6.2 Access Methods - Sequential, Direct, Swapping, File Allocation Methods- Contiguous, Linked, Indexed.

6.2 Access Methods and File Allocation Methods

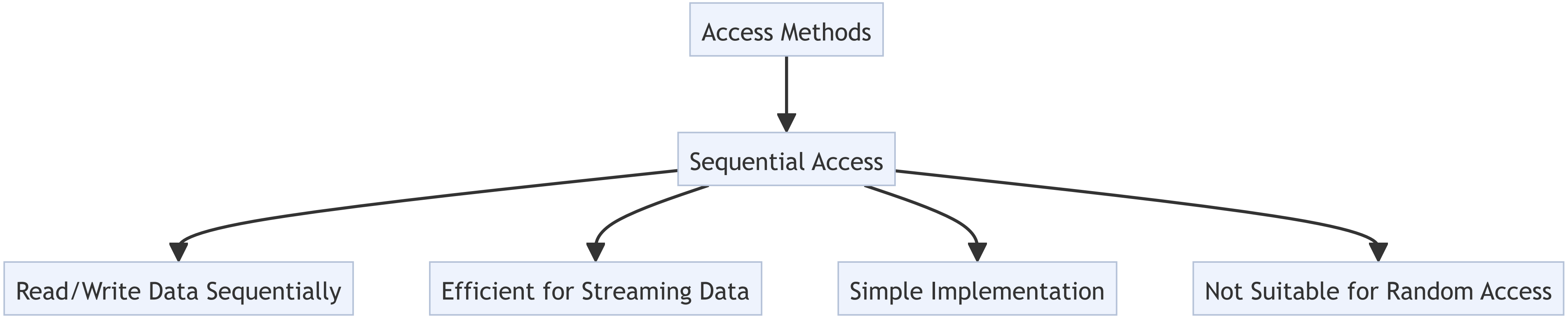
Access Methods

Access methods define how data is read from or written to a file. The primary access methods are:

Sequential Access

Working, Simple Usage, Explanation:
Sequential access reads or writes data in a linear order from the beginning to the end of the file. It is often used for text files and streaming data.

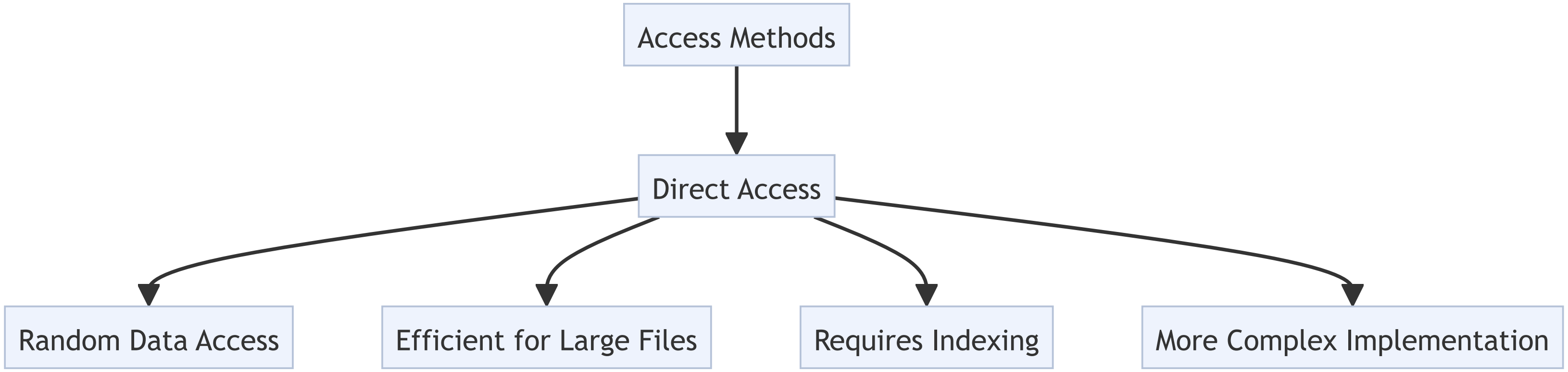
Diagram


Access Methods
Sequential Access
Read/Write Data Sequentially
Efficient for Streaming Data
Simple Implementation
Not Suitable for Random Access

Direct Access

Working, Simple Usage, Explanation:
Direct access allows data to be read or written in any order, without following a sequence. It is often used in databases and situations where quick access to specific records is required.

