FILE ALLOCATION

Module Number 4. Section 3
COP4600 – Operating Systems
Richard Newman

FILE SYSTEMS TOPICS

- Introduction
- Directories
- File Allocation
- Block Management
- File System Reliability
- File System Optimization

FILE ALLOCATION

- Methods
  - Contiguous
  - Indexed
  - Linked List (Chained)
  - FAT
  - Linked Indexed
  - Multilevel Indexed
- Metrics
  - Number of disk accesses to access byte of file
  - Complexity
  - Maximum file size

ALLOCATION METHODS

How is file stored on disk, access structures

1. Contiguous Storage:
   - FCB holds physical block number of block 0 (base)
   - Logical block i is physical block Base + i
2. Indexed Storage:
   - FCB holds location index block (IB)
   - IB holds pointers to blocks of file
   - File size limited by pointer size and block size
3. Linked List (Sequential/Chained) Storage:
   - FBC holds physical block number of block 0
   - Each block holds physical block number of next block in file (pointer/address)
   - Decreases storage available for file data
   - Require N disk accesses to reach logical block N

ALLOCATION METHODS(2)

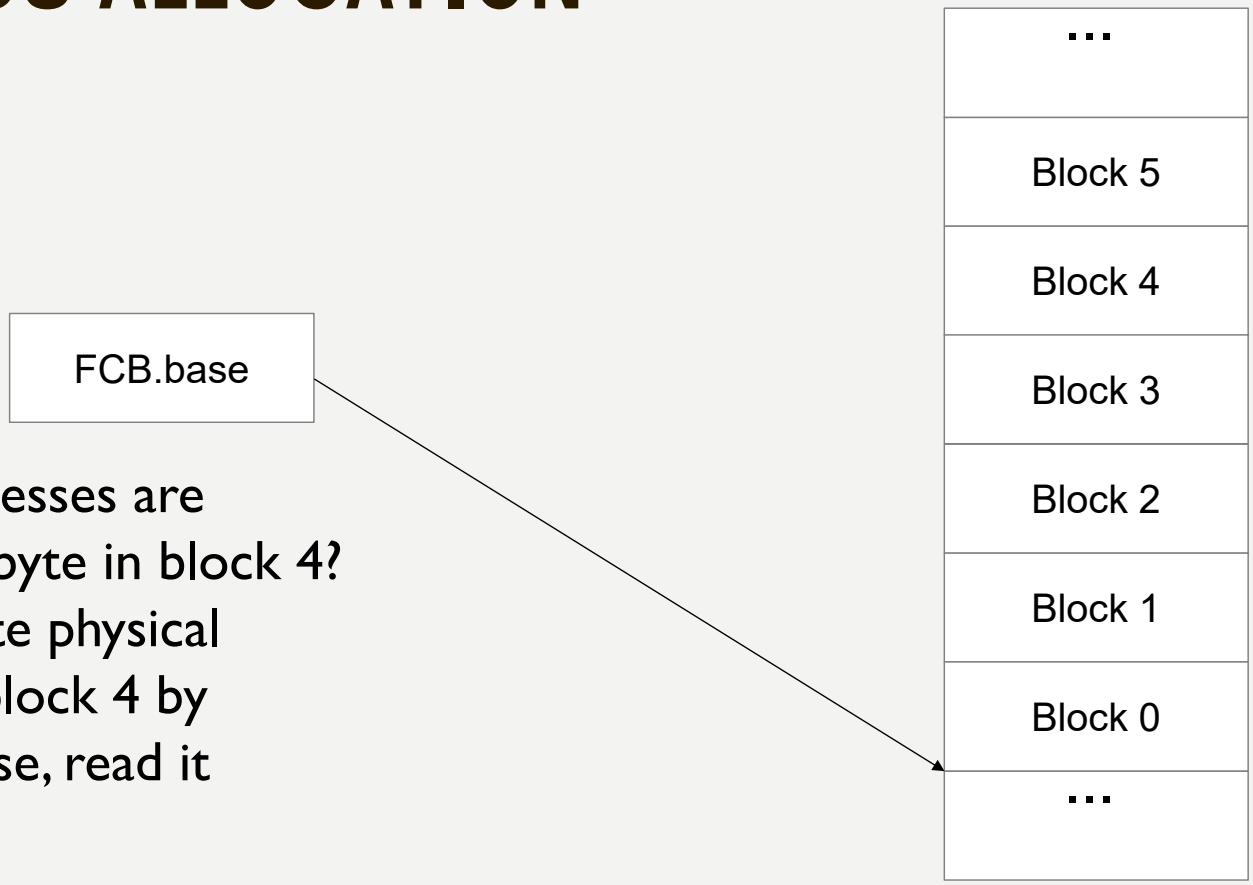
4. File Allocation Table (FAT)
   - Like linked list, except links are all in FAT not file
   - Chase links in FAT to find block N
   - Can also hold free block list(s)
5. Linked Indexed
   - One pointer set aside to point to next primary IB
6. Multi-level Indexed
   - Primary index blocks point to file blocks
   - Secondary index blocks point to primary Ibs
   - Tertiary IBs point to secondary IBs

