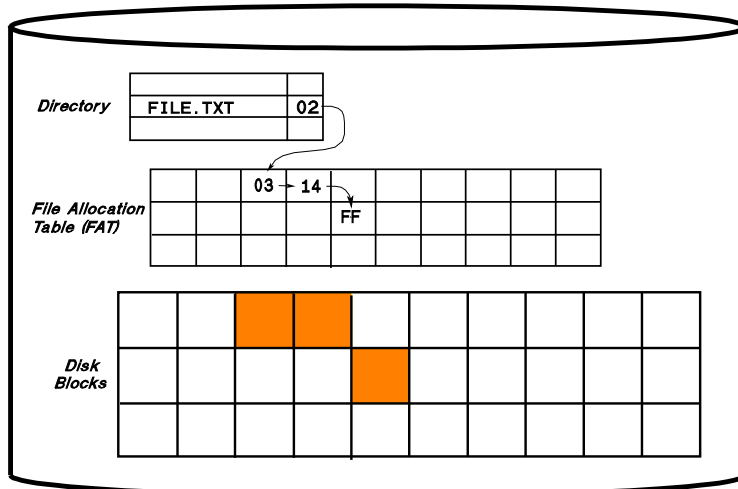
Disk Partition



DISK0020

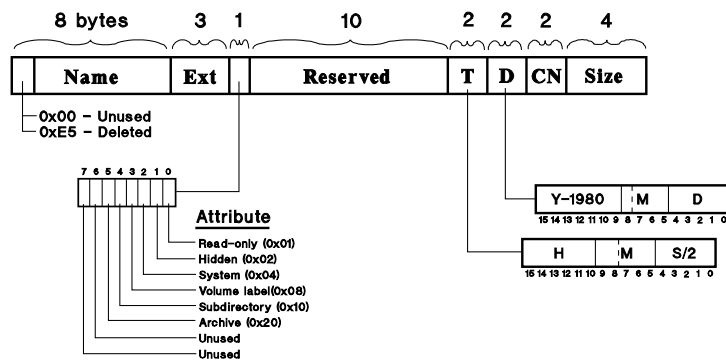
University of Idaho

FAT File Example



FATDISK00

DOS Directory Entry

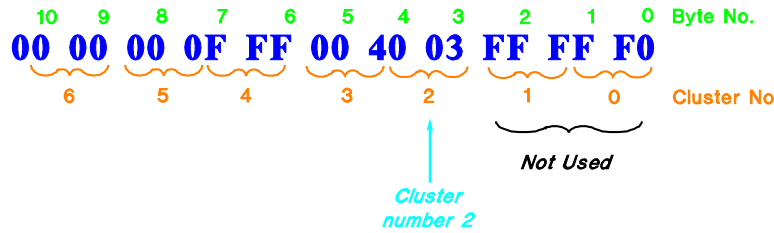


8 entries per block

DISK0030

12 Bit FAT Example

The following is the FAT table for a 1200 byte file,
1 block/cluster, 512 bytes/block, starting at
cluster number 2. DOS values are stored "little endian."



DSK0050

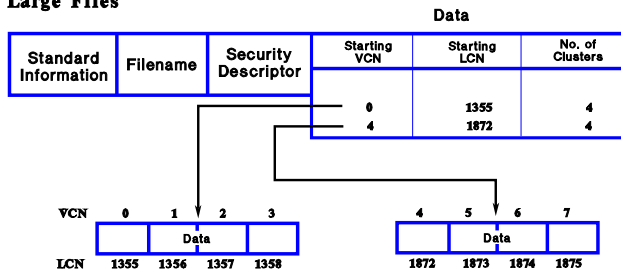
University of Idaho

NTFS "Regular" Files

Small Files

Standard Information	Filename	Security Descriptor	Data
----------------------	----------	---------------------	------

Large Files



NTFS0030

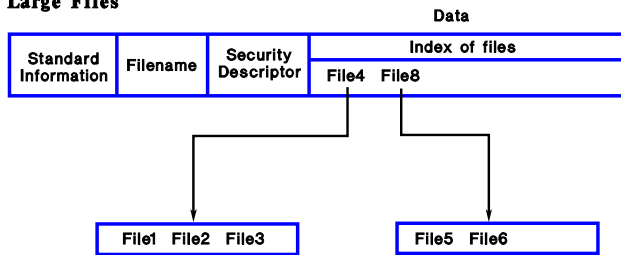
University of Idaho

NTFS Directory Files

Small Files

Standard Information	Filename	Security Descriptor	Index of files		
			File1	File2	File3 ...

Large Files



NTFS0040