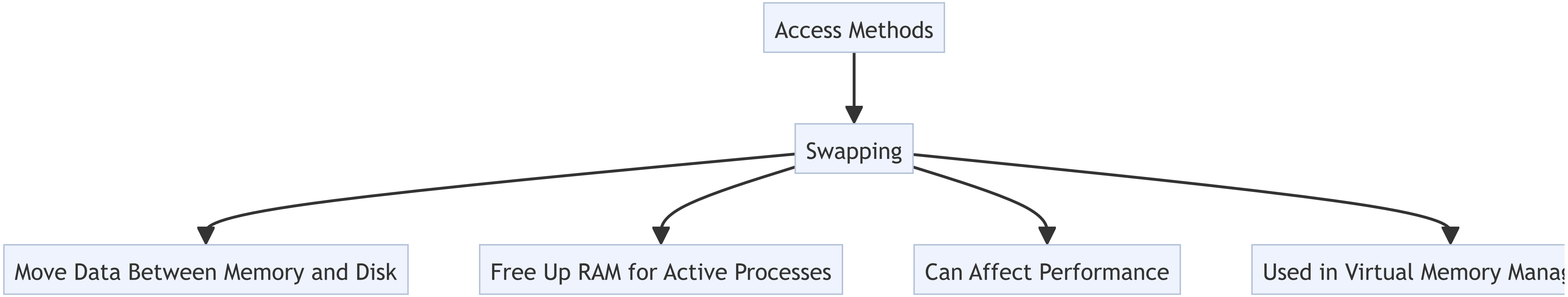
Diagram



Swapping

Working, Simple Usage, Explanation:
Swapping involves moving data between main memory and secondary storage. This method is used to free up memory for processes by temporarily storing inactive data on disk.

Diagram


Access Methods
Swapping
Move Data Between Memory and Disk
Free Up RAM for Active Processes
Can Affect Performance
Used in Virtual Memory Mana

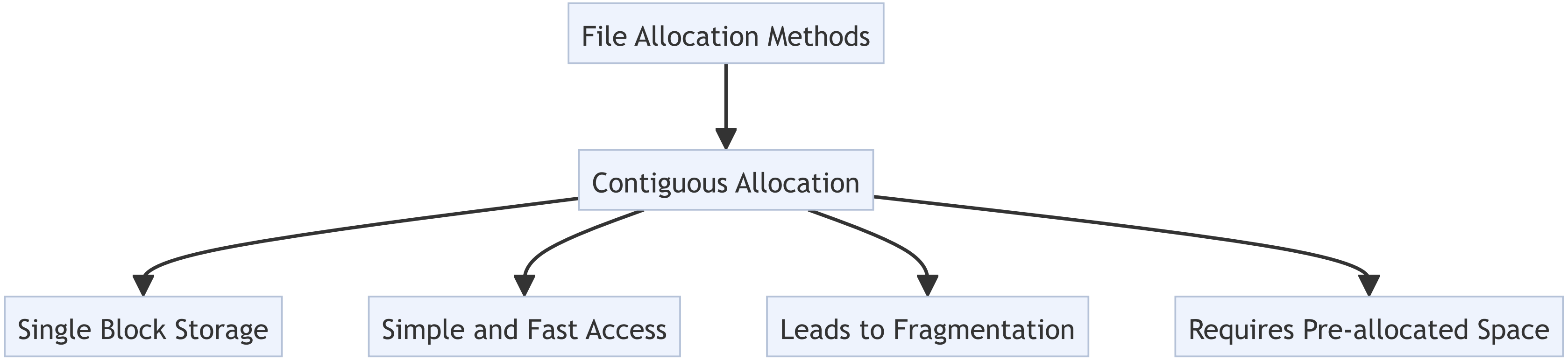
File Allocation Methods

File allocation methods determine how files are stored on disk and how they can be accessed.

Contiguous Allocation

Working, Simple Usage, Explanation:
Contiguous allocation stores a file in a single, contiguous block of disk space. It simplifies access but can lead to fragmentation.