How many disk accesses required to read byte i?

What is maximum file size?

What is complexity of mapping?

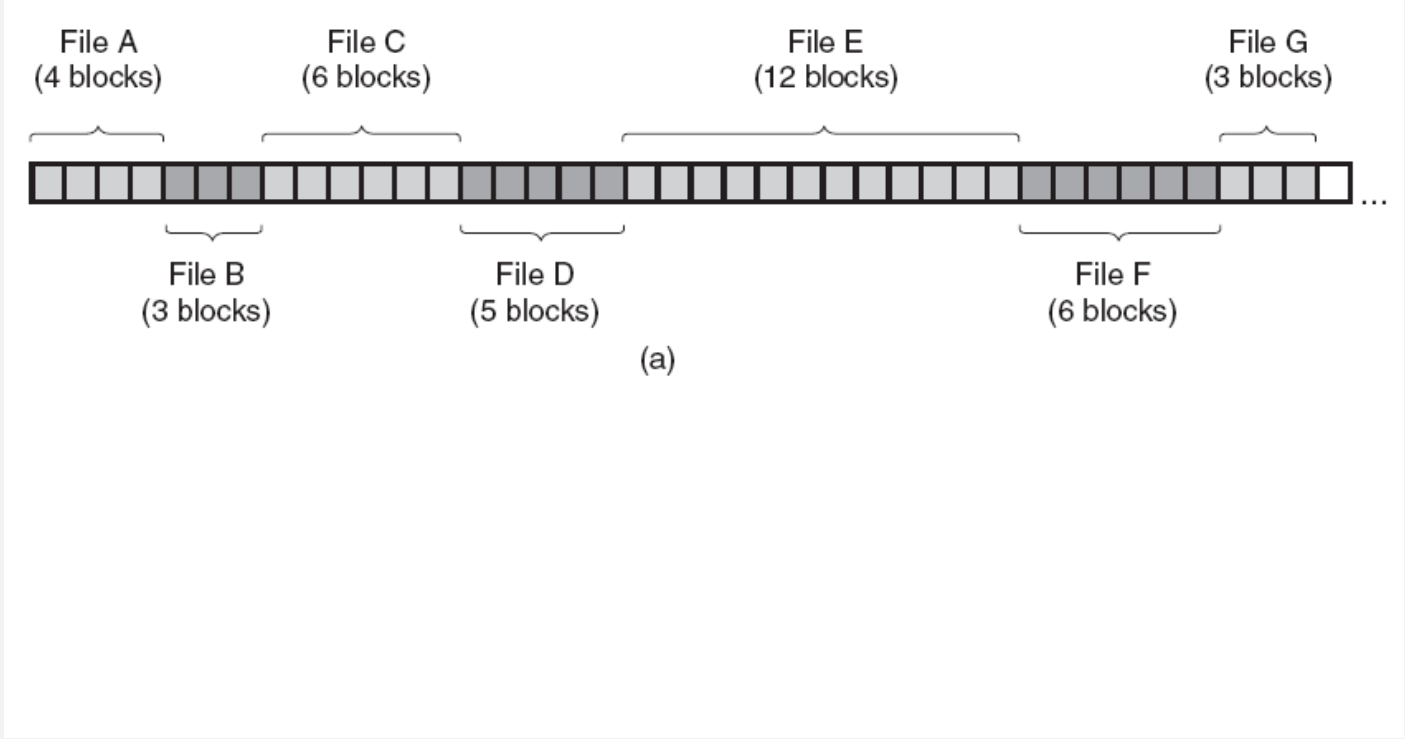
CONTIGUOUS ALLOCATION



How many disk accesses are required to read a byte in block 4?
Only one – compute physical location of logical block 4 by adding 4 to FCB.base, read it

Storing a file as a contiguous disk blocks.

