

AOS Project

Problem statement - To simulate the functioning of getBlock algorithm in a multi programming environment either by fork or thread.

Program Language – Java

AOS concept implemented

- **Multi-Programming by** : Multi threading
- **Locking mechanism** : Using synchronized concept of java and making own critical section for concurrently executing processes.
- **Signal Handling** : Done, using *wait()*, *notify()* and *notifyAll()* methods of java thread class.

getBlock cases handled

- All five cases according to Morris Bach book can be handled by the program that we have built.
- *bLockRequest()* method is to be called it has 2 parameters, 1st blockNumber 2nd name of the thread requesting for the block.
- Delayed write case will be handled in accordance to Morris Bah Book.

Contributions

Abhinav Kumar

Saurav Bharti

Synchronization part	Data structure definition
Threading	Insertion and deletion method
Signalling part	Initial env setup part

AOS Project

Learning part

Abhinav – got to know how threads really work and when they should be used, how to handle signalling between multiple threads executing concurrently using wait() and notify() and it's variation, how to use join() for threads, how to handle time for large projects with complex corelated methods to be implemented.

Saurav Bharti

- How freelist and hashqueues are maintained internally
- How requests are made for a block and how they are handled.