

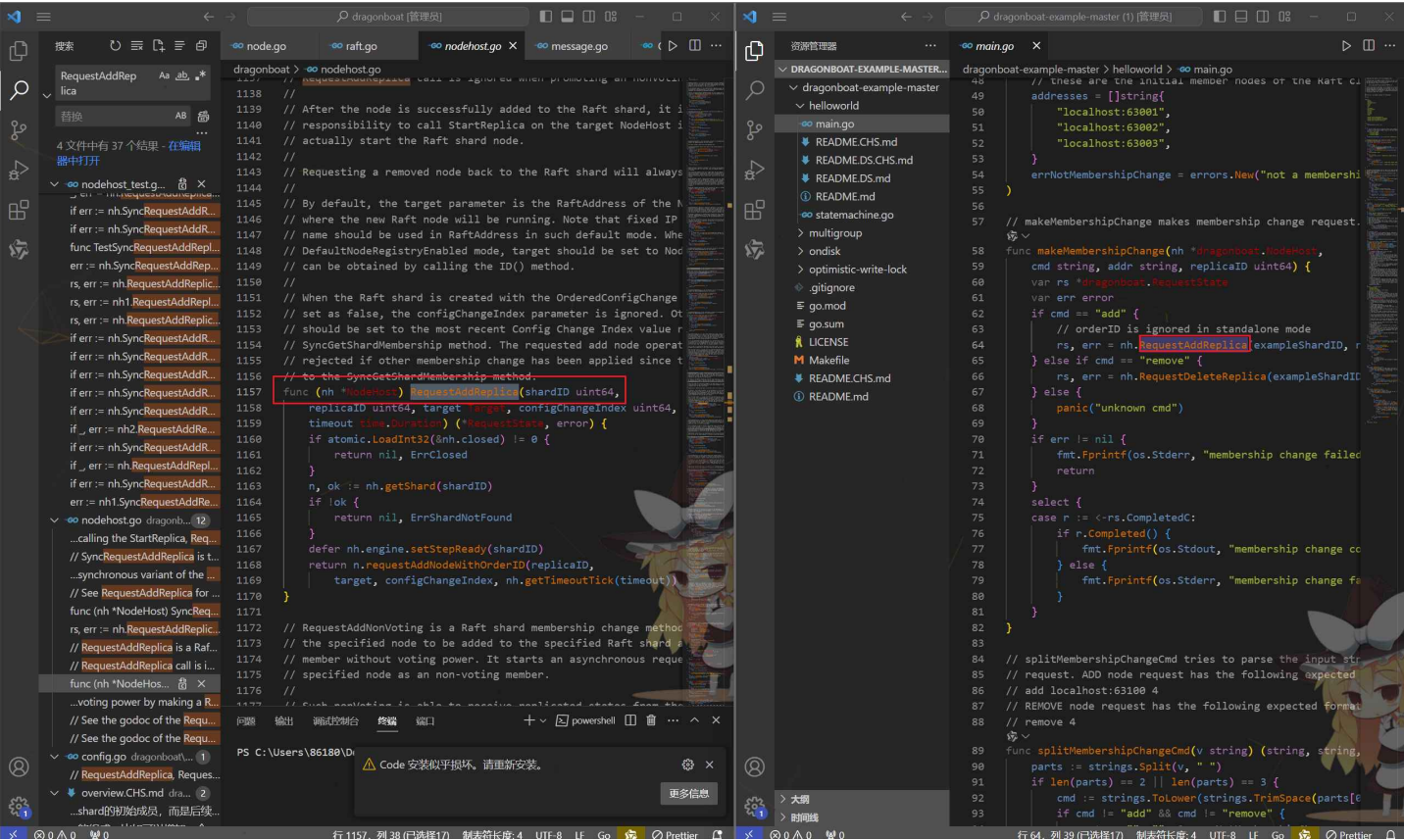
龙舟示例代码分析

示例代码中，主体为main.go和statemachine.go两个文件。

main.go含有main()入口函数，在里面我们实例化了一个NodeHost实例，把所创建的Raft集群节点加入了这个实例。它同时使用多个goroutine来做用户输入消息和Ctrl+C信号的处理。同时请留意MakeProposal()函数的错误处理部分代码和注释。

statemachine.go中，ExampleStateMachine结构体用来实现statemachine.IStateMachine接口。这个data store结构体用来实现应用程序自己的状态机逻辑。具体的内容将在[下一示例](README.DS.CHS.md)中展开。

RequestAddReplica函数可能可以尝试做更改、RequestDeleteReplica一起的，那自然有相应好些函数。



SyncPropose函数更新实例，应该起到更新信息的作用

```

func (nh *NodeHost) SyncPropose(ctx context.Context,
    session *client.Session, cmd []byte) (sm.Result, error) {
    timeout, err := getTimeoutFromContext(ctx)
    if err != nil {
        return sm.Result{}, err
    }
    rs, err := nh.Propose(session, cmd, timeout)
    if err != nil {
        return sm.Result{}, err
    }
    result, err := getRequestState(ctx, rs)
    if err != nil {
        return sm.Result{}, err
    }
    rs.Release()
    return result, nil
}

```

```

287 r {
288     select {
289     case v, ok := <-ch:
290         if !ok {
291             return
292         }
293         // remove the \n char
294         msg := strings.Replace(v, "\n", "", 1)
295         if cmd, addr, replicaID, err := splitMembershipChangeCmd(msg);
296             // input is a membership change request
297             makeMembershipChange(nh, cmd, addr, replicaID)
298         } else {
299             // input is a regular message need to be proposed
300             ctx, cancel := context.WithTimeout(context.Background(), 3
301             // make a proposal to update the IStateMachine instance
302             _, err := nh.SyncPropose(ctx, cs, []byte(msg))
303             cancel()
304             if err != nil {
305                 fmt.Fprintf(os.Stderr, "SyncPropose returned error %v\n
306             }
307         }
308     case <-raftStopper.ShouldStop():
309         return
310     }

```

怎么会有人代码里大小写写两个函数啊。。。

```
func (nh *NodeHost) Propose(session *client.Session, cmd []byte,
    timeout time.Duration) (*RequestState, error) {
    return nh.propose(session, cmd, timeout)
}
```

```
func (nh *NodeHost) propose(s *client.Session,
    cmd []byte, timeout time.Duration) (*RequestState, error) {
    if atomic.LoadInt32(&nh.closed) != 0 {
        return nil, ErrClosed
    }
    v, ok := nh.getShard(s.ShardID)
    if !ok {
        return nil, ErrShardNotFound
    }
    if !v.supportClientSession() && !s.IsNoOPSession() {
        panic("IONDiskStateMachine based nodes must use NoOPSession")
    }
    req, err := v.propose(s, cmd, nh.getTimeoutTick(timeout))
    nh.engine.setStepReady(s.ShardID)
    return req, err
}
```

```
func (e *engine) setStepReady(shardID uint64) {
    e.stepWorkReady.shardReady(shardID)
}
```