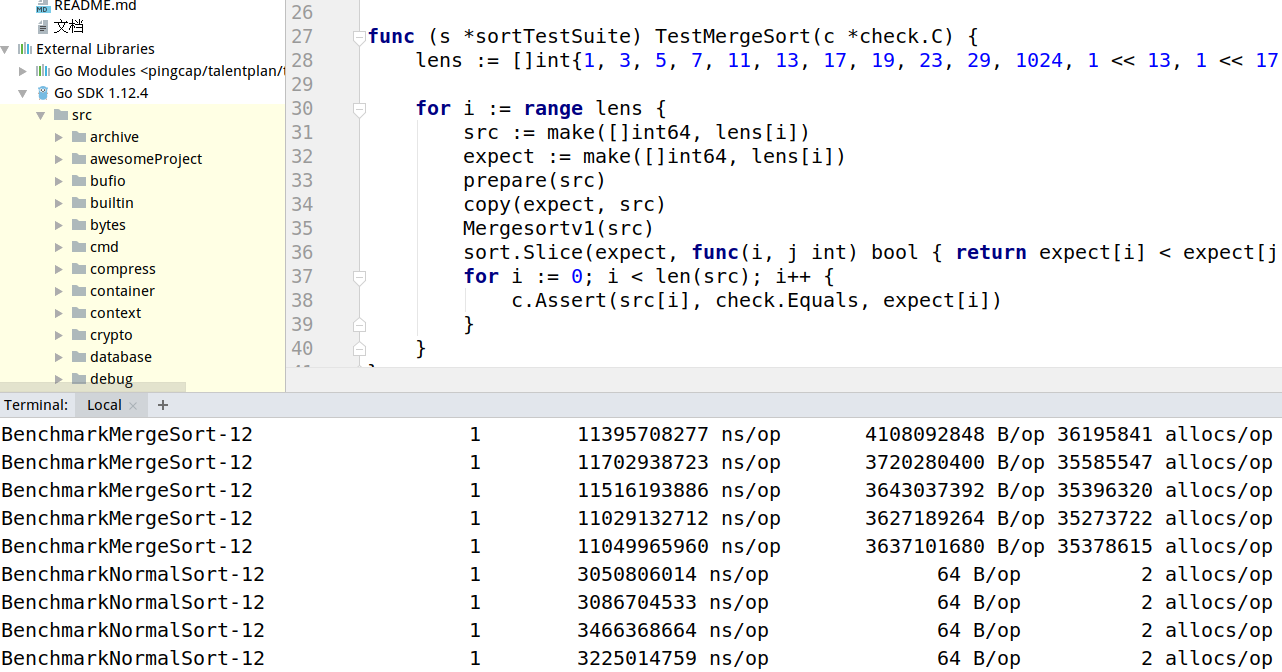
**V1版本**

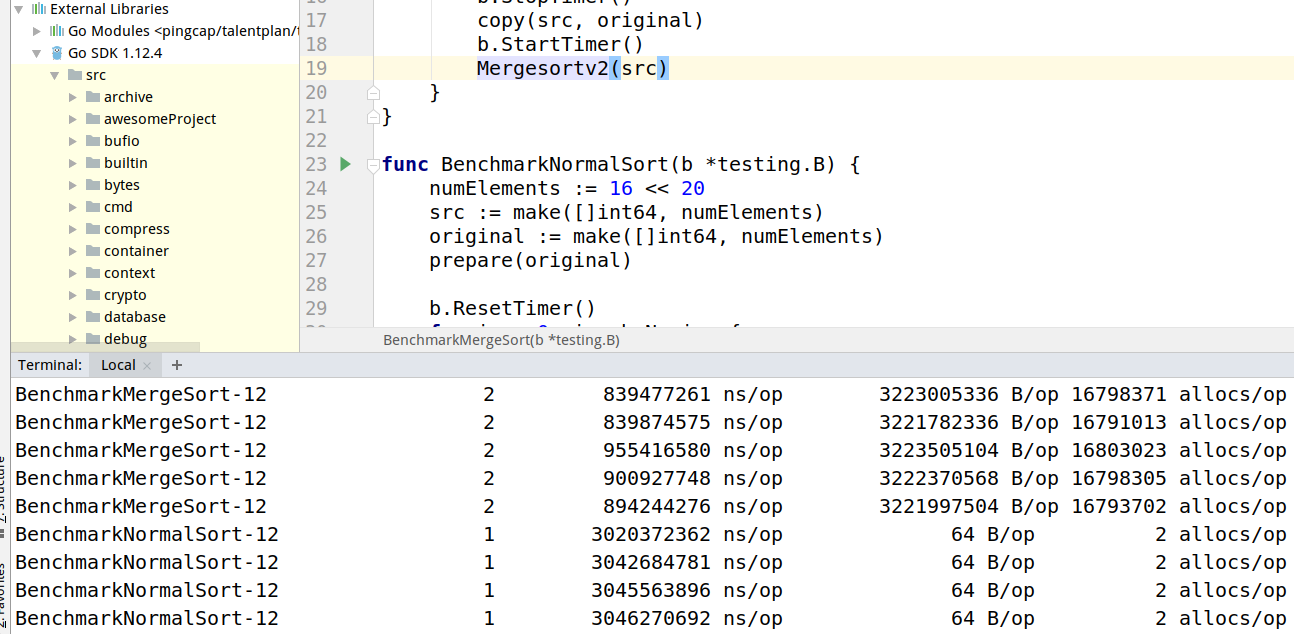
使用WaitGroup同时使用两个goroutine,并在goroutine执行结束之后开始merge。make bench 35.391s



