A main server process is spawned initially that creates a process pool and a thread pool corresponding to it. Each thread is blocked on the accept call. The accept call in each thread is surrounded by a mutex to avoid the thundering herd problem and ensure that only one thread accepts any incoming connections. The child processes communicate to the parent process that they have exhausted their count of connections via pipes. Once the count of connections is reached, the child exits and signals the parent that spawns a new process and a pool of pthreads in it.
The Message queue is used as an ipc to store messages sent by clients in the server.
Shared memory stores a structured table as a mapping between names and client ids that will be used as mtype field to store messages from the clients in the message queue.This helps to avoid possible collisions by using hashing schemes. The shared memory is guarded by system V semaphores as shown below: