CHAPTER 7
CONCLUSION AND FUTURE SCOPE

In this study, a buddy allocator dynamic memory allocation algorithm for efficient memory management in real time environment has been implemented. In this method when a memory block is allocated or released then the free and used memory blocks (displayed in the hyper terminal) can be easily found by simple address computation. This method enables easy allocation and deallocation of memory blocks, minimizes the fragmentation, ensures optimal utilization of memory and good locality among memory blocks. Although, this method is optimal, but has a residual fragmentation due to restricted block sizes. The scheduler for resource awareness embedded system is designed in real-time environment. Two real-time case studies are taken one for sorting of numbers using non threading and threading concepts and second one being the sorting of alphabets using threading as well as non threading concepts and for both cases the scheduler performance is obtained. The scheduler presented shows improved performance over the reported methods in terms of performance criteria like Utilization, Throughput, Queuing Time, Response time etc. The method presented is effective studied with practical implementation using multicore units also.

Solution for Priority inversion problem by three tasks(Task 1, Task 2, Task 3) having varied priorities is presented and comparative study is made for slow motion system (SMS) using Semaphore and Fast Motion System (FMS) using Mutex. By using Mutex SMS is converted into FMS and solves priority inversion problem. Thus the input sampling time has been improved. Searching a data in large data base has been a profound area for researchers. In this research work OpenMP Tools is used to perform multithreading based search. The motive to use OpenMP is that the user can specify a parallelization strategy for a program. The experiment of data searching using multithreading is conducted for a large data base. The experiments are conducted with and without OpenMP and their performance is presented. The results obtained shows that the time required for searching a data using proposed work is less compared to reported works. The research work for its practical implementation requires atleast the same number of multicore units as that of the number of threads.
Future direction of research shall be on overcoming the residual fragmentation limitation and design a memory allocator that solves the problem of allocation based on the task behavior and extend the use of real time concepts in message box and mail queue. Performance improvement in Queue handling for multiple tasks in real time applications with more metrics is also an unexplored area for researchers.