**Go message struct**

**1.数据槽(默认数据页容量GROOVE\_CAP = 1024)**

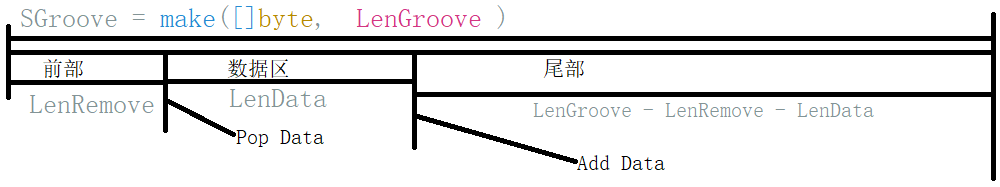
：SGroove []byte 切片，数据槽

：LenGroove int 数据槽总长度

：LenRemove int 数据槽取掉数据（前部）的长度

：LenData int 数据槽当前数据长度

：shortFlag int 是否需要缩短数据槽标志位



**2.数据槽变换（增长或者缩短）模式**

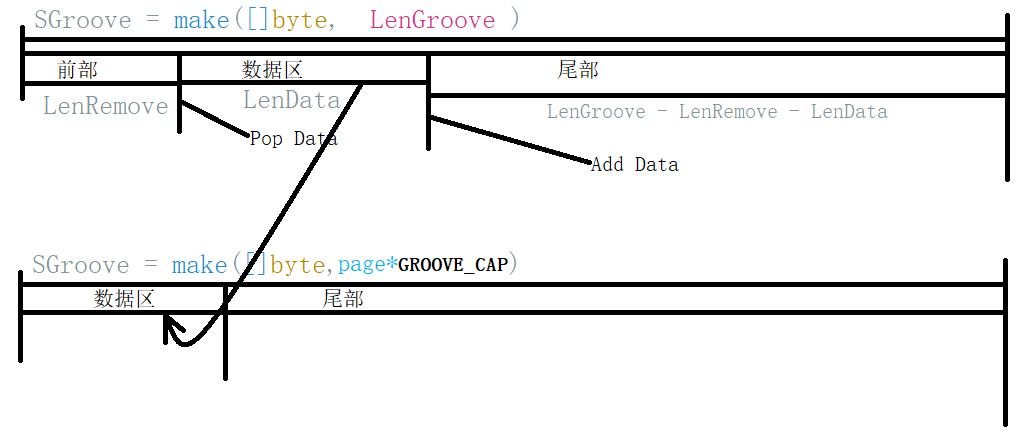
A 数据槽扩容迁移（前部变为0）

当数据槽要增加的数据足够多，原始数据槽已经不满足要求，需要按页扩容。

扩容页：page = LenData + len(AddData) /GROOVE\_CAP + 1

a.新开辟一个 page 页数据槽

b.对数据进行拷贝迁移操作（操作3.A）

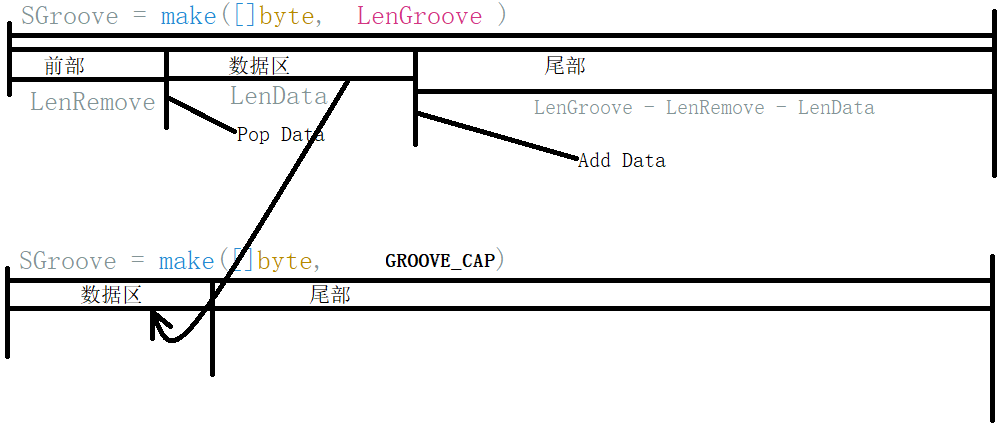


B 数据槽回根迁移（前部变为0）

当数据槽超过2页后，如果后续每次新增数据加入数据槽都不超过1页（通常情况下数据都会被取干净，或者残留很少），规定次数（比如 5次）后缩短为1页，动态调整数据槽。

a.新开辟一个单页数据槽

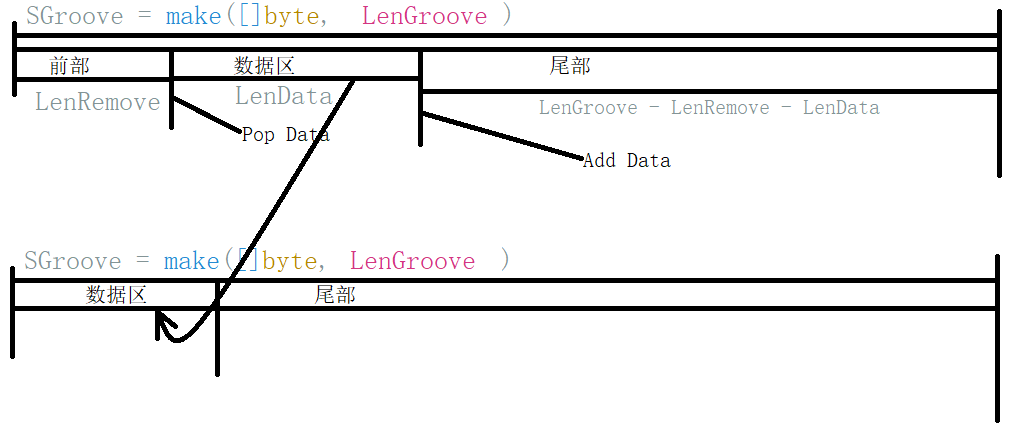
b.对数据进行拷贝迁移操作（操作3.A）



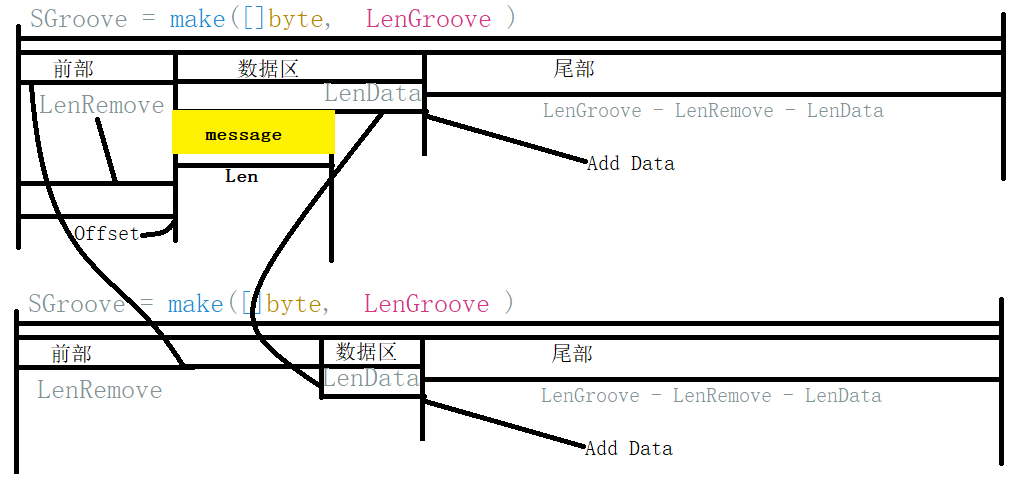
**3.数据槽数据操作**

A 迁移数据（前部变为0）

对数据区的数据进行迁移，把数据区拷贝到槽（或者新槽）顶端，令LenRemove = 0



B 读出数据

对数据区数据进行读取，如果数据区LenData 足够读出一个完整消息包，读取成功后LenRemove = LenRemove - len(AddData)；LenData =LenData - len(AddData)。如果读完数据发现数据槽已经没有数据可读，那么下次写入数据可以从头开始写入，如果LenData = 0，那么令LenRemove = 0

B 写入数据

a.当数据槽尾部足够追加数据

数据添加在数据区后面，同时 LenData =LenData + len(AddData)

b.当数据槽尾部不足够追加数据，但是数据槽的前部和尾部可以足够追加数据，那么