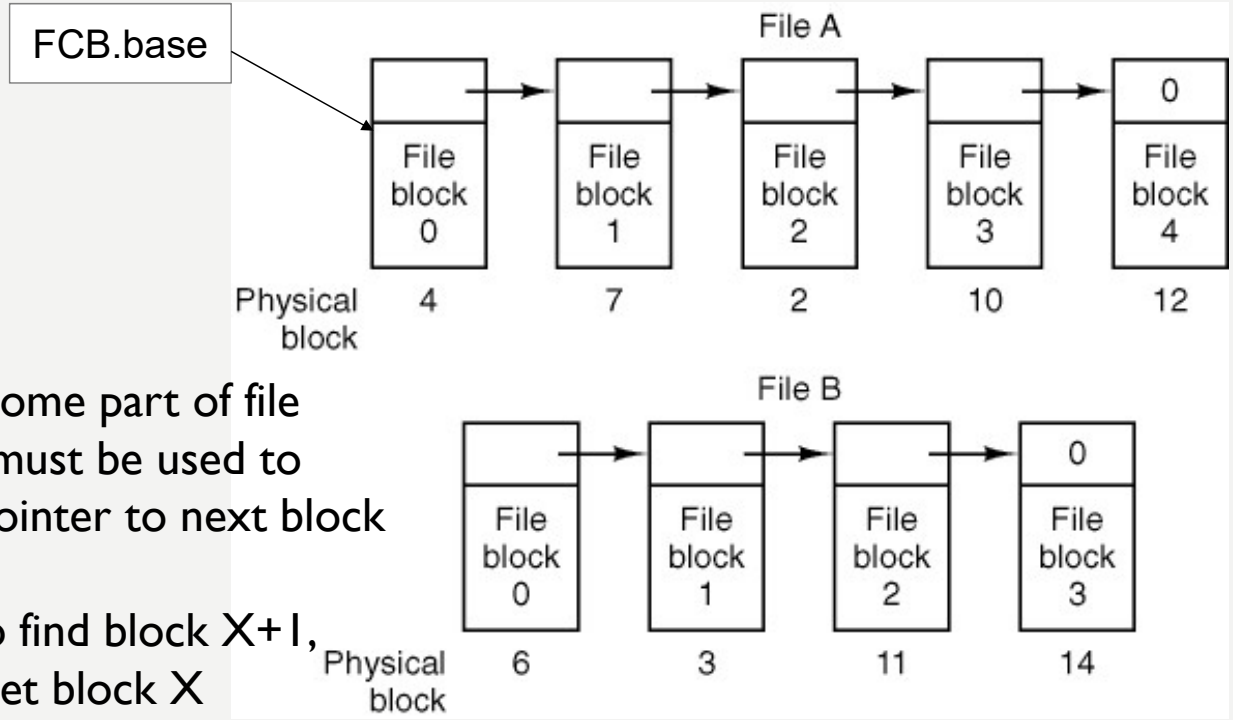
CONTIGUOUS ALLOCATION (2)



Leads to … Fragmentation! Which kind? External!

Figure 4-10. (a) Contiguous allocation of disk space for seven files. (b) The state of the disk after files *D* and *F* have been removed.

LINKED LIST ALLOCATION



Now some part of file block must be used to hold pointer to next block

…

And to find block X+1, Must get block X

disk accesses needed to read a byte in block 4?

Need 5!

