Maayez Imam & Abdullah Mohammed

report.pdf

LLRB-based set, map and multimap:

For the map container I created from the set container, I determined the coverage of the map_tester.cc by first creating a map container object. I then created vectors for the keys and the values and inserted them all, using Insert(), into the map container object. I then printed them out, using Print(), to check that each node has the correct key and value associated with it. I modified the Print() method temporarily in order to also see the value attached to each key. After inserting and printing the nodes, I implement the Get() method to get the value of a specific key and I print it out so that I can check if it is correct. I then test the Remove() method by iterating through the keys and deleting them from the map container object. I use the Print() method again to print out the nodes after deletion, which ideally should print nothing.

For the multimap container I created from the map container, I determined the coverage of the mulitmap_tester.cc by first creating a multimap container object. I then created a vector of keys and vector of lists of values. I tested the Insert() method by inserting them all into the multimap container object and printed them out using the Print() method. I temporarily modified the print method to print out the values as well as the keys. After checking to see if they were all inserted properly, I implement the Get() method to test it out by printing out what Get() returns for a specific key. I then test the Remove() method by implementing it and checking if the entire node was removed if it had only one value attached to it or if the first value in the list of values was removed if the node had multiple values attached to it.

Completely Fair Scheduler:

A CFS or a completely fair scheduler manages tasks to ensure that every task has enough access to the processor to run. For our implementation of the CFS scheduler, we decided to use a variety of data structures so that everything would run smoothly. To store the data from the file containing each task, we utilized structs which were stored in a pointer of the struct type which was finally pushed to a vector full of pointers. In the struct, the identifier, start time, duration, and virtual runtime were stored. Then the code proceeds to the while loop which is also the scheduling loop and follows the distinct steps outlined in the prompt. A multimap is used for the timeline of tasks and the key of tasks added to the timeline are their virtual runtime. When all the distinct steps outlined in the prompt are completed and there are no more tasks to complete, a boolean variable called allTasksCompleted is set to true and the while loop terminates. Some functions which were utilized for this assignment's implementation were the ReadFile which reads the file and assigns values to the structs and returns them in a vector, a function to find which keys start time match a specific tick, and a compare method so that sorting with the key works.