Postmortem support

Postmortem metadata are constants present in the final build which can be used by debuggers and other tools to navigate through internal structures of software when analyzing its memory (either on a running process or a core dump). Node.js provides this metadata in its builds for V8 and Node.js internal structures.

V8 postmortem metadata

V8 prefixes all postmortem constants with `v8dbg_`, and they allow inspection of objects on the heap as well as object properties and references. V8 generates those symbols with a script (`deps/v8/tools/gen-postmortem-metadata.py`), and Node.js always includes these constants in the final build.

Node.js debug symbols

Node.js prefixes all postmortem constants with `nodedbg_`, and they complement V8 constants by providing ways to inspect Node.js-specific structures, like `node::Environment`, `node::BaseObject` and its descendants, classes from `src/utils.h` and others. Those constants are declared in `src/node_postmortem_metadata.cc`, and most of them are calculated at compile time.

Calculating offset of class members

Node.js constants referring to the offset of class members in memory are calculated at compile time. Because of that, those class members must be at a fixed offset from the start of the class. That's not a problem in most cases, but it also means that those members should always come after any templated member on the class definition.

For example, if we want to add a constant with the offset for `ReqWrap::req_wrap_queue_`, it should be defined after `ReqWrap::req_`, because `sizeof(req_)` depends on the type of T, which means the class definition should be like this:

```
template <typename T>
class ReqWrap : public AsyncWrap {
 private:
  // req_wrap_queue_ comes before any templated member, which places it in a
  // fixed offset from the start of the class
  ListNode<ReqWrap> req_wrap_queue_;

  T req_;
};
```

instead of:

```
template <typename T>
class ReqWrap : public AsyncWrap {
 private:
  T req_;

  // req_wrap_queue_ comes after a templated member, which means it won't be in
  // a fixed offset from the start of the class
  ListNode<ReqWrap> req_wrap_queue_;
};
```

There are also tests on `test/cctest/test_node_postmortem_metadata.cc` to make sure all Node.js postmortem metadata are calculated correctly.

Tools and references

- llnode: LLDB plugin
- `mdb_v8`: mdb plugin
- nodejs/post-mortem: Node.js post-mortem working group