Read B0 to find B1

Read B1 to find B2

Read B2 to find B3

Read B3 to find B4

Read B4 to get data

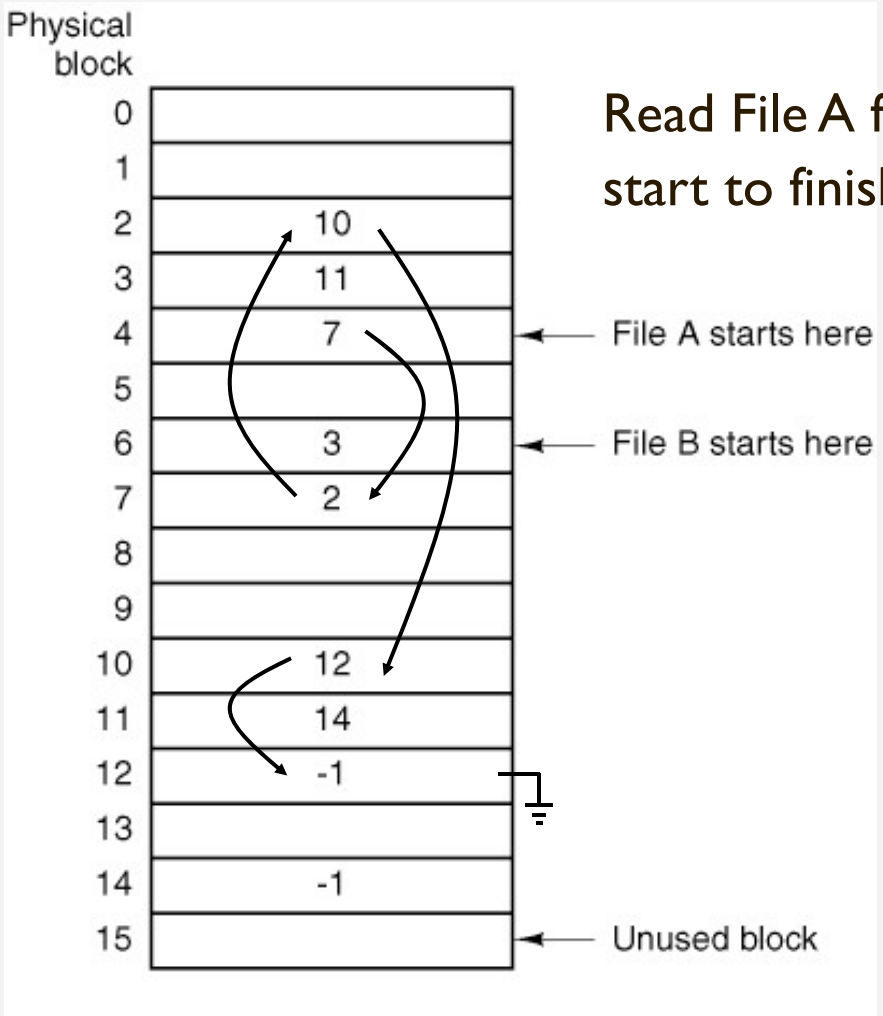
- Figure 4-11. Storing a file as a linked list of disk blocks.

LINKED LIST ALLOCATION USING A TABLE IN MEMORY

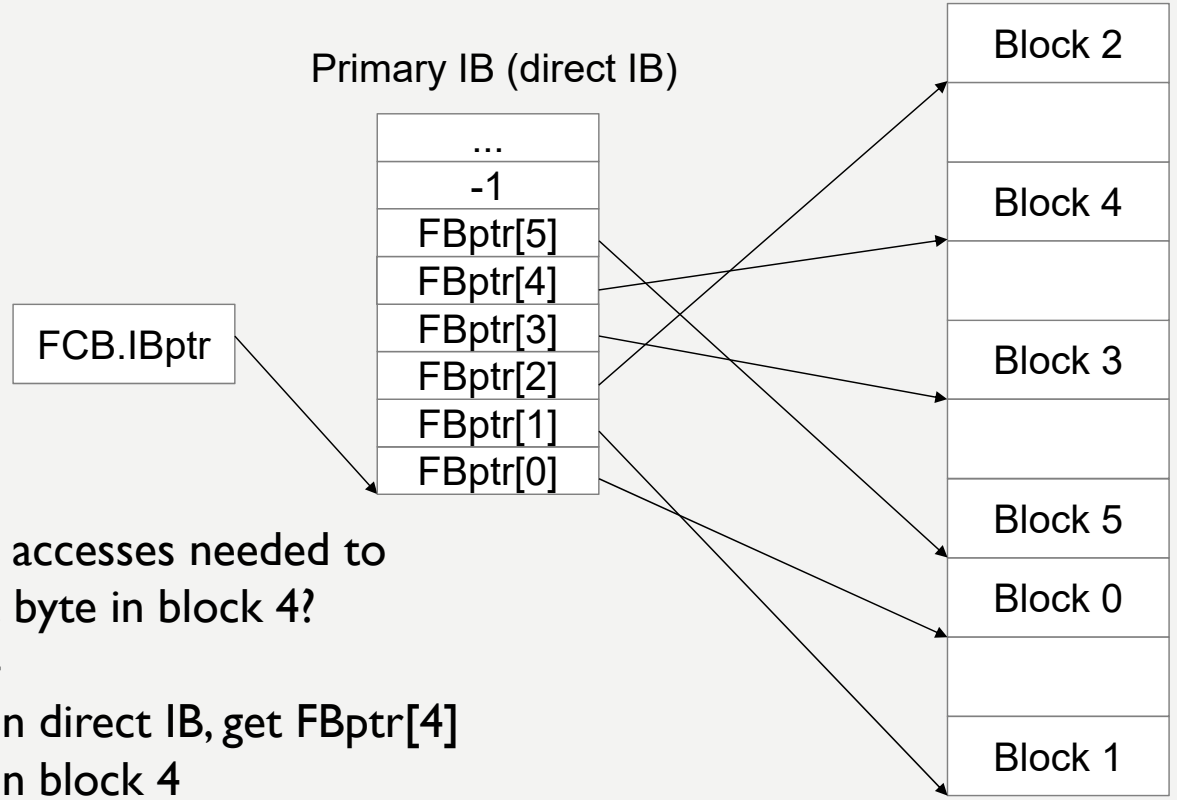
Figure 4-12. Linked list allocation using a file allocation table in main memory.

