#ConditionalSync

```
go func() { // Inside accept()
  defer func() { shutdown <- true }()
  for { // Processes any messages received.
    select { // Guarded suspension.
    case msg := <-received:
      client.decodeAndExecute(msg)
    case sub := <-client.subscribe:
      username := client.getUsername()
      if sub.follower != username {
        continue
      }
      if sub.unfollow {
        client.Write("unfollow "+sub.followee)
      } else {
        client.Write("follow " + sub.followee)
      }
    }
  }
}()
<-shutdown // Wait for shutdown.
```

#Semaphore

```
client := wsClient{
    username:  make(chan string, 1),
…
}

// client.username is used as a semaphore.
go func() {
    client.username <- ""
}()

func (client *wsClient) getUsername()
(username string) {
    username = <-client.username
    client.username <- username
    return
}

func (client *wsClient) setUsername(username
string) {
    <-client.username
    client.username <- username
}
```