Filesystem loops

Wesley Aptekar-Cassels

Hack && Tell NYC #44

Definitions

- Hard drives/SSDs
  - “Block devices” - Store data as an array of “blocks”
  - Block size varies, anywhere from 512B - 4KB
- Filesystems
  - Provide a way to go from file paths to blocks
  - Ex. `/home/wesley/foo.txt` -> Blocks 2343, 2353, 2647, and 3014

What filesystems are there?

Many, many filesystems. Most common include:

- FAT (1977, Windows < NT 3.1)
- NTFS (1993, Windows ≥ NT 3.1)
- ZFS (2005, Solaris, some BSDs)
- ext4 (2006, Linux)
- btrfs (2009, Linux)

FAT


Boot Record
Extended Boot Record
File Allocation Table
Data

FAT

Boot Record	Extended Boot Record	File Allocation Table	Data

Boot record:

- Master Boot Record (MBR)
- Partition information

FAT

Extended Boot Record:

- Volume label, etc

FAT

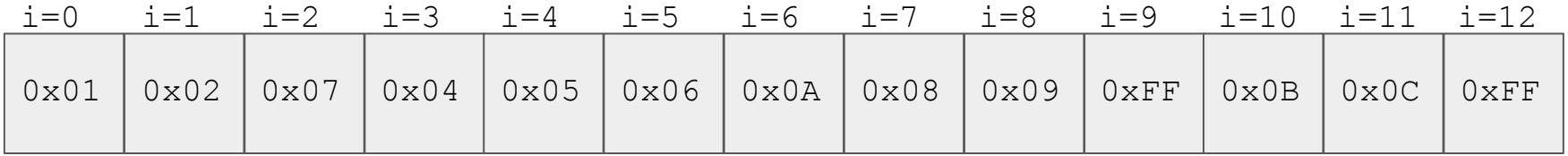
File Allocation Table

- Table that maps blocks -> next block in file
- Basically creates a linked list of blocks

FAT

File Allocation Table

- Table that maps blocks -> next block in file
- Basically creates a linked list of blocks

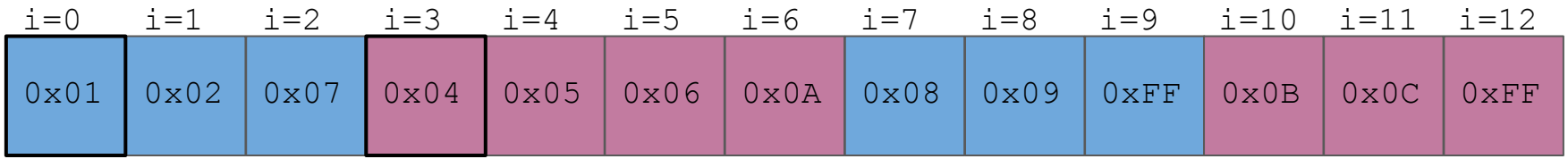


FAT

Boot Record	Extended Boot Record	**File Allocation Table**	Data

File Allocation Table

- Table that maps blocks -> next block in file
- Basically creates a linked list of blocks



FAT

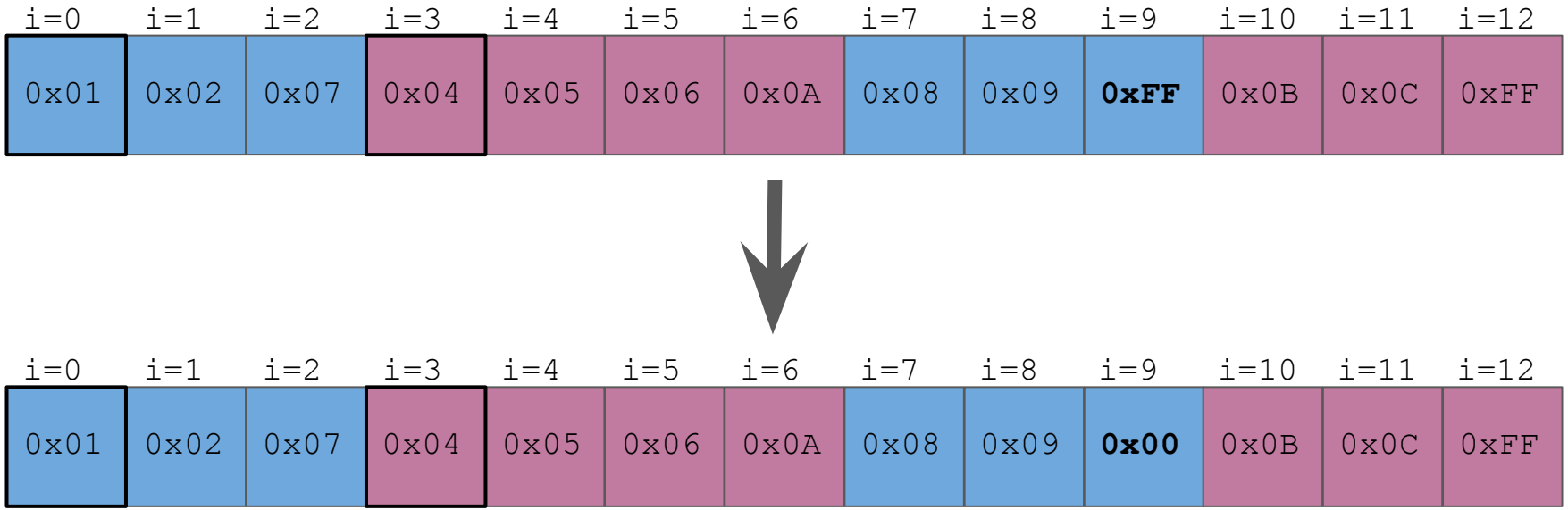
Boot Record	Extended Boot Record	File Allocation Table	**Data**

Data

- Split into blocks
- Each block can either be part of a file or a directory
- Directory metadata:
  - Name
  - Creation time
  - Last access time
  - Last modification time
  - File size

How can we break this?

Loops!


i=0 i=1 i=2 i=3 i=4 i=5 i=6 i=7 i=8 i=9 i=10 i=11 i=12
0x01 0x02 0x07 0x04 0x05 0x06 0x0A 0x08 0x09 0xFF 0x0B 0x0C 0xFF
i=0 i=1 i=2 i=3 i=4 i=5 i=6 i=7 i=8 i=9 i=10 i=11 i=12
0x01 0x02 0x07 0x04 0x05 0x06 0x0A 0x08 0x09 0x00 0x0B 0x0C 0xFF

Demo