Note that (-1) is used as a sentinel value to indicate end of list.

Avoids N disk accesses to get block N of file.



INDEXED ALLOCATION



disk accesses needed to read a byte in block 4?
Two –
Read in direct IB, get FBptr[4]
Read in block 4

What is maximum file size?
Assume 1 KB blocks
Assume 32 bit ptrs

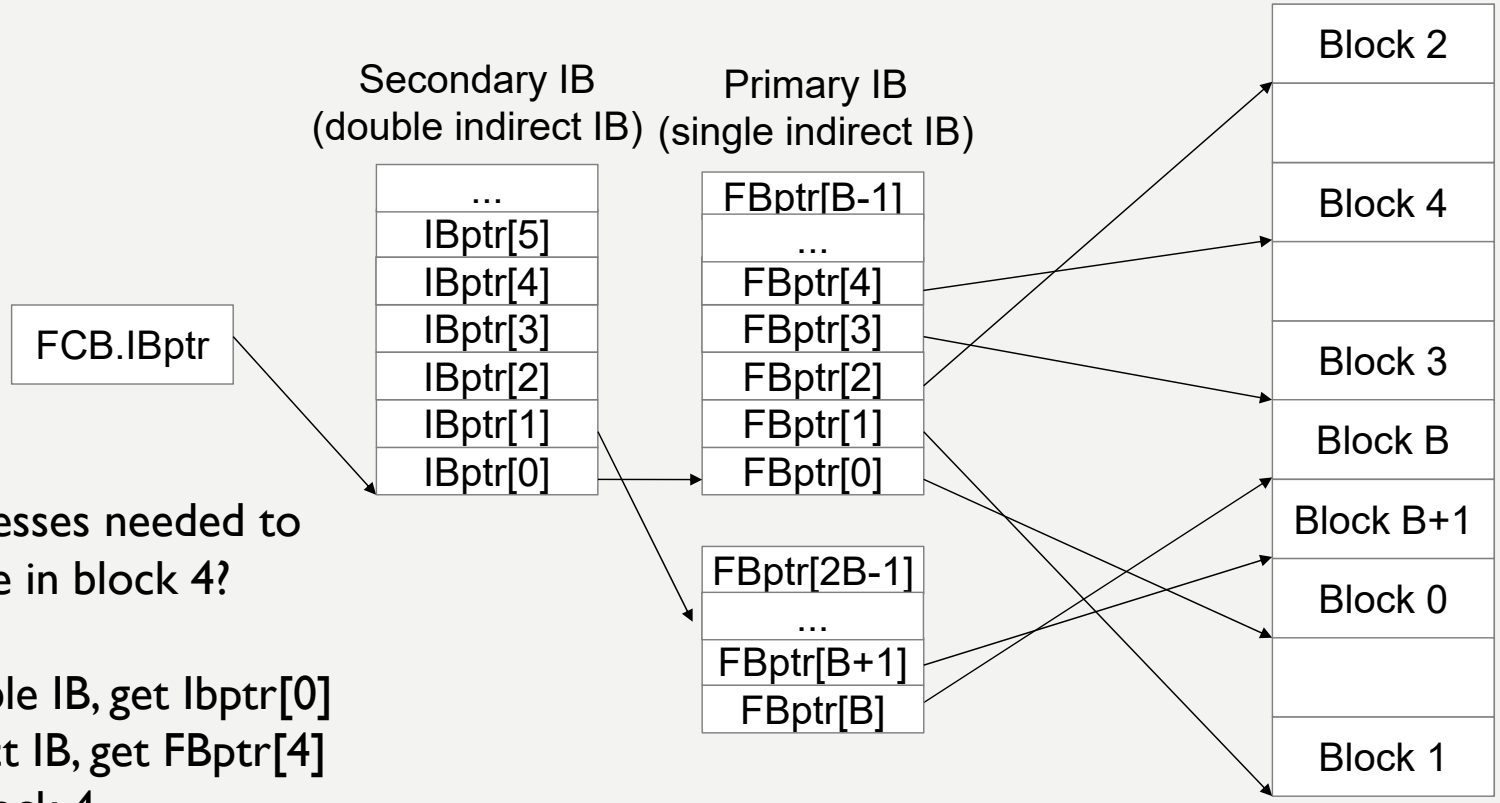
1 KB/4 B = 256 ptrs per index block

1 KB file block per pointer, so
256 KB is max