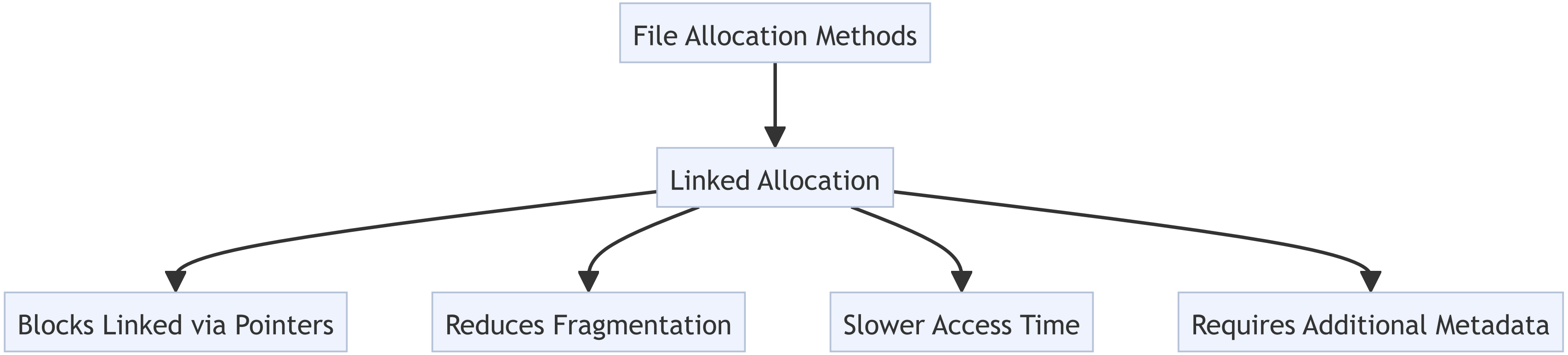
Diagram



Linked Allocation

Working, Simple Usage, Explanation:
Linked allocation stores file blocks scattered across the disk but keeps track of their order using pointers. It reduces fragmentation but may lead to slower access times.

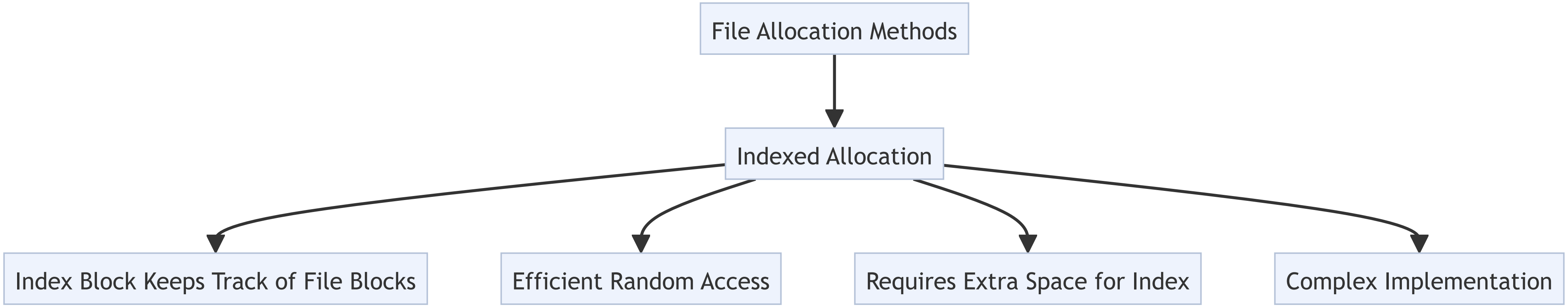
Diagram



Indexed Allocation

Working, Simple Usage, Explanation:
Indexed allocation uses an index block to keep track of all the disk blocks used by a file. It allows for efficient random access but requires extra space for the index block.

Diagram



Summary Table

Aspect	Details
Sequential Access	Linear data read/write, simple, efficient for streaming data.

Aspect	Details
Direct Access	Random data access, efficient for large files, complex.
Swapping	Moves data between memory and disk, frees RAM, can affect performance.
Contiguous Allocation	Single block storage, simple access, leads to fragmentation.
Linked Allocation	Blocks linked by pointers, reduces fragmentation, slower access.
Indexed Allocation	Index block tracks file blocks, efficient random access, requires extra space.