**Skynet注释**

1. **Skynet中lua服务的消息处理**
2. 服务初始化时,skynet.start函数会调用skynet.core.callback函数，重置消息处理函数指向lua-skynet的\_cb函数，该函数又会调用lua函数skynet.dispath\_message函数，并传入参数 @1消息type @2消息data @3数据length @4消息session @消息source
3. skynet.dispatch\_message函数会调用raw\_dispatch\_message函数，参数依次传入
4. **协程间的交互**
5. 服务初始化时，Skynet.start函数会调用skynet.timeout函数，push服务处理的第一条消息，消息类型为PTYPE\_RESPONSE，消息号为session。然后调用co\_create（func）创建一个协程，func指向服务实际的消息处理函数。消息号session和消息处理协程co保存为kv对session\_id\_coroutine[session] = co，当服务处理到消息号为session的消息时，调用suspend(co,coroutine\_resume(co,session,source,msg,sz))的方式唤醒协程，并传入参数。
6. 服务处理非PTYPE\_RESPONSE类型的消息时，会根据类型选取对应的proto，该对象包含服务对于该种消息的pack、unpack、消息处理函数。然后把消息处理函数传递给一个空闲协程（如无空闲则新建），使用两个table session\_coroutine\_id、session\_coroutine\_address分别保存协程对应处理消息的session及消息source。调用suspend(co, coroutine\_resume(co, session,source, p.unpack(msg,sz)))函数唤醒协程并传递参数。
7. 消息处理结束后，协程保存到一个局部table coroutine\_pool中，以备后用。并向主协程返回”EXIT”命令，用于清理。
8. “CALL”命令：表示执行消息处理的协程未执行完毕，暂且保存，消息号为session\_id\_coroutine[session] = co，等待其他服务完成操作后，会返回一条消息唤醒。
9. “SLEEP”命令：
10. **Skynet函数**
11. skynet.call(addr, typename, ...):消息发送，需要目标服务响应。向目标服务@addr发送类型为@typename的消息，变参为消息数据，调用yiled\_call执行等待。
12. yiled\_call(service,session):挂起协程。返回“CALL”命令，挂起并保存协程，等待消息响应。响应的消息类型为PTYPE\_RESPONSE。
13. skynet.send(addr, typename, ...):消息发送，无需响应。
14. skyent.wait(co):挂起协程并等待。逻辑为分配一个session，并调用“SLEEP”命令，保存session-co对，sleep\_session[co]=session，等待调用skynet.wakeup(co)。
15. skynet.wakeup(co):唤醒协程。在wakeup\_session表中标记co状态为可唤醒。每次消息处理结束，会调用一次dispatch\_wakeup函数。
16. dispatch\_wakeup():从wakeup\_session表中依次取出挂出的co，通过suspend(co, coroutine\_resume(co, false, "BREAK"))唤醒。
17. skynet.newservice(name, ...):通过skynet.call启动服务。具体过程如下：实例，A服务调用该函数启动服务B

* A调用skynet.call(".launcher", "lua" , "LAUNCH", "snlua", name, ...)，向launcher服务发送一条"LAUNCH"命令，等待响应。
* Launcher服务通过command.launch函数处理该消息。
* command.launch函数调用launch\_service函数。
* launch\_service函数调用skynet.launch函数启动服务B，调用skynet.response函数取得响应函数并保存。
* 服务B启动后，会想launcher服务发送一条”LAUNCHOK”命令
* Laucher服务执行command.LAUNCHOK命令，通过保存的响应函数向服务A返回响应信息，即服务B的handle值

1. skynet.ret(msg,sz)：调用“RETURN”命令，返回msg，sz。
2. skynet.response(pack):调用“RESPONSE”命令，命令执行时，会取出消息源服务，返回一个响应函数，该函数执行时，会把数据发送到数据源服务中。