RTOS exercise

Rendez-vous with unique-ID server

In many situations you need an unique ID. Imagine we have system with many parallel threads, which from time to time needs an unique ID. We will provide a server in the system which will hand out unique ID's to the threads. The threads will perform a "rendez-vous" with the unique_ID-server thread from time to time and passing a pointer to where it wants the unique_ID back.

a) unique char server

Declare a request queue of pointers to char. The client task has in the global scope a variable for storing its unique char. Declare a char pointer to it and let it point to the variable. The contents of this pointer will be sent to the queue (notice QueueHandle_t type is xQueueHandle in our library). Make a unique_ID-server task receiving from the queue and assigning the received address to its own pointer to char for handing out a unique char. Make some (at least two) client tasks requesting unique chars from the unique_ID server. We want the client tasks to block in the rendez-vous – thus the queue size must be 1 and the priority of the unique_ID-server task must be higher than the clients.

b) two service entries

Some threads need unique integers and some threads need unique characters. Thus the unique_ID-server will have two "entries" the first one serves the unique-integer requests and the second one serves the unique-character requests. FreeRTOS queue sets would have been an ideal solution for this, but they are not present in our version of the freeRTOS-kernel. So instead of two queues (one for each service entry) we have to live with only one queue of requests. A request will now consist of a struct with an int (telling if a char or int is wanted) and a void pointer. Threads waiting for unique-integers will assign the void pointer in the struct to the pointer to int and then send it to the queue. The unique-ID server thread will read the int in the struct and decide to type-cast the void pointer back to a pointer to int or a pointer to char according to what service is needed. Take care of the fact that you only have approximately 28 unique characters and 9 unique one-digit integers.

c) bonus round

Only, if you have extra time: you may also implement the returning of unique-ID's to the unique-ID server - thus enabling reuse of unique-ID's.