

Kuan Yang kxy200010@utdallas.edu

1. Describe synchronization mechanism for the sorting, merging, and Input threads

For sorting and merging:

There are 2 locks to guarantee mutual exclusion, one lock for sort_thread, one for merge_thread. Sort_thread only allow 1 thread to sort at a time, merge_thread only allow 1 thread to merge at a time.

Inside sorting process, a new sorting request must wait semaphore until previous sorting requests have finished layer2(each 4 segments of sorting finished). When layer2 finished sorting, the array will go to layer3 for merging, which means Layer 2 sorting is available now, it will signal semaphore to accept new sorting job and sort. In addition, the merge layer will wait semaphore until all the segments of sorting has finished before start merging.

For input threads :

In cal.c if all sorter arrays have been in use, the new input must wait semaphore until one sorter array becomes available. When any sorter arrays finish 4 layers of sorting and merging, and has printed the sorted result, it will signal semaphore to accept new input to sort. Thus, the input layer will synchronize with last layer merger.

2. Describe synchronization mechanism between Admin and Cal

Admin forwards sorting jobs to cal.c through pipe. Cal.c have N number of sorter arrays, if all the arrays have been in use, the new request have to wait semaphore until one becomes available. When any sorter arrays finished sorting, merging , it becomes available, it will signal semaphore to accept new arrays to sort. Thus, the input layer will synchronize with last layer merger.

Pseudo code between admin and cal:

<p>empty = N ,full = 0 , number of sorter array=N</p> <p>Admin :</p> <p>Loop</p> <p>Wait(empty);</p> <p>Wait(mutex);</p> <p>send array;</p> <p>signal(mutex);</p> <p>signal(full)</p> <p>end</p>	<p>Cal:</p> <p>Loop</p> <p>Wait(full);</p> <p>Wait(mutex);</p> <p>receive array;</p> <p>signal(mutex);</p> <p>signal(empty)</p> <p>end</p>
--	--

Explain: admin waits on semaphore empty to check how many sorter arrays are available to sort. If there are sorter arrays, it takes the lock, enters the critical section(CS) and send the array to cal.c. Once the

array is sent, admin exits the CS, signal semaphore full. cal.c now enters the CS, receive the array to sort, and signal semaphore empty. Admin cannot send any request when all the sorting arrays are taken. Cal.c cannot sort when there is no array received.