7.4 本章总结

从第6章中的基于锁的数据结构起，本章简要的描述了一些无锁数据结构的实现(通过实现栈和队列)。在这个过程中，需要小心使用原子操作的内存序，为了保证无数据竞争，以及让每个线程看到一个预制相关的数据结构。也能了解到，在无锁结构中对内存的管理是越来越难。还有，如何通过帮助线程的方式，来避免忙等待循环。

设计无锁数据结构是一项很困难的任务，并且很容易犯错；不过，这样的数据结构在某些重要情况下可对其性能进行扩展。但愿，通过本章的的一些例子，以及一些指导意见，可以帮助你设计出自己的无锁数据结构，或是实现一份研究报告中的数据结构，或用以发现离职同事代码中的bug。

不管在线程间共享怎么样的数据，你需要考虑数据结构如何使用，并且怎么样在线程间同步数据。通过设计并发访问的数据结构，就能对数据结构的功能进行封装，其他部分的代码就着重于对数据的执行，而非数据的同步。你将会在第8章中看到类似的行为：将并发数据结构转为一般的并发代码。并行算法是使用多线程的方式提高性能，因为算法需要工作线程共享它们的数据，所以对并发数据结构的选择就很关键了。