What if 4 KB blocks?
1024 pointers, and
4 KB per pointer, so
4 MB is max

Storing a file using an index (indirect) block.

MULTI-LEVEL INDEXED ALLOCATION



disk accesses needed to read a byte in block 4?
Three –
Read double IB, get Ibptr[0]
Read direct IB, get FBptr[4]
Read in block 4

Storing a file using an double indirect index block.

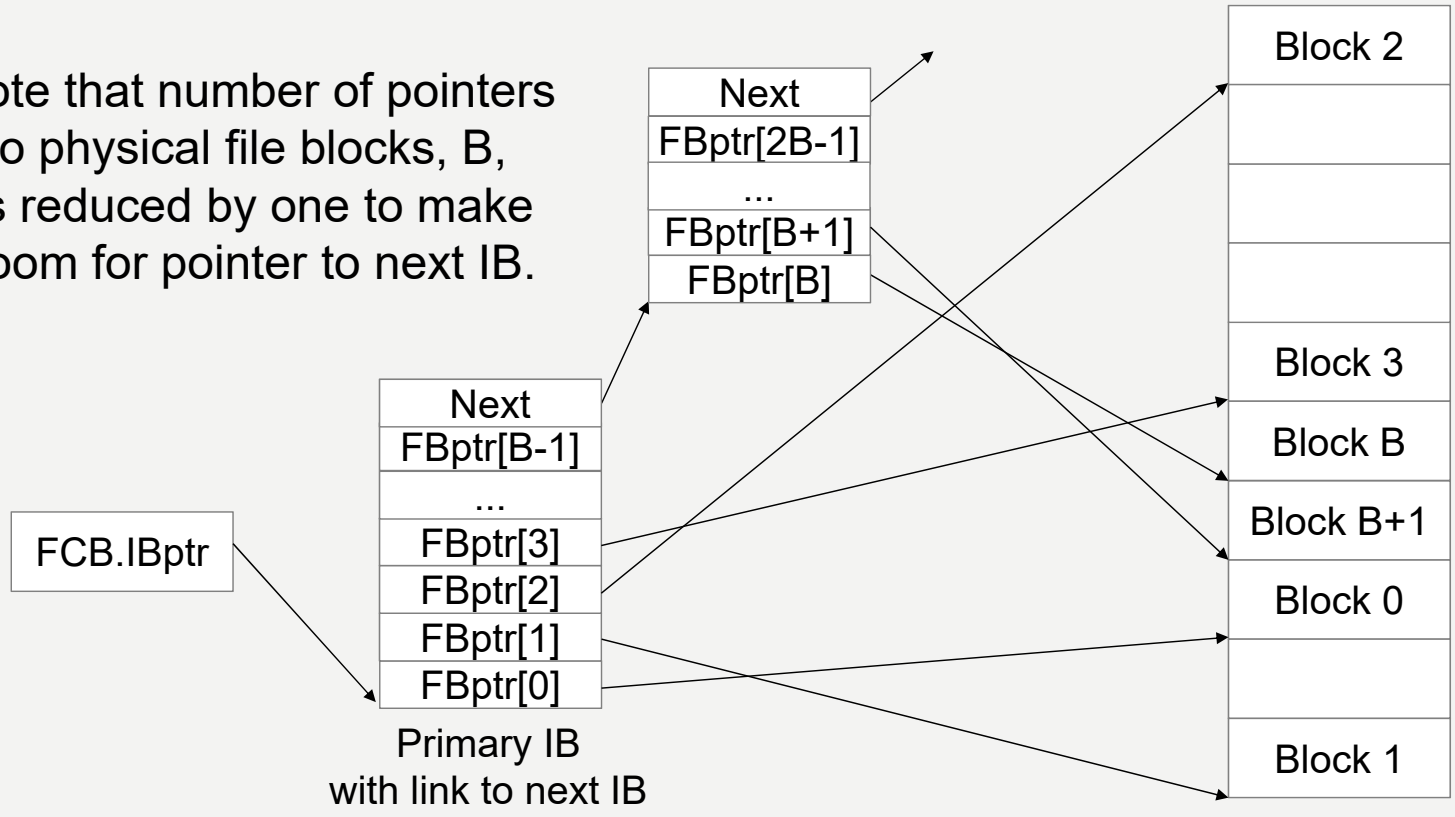
What is cost (in disk accesses) to read byte i?