运行时间是normalsort的三倍。

**V2版本**

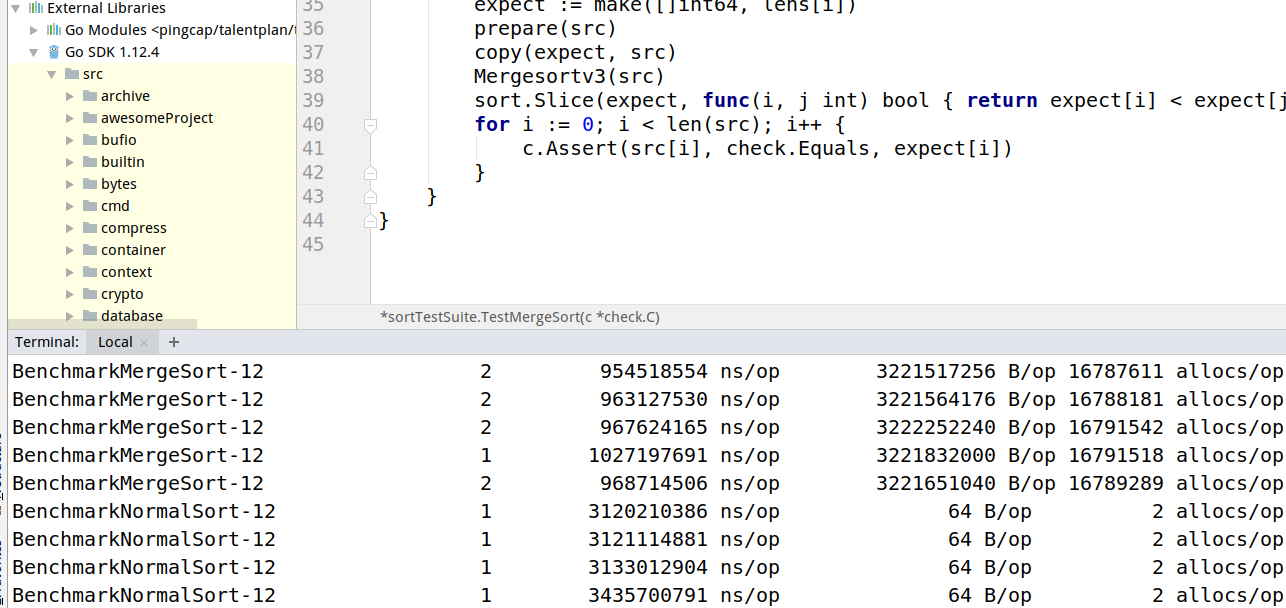
假如我有一个512个元素的切片，那么作用于mergesort的就有256，以此类推直到1个元素，但此时我们还要为它单独启一个goroutine,这就很浪费了，所以我们设置一个最大值max在小于max时使用循序实现（mergesort），在大于max时使用并发实现(v1)。经过多次测试max为1<<11,等于2048时，表现最佳。make bench 33.398s



运行时间比normalSort快。

**V3版本**

只生成一个goroutine处理调用，用主goroutine处理另外一个调用。make bench 33.840s



运行时间比normalSort快。比起V2无明显优化。