FAT File-System

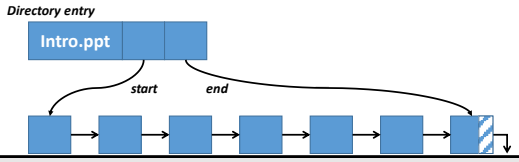
Dr Andrew Scott

a.scott@lancaster.ac.uk

1

Recap: *Linked-List Allocation*

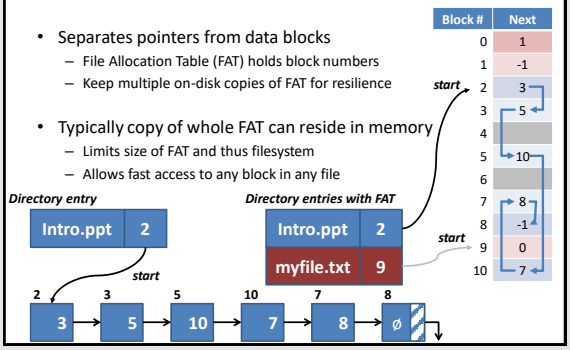
- Avoids external fragmentation problem
  - Blocks no longer need to be contiguous *(we accept overhead)*
    - All blocks can be used, but pointers take space
  - Fragmentation (in this case internal) confined to last block
- Only offers sequential access
  - Must follow each block link to reach given file position



2

Linked-List: *File Allocation Table (FAT)*

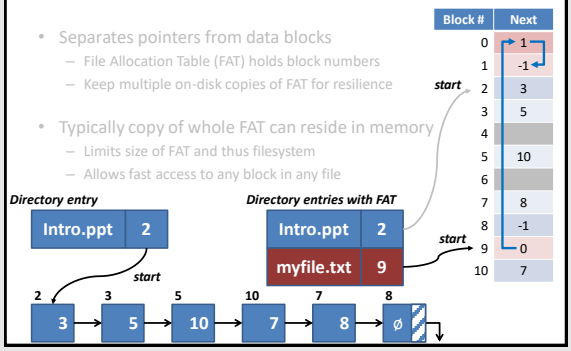
- Separates pointers from data blocks
  - File Allocation Table (FAT) holds block numbers
  - Keep multiple on-disk copies of FAT for resilience
- Typically copy of whole FAT can reside in memory
  - Limits size of FAT and thus filesystem
  - Allows fast access to any block in any file



3

Linked-List: *File Allocation Table (FAT)*

- Separates pointers from data blocks
  - File Allocation Table (FAT) holds block numbers
  - Keep multiple on-disk copies of FAT for resilience
- Typically copy of whole FAT can reside in memory
  - Limits size of FAT and thus filesystem
  - Allows fast access to any block in any file



4