You are to write an omegaRPC-based client-server application, which is an object server with locking. The server maintains a set of objects, which can be \textit{checked out} by clients, used by the client for a while and then \textit{released} by the client. While an object is checked out (but not released) by a given client, no other client can access it.

You are to implement the following remote procedure calls:

\begin{verbatim}
anyType checkOut(int32 oid); // lock and return the object
                        // specified by the oid
bool release(int32 oid,anyType); // update the object and release the lock
                        // return true if successful
                        // Each release must correspond to a
                        // checkout.
int32 create(anyType); // add a new object to the server,
                        // returning its oid
bool remove(int32); // delete object from server
\end{verbatim}

Your solution must satisfy the following criteria:

- When checking out an object, the checkOut may block at the server until the client which currently holds the object releases it. This should not block other clients from performing operations on the server.

- The server should handle an arbitrary number of objects, including classes which were designed after the server was compiled. You should test your design with at least two classes.

- You should not use the class definition at the server for any other reason than constructing the registry entry.

- Your solution should not deadlock. You can order the accesses to provide this property.