What is maximum file size?(32 bit pointers, 4 KB blocks)

1024 pointers, per IB
1024 Primary Ibs so
1024^2 blocks, each 4 KB
So 4 GB is max file size

CHAINED INDEXED ALLOCATION

Note that number of pointers to physical file blocks, B, is reduced by one to make room for pointer to next IB.



Storing a file using an chained index block.

What is cost (in disk accesses) to read byte i?

What is maximum file size?

COST OF LOADING FILE BYTE I

Once FCB is in RAM, test i < Length

Contiguous Storage – get Base from FCB:

- N = i/B (B is block size) = logical block number
- Retrieve block Base+N

Indexed Storage – get IB loc. from FCB:

- N = i/B
- Retrieve Index Block (IB) if not in RAM
- Retrieve block IB[N]

Linked List Storage – get Block 0 loc. from FCB:

- N = i/(B-P) where P is pointer size in bytes
- Retrieve Block 0 to get location of block 1, etc....
- Until get Block N

How many disk accesses required?

IMPLEMENTING FILES I-NODES

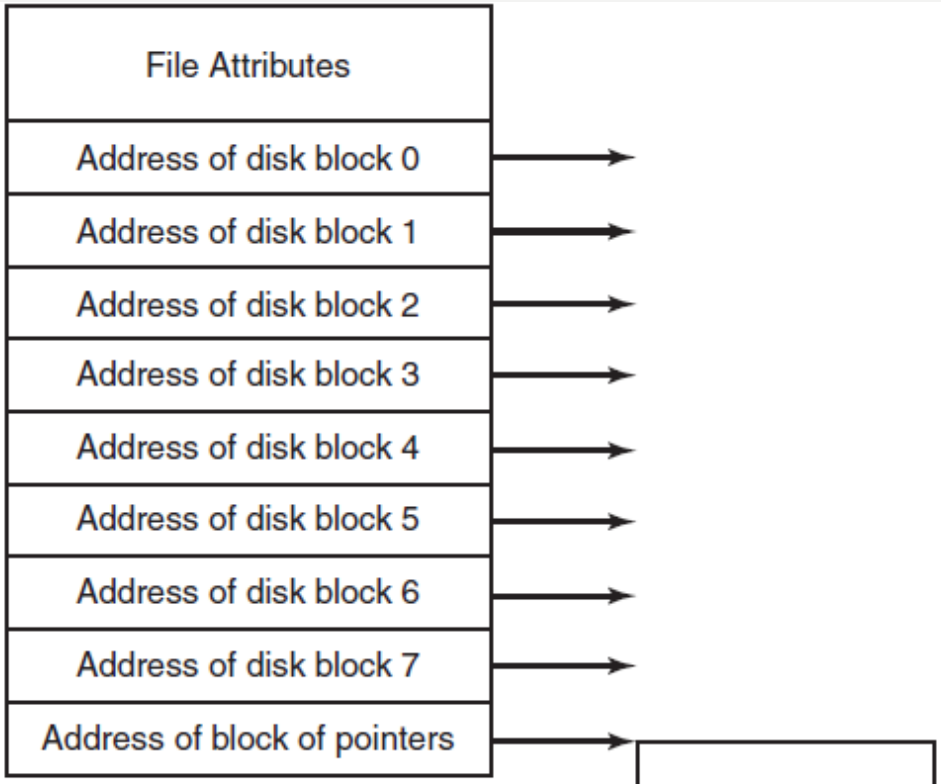


Figure 4-13. An example i-node.

I-NODES

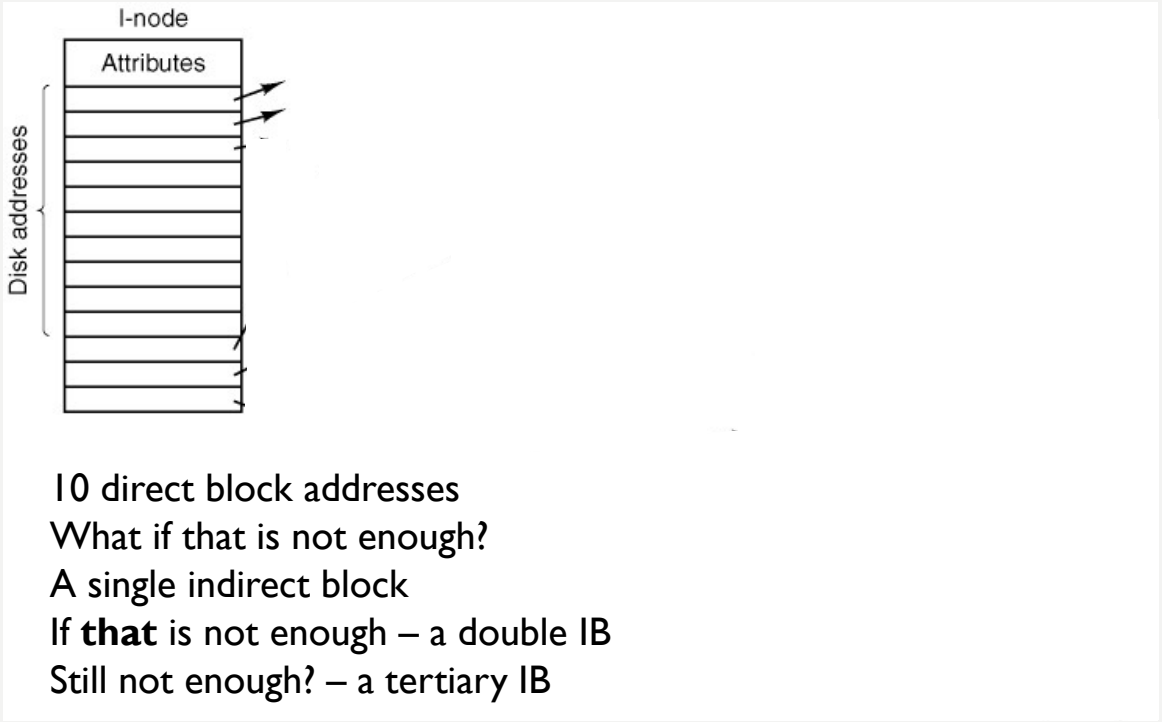


Figure 4-33. A UNIX i-node. Three levels of indirect blocks.

Access to first few blocks is very fast, but can handle large files.

What is maximum file size? (32 bit pointers, 4 KB blocks)

SUMMARY

- Methods
  - Contiguous – fastest, but fragmentation issues
  - Indexed – fast, max file size
  - Linked List (Chained) – slow for random access
  - FAT – puts chain links all together, saves space, fast
  - Linked Indexed – gets around max file size problem
  - Multilevel Indexed – quickly grow max
  - Unix Fast FS - fast for small files, can handle very large files
- Metrics
  - Access speed tied to number of disk accesses required
  - Maximum file size depends on allocation method, block size, pointer size, size of disk