

NTFS

Log Recovery

NTFS Log Recovery

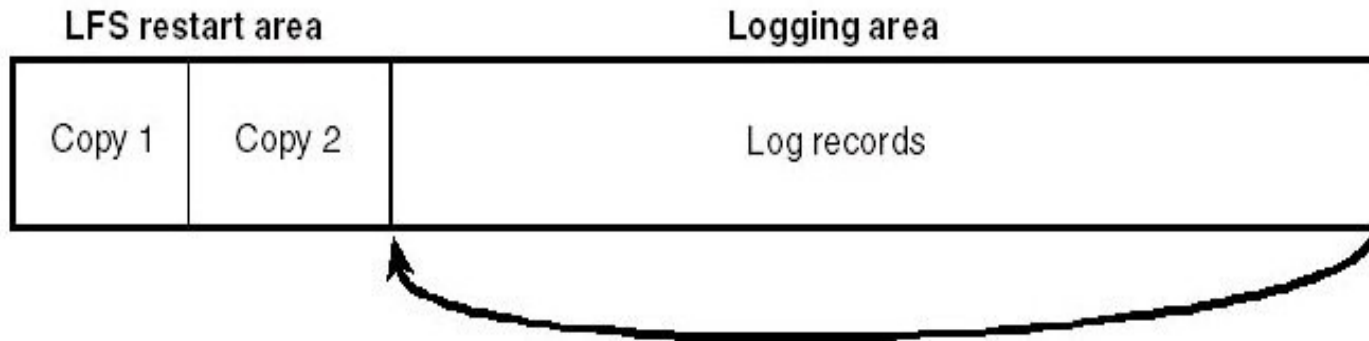
- NTFS provides file system recoverability by means of a logging.
- In NTFS logging, the suboperations of any transaction that alters important file system data structures are recorded in a log file before they are carried through on the disk.
- That way, if the system crashes, partially completed transactions can be redone or undone when the system comes back online
- N.B:
 - The FAT file system uses a *write-through* algorithm that causes disk modifications to be immediately written to the disk

LFS

- The log file service (LFS) is a series of kernel-mode routines inside the NTFS driver that NTFS uses to access the log file.
- Although originally designed to provide logging and recovery services for more than one client, the LFS is used only by NTFS.
- The caller—NTFS in this case—passes the LFS a pointer to an open file object, which specifies a log file to be accessed.

LFS

- The LFS divides the log file into two regions: a restart area and an "infinite" logging area



LFS

- NTFS calls the LFS to read and write the restart area.
- NTFS uses the restart area to store context information such as the location in the logging area at which NTFS will begin to read during recovery after a system failure.
- The LFS maintains a second copy of the restart data in case the first becomes corrupted or otherwise inaccessible.

LFS

- The remainder of the log file is the circular logging area.
- The LFS uses 64 bit logical sequence numbers (LSNs) to identify records written to the log file.
- NTFS never reads transactions from or writes transactions to the log file directly .

