TODO

Near-term action items:

- **DONE:** OMWorld (external object to ObjectMonitor mapping): Combine LM_PLACEHOLDER into LM_LIGHTWEIGHT
  - The intention is that the OMWorld locking implementation will replace the Lightweigt locking. It will either go in with Lilliput or extracted into a separate, stand-alone RFE
  - Note: this means that the OMWorld implementation will still be used when the user runs with `-XX:-UseCompactObjectHeaders`
  - Owner: Axel

- **DONE:** Create a "patch queue" of all changes and maintain those patches as the top-most changes in https://github.com/openjdk/lilliput/tree/master
  - This will help
  - The "patch queue" will be rebased whenever the upstream (openjdk/jdk) is merged.
    - This will use non-destructive rebasing with "git merge -s ours", which "forgets" the old "patch queue"
  - Example patch queue:
    - * 4363dd8a1ef (HEAD -> lilliput-all, lilliput-stefank/lilliput-all) OMWorld Lilliput changes
    - * 916d0958a7f JDK-8325104: Lilliput: Shrink Classpointers
    - * 1346db1b546 Lilliput: Compact object headers
    - * 8fb10dc6f1d 8305898: Alternative self-forwarding mechanism
    - * 3726027944e 8305896: Alternative full GC forwarding
    - * c1f2dfb8735 OMWorld
    - * 998d0baab0f (tag: jdk-23+12, lilliput/jdk) 8324799: Use correct extension for C++ test headers
  - Owner: Roman

Workable items:
+ Lilliput Repository
| <= + OMWorld Lilliput changes (Make UseCompactObjectHeaders imply LM_PLACEHOLDER)
| <= Depends on OMWorld
| <= Roman has initial patch
| <= Stefan has a set on top of this
| <= Maybe fold this into 'Lilliput: Compact object headers'

| <= + JDK-8325104: Lilliput: Shrink Classpointers (TinyCP)
| <= Determine if 22-bit klass IDs are good enough
| | <= For Experimental UseCompactObjectHeaders
| | <= For productized UseCompactObjectHeaders
| <= Maybe: Figure out if UseCompressedClassPointer should == UseCompactObjectHeaders
| <= Figure out if hyperalignment is causing cache performance issues in anything other than microbenchmarks
| <= Review CDS parts: Runtime
| <= Review Metaspace parts: GC / Runtime
| <= Review C2 parts: Esp. klass_offset_in_bytes and related discussion

| <= + Cleanups after 'Lilliput: Compact object headers'
| | <= Go over the GC code and clean out "safe" klass fetches.

| <= + Lilliput: Compact object headers
| | Review

| <= + 8305898: Alternative self-forwarding mechanism
| | <= Clean out preserved_marks for 32-bit JVMs
| | <= Review

| <= + 8305896: Alternative full GC forwarding
| | Review

| <= + OMWorld
| | <= + Monitor deflation heuristic
| | | <= Rewrite
| | | <= Interaction with ConcurrentHashMap sizing
| | <= Decide C2 monitor cache layout
| | | <= should be loop unrolled or not
| | | <= size?
| <= Cleanup code used for various experiments (after evaluation)
| <= Full platform support?
| | <= Make non-primary platform call into Runtime until porters have added full support
| | <= PPC (optional)
| | <= RiscV (optional)
| | <= s390 (optional)
| | <= 32-bit arm (optional)
| | <= x86 (optional)

+ Testing (after "patch queue" has been created)
| <= + Stability testing

| <= + Performance testing