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

UNIX i-node

- Directory Entry
- Pointer
  - Mode (file type)
  - Owner/Group
  - Timestamp
  - File size
  - Link count
  - Direct blocks
    - Single indirect
    - Double indirect
    - Triple indirect
Master Boot Record

Block zero on the disk

0000h
Boot Code

0. BIOS transfers to location 0000h
1. Locate "active" partition
2. Load 1st sector of partition into memory
3. Transfer execution to that code

0111h
Partition 1

0112h
Partition 2

0113h
Partition 3

0114h
Partition 4

Signature

DOS Partition

Boot Block contains

0x64 0x64 jump to bootstrap
0x68 0x68 Computer sig code
0x6b 0x6b bytes per block (map)
0x6c 0x6c Blocks/cluster (map)
0x6d 0x6d Reserved block (for boot record) (10)
0x6e 0x6e # of FATs (DPT)
0x6f 0x6f # of root directory entries (end)
0x70 0x70 # of logical blocks
0x71 0x71 Root Directory
0x72 0x72 # of blocks/FAT (map)
0x73 0x73 Master/track
0x74 0x74 # of heads (surfaces)
0x75 0x75 # of hidden blocks (2x)
0x76 0x76 ... Bootstrap program

Disk Partition

0 Boot Block
1 First block in
   First FAT

2 Last block in
   First FAT
   Blocks in any
   directory FATs

3 First block in
   root directory

4 Last block in
   root directory

5 Last block in
   data areas on disk
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."

```
10 9 8 7 6 5 4 3 2 1 0 Byte No.
00 00 00 OF FF 00 40 03 FF FF F0
8 5 4 3 2 1 0 Cluster No.
```

Cluster number 2

---

NTFS "Regular" Files

Small Files

<table>
<thead>
<tr>
<th>Standard Information</th>
<th>Filename</th>
<th>Security Descriptor</th>
<th>Data</th>
</tr>
</thead>
</table>

Large Files

```
<table>
<thead>
<tr>
<th>Standard Information</th>
<th>Filename</th>
<th>Security Descriptor</th>
<th>Data</th>
</tr>
</thead>
</table>
```

---

University of Idaho
NTFS Directory Files

Small Files

<table>
<thead>
<tr>
<th>Standard Information</th>
<th>Filename</th>
<th>Security Descriptor</th>
<th>Index of File</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>File1 File2 File3...</td>
</tr>
</tbody>
</table>

Large Files

<table>
<thead>
<tr>
<th>Standard Information</th>
<th>Filename</th>
<th>Security Descriptor</th>
<th>Index of File</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>File1 File2 File3...</td>
</tr>
</tbody>
</table>

Data

<table>
<thead>
<tr>
<th>File1</th>
<th>File2</th>
<th>File3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>File4</th>
<th>File5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>