

PROGRAMMING 2 PROJECT

Protik Banerji (s2342898), Dharanish NH (2372525)

17/04/2020

Problem 7.3.1

What is a priority queue?

A priority queue is an advanced version of the queue where we assign priority to every element in the queue. Elements that have the highest priority are popped from the queue (dequeued) first. If there are any two elements in the queue with the same priority they are removed according to their order in queue..

Source: <https://www.geeksforgeeks.org/priority-queue-set-1-introduction/>

Why would the writer of the code of MessageBus.h have chosen for storage of TextMessage*, instead of TextMessage?

In general it is always better to call data by the pointer of the variable. When we have large value for a variable. As it avoids creating multiple copies of variable while performing some operation and also reduce the time in fetching variables that are large. Pointer also helps in referring the same space in memory from multiple locations. Polymorphism means the some code or operations or objects behave differently in different contexts. So it would be very useful to point the location with pointers of different data types based on the contexts. Object slicing happens when a derived class object is assigned to base class and the additional attributes of derived class are sliced off to form base class object. To overcome this we can use pointers to base is required to be able to refer to an object of any type derived from base and still retain the correct type for the derived object. Also we can virtual functions to overcome object slicing. The use of implementing Textmessage* can be clearly understood in challenge 2.

Source: <https://www.geeksforgeeks.org/object-slicing-in-c/>

Explain how TextMessageCompare is used in the priority queue. Note that this is an example of operator overloading.

In c++ there is a feature called operator overloading which means that user can redefine the meaning of the operator for user defined types. The operator used here is () which allows to use any type and number of data. TextMessagecompare is used priority in priority queue. The operator will when the message is being pushed into queue. While pushing its going to compare the priority of the nodes and rearrange the queue.

<https://www.learncpp.com/cpp-tutorial/99-overloading-the-parenthesis-operator/>

One could argue that code is missing as we are allocating memory from the heap (that is, we use new). Explain why this is a problem.

By using **new** we are allocating memory from heap which is dynamically occupying the free memory available. Once the memory is occupied it has to be deleted if it is not deleted it will cause memory leak. The memory gets deallocated either by using **delete** or when the operation ends. If there is memory leak in an application which keeps on running then eventually all the memory will be occupied.

Problem 7.4

Explain again why using a pointer to instead of an object itself in a storage structure is beneficial. As explained before using pointer to store than object is very beneficial while storing data. First we restrict duplication of memory calling the object everytime. Secondly when using pointer the values stored can be easily altered from any point of the program. Also if the data is large it would be ideal to fetch the location of data instead copying the whole data into memory. So for large data it will also help in improving the performance. The memory resources allocated by object within a block will be there only in the block where it is created after the execution it gets destroyed where else by using pointers we can assign memory from heap and its deleted only when the delete is called.

Problem 7.5

Would you use join or a detach in the start method, considering the fact that in Simulator.cpp a getline is explicitly used? Explain your choice.

Detach would be used here. If we use join the calling thread will be stopped until the called thread finishes the task. By using detach we do not actually stop the main thread that is the calling thread but it allows both the threads to execute independently. Since getline is explicitly used the main thread waits for user to press enter.