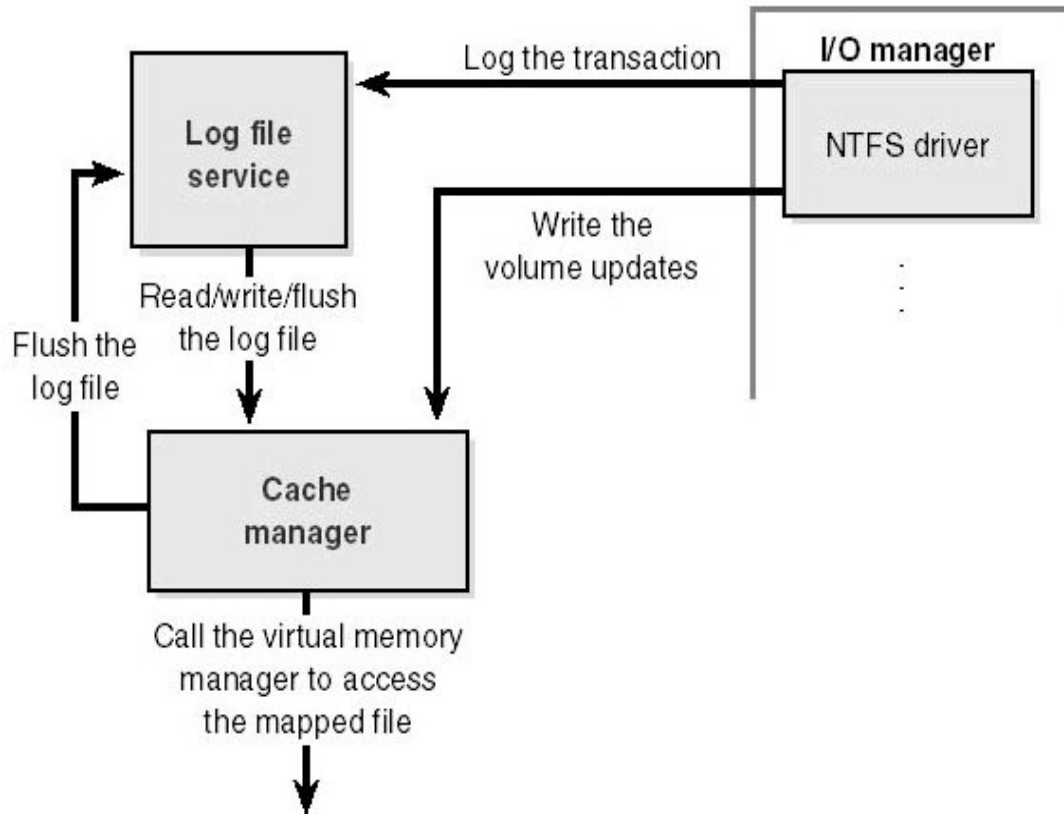
LFS

- The LFS provides services NTFS calls to:
 - open the log file
 - write log records
 - read log records in forward or backward order
 - flush log records up to a particular LSN
 - set the beginning of the log file to a higher LSN

LFS

- NTFS:
 - calls the LFS to record in the (cached) log file any transactions that will modify the volume structure.
 - modifies the volume (also in the cache)
 - The cache manager prompts the LFS to flush the log file to disk.
 - The LFS implements the flush by calling the cache manager back, telling it which pages of memory to flush
 - After the cache manager flushes the log file to disk, it flushes the volume changes (the metadata operations themselves) to disk.

LFS



Log Record Types

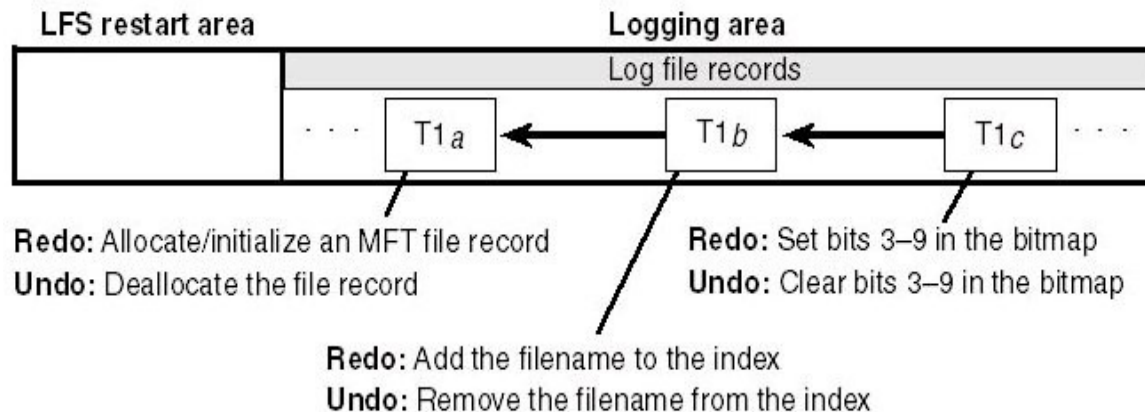
- The LFS allows its clients to write any kind of record to their log files
- NTFS writes several types of records
- Amongst the others:
 - update records
 - checkpoint records

