3221 Assignment 3

requirements checklist

- [ ] A3.2. The main thread in “New_Alarm_Cond.c”
- [ ] The main thread in “New_Alarm_Cond.c” will first determine whether the format of each alarm request is consistent with the formats specified above; otherwise the alarm request will be rejected with an error message. If a Message exceeds 128 characters, it will be truncated to 128 characters.
- [ ] INSERTION OF TYPE A ALARM REQUEST
- [ ] Assuming a thread is not running the requests, Check if alarm with the same Message_Number exists, and if it doesn't exist insert the new alarm into list sorted by Message_Number
- [ ] Assuming a thread is not running the requests, Check if alarm with the same Message_Number exists, and if it does replace it with the new alarm
- [ ] Ensure that when the alarm is replaced, the old alarm is freed from memory
- [ ] CREATION OF THREAD TO PROCESS ALARMS
- [ ] A.3.2.3. For each Type B alarm request received, if there does not exist any Type A alarm request with the specified Message Type in an alarm list, then the main thread will print the error message: “Type B Alarm Request Error: No Alarm Request With Message Type (Message_Type)!”
- [ ] A.3.2.4. For each Type B alarm request received, if there exists a Type B alarm request with the same Message Number in the alarm list, then the main thread will print the error message: “Error: More Than One Type B Alarm Request With Message Type (Message_Type)!”
- [ ] A.3.2.5. For each Type B alarm request received, if there exists a Type A alarm request with the same Message Type and there does not exist a Type B alarm request with the same Message Type in the alarm list, then the main thread will insert the Type B alarm request into the alarm list, and the main thread will print: “Type B Create Thread Alarm Request For Message Type (Message_Type) Inserted Into Alarm List at <time>!”.
- [ ] A.3.2.6. For each Type C alarm request received, if there does not exist any Type A alarm request with the same Message Number in an alarm list, then the main thread will print the error message: “Error: No Alarm Request With Message Number (Message_Number) to Cancel!”
- [ ] A.3.2.7. For each Type C alarm request received, if there exists a Type C alarm request with the same Message Number in the alarm list, then the main thread will print the error message: “Error: More Than One Request to Cancel Alarm Request With Message Number (Message_Number)!”
- [ ] A.3.2.8. For each Type C alarm request received, if there exists a Type A alarm request with the same Message Number and there does not exist a Type C alarm request with the same Message Number in the alarm list, then the main thread will insert the alarm request into the alarm list, in which all the outstanding alarm requests are placed in the order of their Message Numbers, and the main thread will print: “Type C Cancel Alarm Request With Message Number (Message_Number) Inserted Into Alarm List at <time>: <alarm_request>”..
- [ ] THE ALARM_THREAD FUNCTION
- [ ] A.3.3.1. On finding a new Type A alarm request in the alarm list, If a periodic_display_thread has been previously created, but all Type A alarms with the specific message types have all been processed (and may be removed), then terminate the periodic_display_thread responsible for displaying messages with the specified Message_Type, then print: “Type A Alarm Request Processed at <time>: Periodic Display Thread For Message Type (Message_Type) Terminated: No more Alarm Requests For Message Type (Message_Type). ”
- [ ] A.3.3.2. On finding a new Type B alarm request in the alarm list, the alarm_thread will immediately create a periodic_display_thread (specified in the next section) responsible for displaying messages with the specified Message_Type, then print: “Type B Alarm Request Processed at <time>: New Periodic Display Thread For Message Type (Message_Type) Created. ”
- [ ] A.3.3.3. On finding a new Type C alarm request in the alarm list, the alarm_thread will do : Remove the Type A alarm request with the specified Message_Number from the alarm list, then print: print: “Type C Alarm Request Processed at <time>: Alarm Request With Message Number (Message_Number) Removed”.
- [ ] A.3.3.3. On finding a new Type C alarm request in the alarm list, the alarm_thread will do: if all Type A alarm requests are removed as a result of A3.3.3 (a) then, terminate the periodic_display_thread. (if a periodic_display_thread responsible for displaying messages with the specified Message_Type has been created), then print: “Type C Alarm Request Processed at <time>: Periodic Display Thread For Message Type (Message_Type) Terminated: No more Alarm Requests For Message Type (Message_Type). ”
- [ ] A3.4. The “periodic_display_threads” in “New_Alarm_Cond.c”
- [ ] A3.4.2. If a periodic_display_thread, when periodically looking up a Type A alarm request with a specific Message Type in the alarm list, finds that a Type A alarm request which previously had that specific Message Type in the alarm list has changed to a different Message Type as a result of operation A3.2.2, then it will print: “Stopped Displaying Replaced Alarm

With Message Type (Message_Type) at <time>: <alarm_request> ”; Then it would stop perform the operation in A.3.4.1. for the alarm request which now has a different Message Type.