**关于LWIP**

LWIP使用邮箱和信号量来实现上层应用与协议栈间、下层硬件驱动与协议栈之间的信息交互。LWIP协议表面上模拟了TCP/IP协议的分层思想，但从实现上看，LWIP只是在一个进程内实现了各个层次的所有工作。

具体来说：LWIP完成相关初始化后，会阻塞在一个邮箱上，等待接收数据进行处理。这个邮箱的数据可能来自底层硬件驱动接收到的数据包，也可能来自应用程序。当在邮箱内取得数据后，LWIP会对数据进行解析，然后再依次调用协议栈内部上层相关处理函数处理数据，处理完成后，LWIP继续阻塞在邮箱上等待下一批数据。当然LWIP还有很多内存管理机制用来避免在各层间交互数据时大量的时间和内存开销。 另外，信号量也可以用在应用程序与协议栈的互相通信中，比如，应用程序要发送数据时，他先把数据发送到LWIP阻塞的邮箱上，然后它挂起在一个信号量上，LWIP从邮箱上取得数据处理后，释放一个信号量，告诉应用程序，要发送的数据已经发送完成了，最后，应用程序得到信号量继续运行，而LWIP继续阻塞在邮箱上等待下一批处理数据。

**关于ZYNQ中的LWIP协议栈**

首先有两种模式：RAW API和SOCKET API两种模式，前者可以裸机使用，后者需要依托于操作系统。（API（Application Programming Interface,[应用程序](https://baike.baidu.com/item/%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F)编程接口）是一些预先定义的[函数](https://baike.baidu.com/item/%E5%87%BD%E6%95%B0)，目的是提供[应用程序](https://baike.baidu.com/item/%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F)与开发人员基于某[软件](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6)或硬件得以访问一组[例程](https://baike.baidu.com/item/%E4%BE%8B%E7%A8%8B)的能力，而又无需访问源码，或理解内部工作[机制](https://baike.baidu.com/item/%E6%9C%BA%E5%88%B6)的细节。）