Update Records

- Each update record contains two kinds of information:
 - **Redo information**
 - How to reapply one sub-operation of a fully logged transaction to the volume if a system failure occurs before the transaction is flushed from the cache
 - **Undo information**
 - How to reverse one sub-operation of a transaction that was only partially logged at the time of a system failure

Update Records

- Each record represents one sub-operation of a transaction
 - The *redo* entry in each update record tells NTFS how to reapply the suboperation
 - The *undo* entry tells NTFS how to roll back the suboperation



Logging

- After logging a transaction NTFS performs the sub-operations on the volume itself, in the cache
- When it has finished updating the cache, NTFS writes another record to the log file, recording the entire transaction as complete

Logging

- When recovering after a system failure, NTFS reads through the log file and redoes each committed transaction.
- After redoing the committed transactions during a file system recovery, NTFS locates all the transactions in the log file that weren't committed at failure and rolls back

Logging

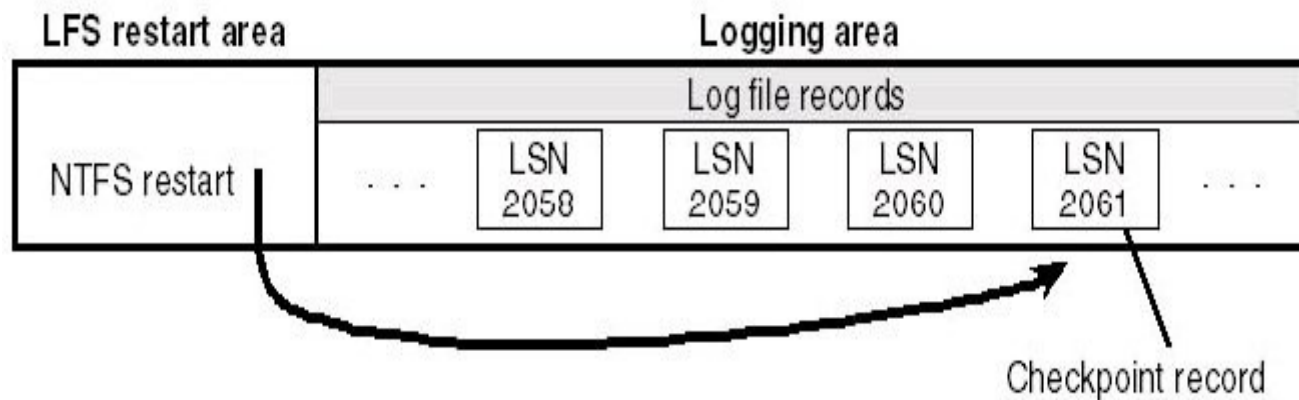
- The format of the update records must ensure that executing redundant redo or undo operations is idempotent
- For example, setting a bit that is already set has no effect, but toggling a bit that has already been toggled does

Logging

- NTFS writes update records for the following operations:
 - Creating a file
 - Deleting a file
 - Extending a file
 - Truncating a file
 - Setting file information
 - Renaming a file
 - Changing the security applied to a file

Checkpoint Records

- In addition to update records, NTFS periodically writes a checkpoint record to the log file



Checkpoint Records

- Tell NTFS what processing would be needed to recover a volume if a crash were to occur immediately.
- After writing a checkpoint record, NTFS stores the LSN of the record in the restart area

Log Full

- If the log file doesn't contain enough available space LFS returns a "log file full" error and NTFS raises an exception.
- The NTFS exception handler rolls back the current transaction and places it in a queue to be restarted later.
- NTFS blocks file creation and deletion and then requests exclusive access to all system files and shared access to all user files.

Log Full

- Gradually, active transactions either are completed successfully or receive the "log file full" exception. NTFS rolls back and queues the transactions that receive the exception.
- After flushing to disk, NTFS resets the beginning of the log file to the current position, making the log file "empty."
- Then it restarts the queued transactions