

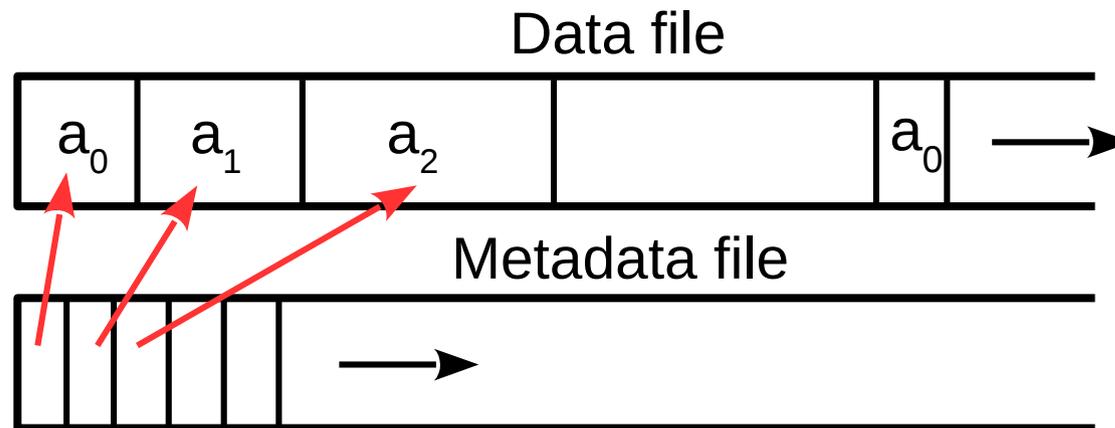
LFS Implementation Document

Header file

- Functions
 - **ifs_write**: writes the given data by appending to the EOF.
 - **read_record**: reads the Metadata file to obtain the dict.
 - **compare_tup**: compares two given tuples to find common area.
 - **ifs_find_chunks**: recursive function that finds the chunks to read.
 - **ifs_read**: performs the read.
- Structs
 - **Tup**: a tuple to address the areas we have written.
 - **ifs_record_on_disk**: data structure for writing to Metadata file.
 - **ifs_record**: our log which works like a mapping dict.

Write Function

- void **lfs_write**(size_t *addr*, char * *data*)
 - Write the given ***data*** at the end of the data file.
 - Write the given ***addr*** and the size of ***data*** in the metadata file.



Read Function

- `size_t lfs_read(size_t addr, size_t size, char * res)`
 - Call `read_record()` function to get updated log list.
 - Call `lfs_find_chunks()` for the records and given query(`addr, addr + size`) so it will return back list of exact chunks that should be read to get the data correctly.
 - Read the given chunks and put them in the given `res` array.

Recursive find_chunks Function

- `lfs_find_chunks(size_t a, size_t b, int index, lfs_record* my_recs, vector<>& chunks_stack)`
- `a, b ==>` start & end of the query
- `Index ==>` index of the item in the `log_list`
- `my_recs ==>` pointer to `log_list`
- `chunks_stack ==>` contains the found parts of latest data for given query

Put B area in the found stack;
Recall yourself for areas A and C;