先进行数据迁移操作（操作3.A）

正常写入数据（操作3.B.a）

c.当数据槽的前部和尾部都不足够追加数据，那么需要先扩容，后迁移，最后正常追加。

先扩容迁移操作（操作2.A）

正常写入数据（操作3.B.a）

D 数据槽回根判断

a.如果数据槽超过两页，进行写入数据时进行是否缩短数据槽登记

如果原有数据和要写入的数据总共都不超过单页最大数据长度GROOVE\_CAP，对数据槽标志位shortFlag加一，如果规定次数（比如 5次）内都不超过单页最大数据长度GROOVE\_CAP，那么

对数据槽进行数据槽回根迁移（操作 2.B）

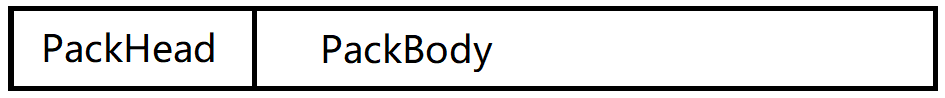
正常写入数据（操作3.B.a）

b.如果规定次数（比如 5次）内有任何一次超过单页最大数据长度GROOVE\_CAP，那么数据槽标志位shortFlag重置为0，重新计数,下次数据写入继续做数据槽回根判断（操作3.D）

**4.消息包message（XCBB）结构**

：PackHead 消息包头

：PackBody 消息内容



A 数据包头

PackHead 因为消息包包头是固定格式（共13 Byte），因此提炼出来单独封装

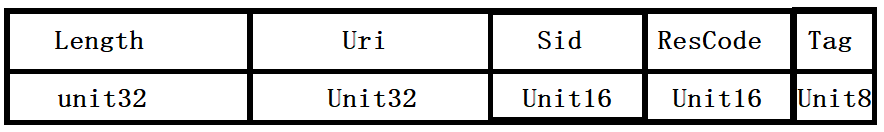
: Length uint32 整个包的长度

: Uri uint32 数据包结构编号

: Sid uint16 房间号（无意义，但可看到服务器当天累计开启几次房间），默认0

: ResCode uint16 错误码，消息正常是 200，默认200

:Tag uint8 无意义，默认是1



B 消息包体

PackBody 对于消息体，每个都有差异，因此只提供数据流模式

: Body []byte 切片，数据源

: Offset int 消息数据偏移量（数据起点位）

: Len int 消息数据长度

C压包（Encode）

每次数据槽写入数据，尾部第一个数据都是包头，包头偏移量Offset = LenRemove + LenData ，数据长度是 len(PackHead)

添加body，根据包头的Uri，对应相应压包流程，对消息message进行压包

压包偏移量Offset = LenRemove + LenData + len(PackHead)，数据长度是 Length - len(PackHead)

D解包（Decode）

每次数据槽读出数据，数据区第一个数据都是包头，包头偏移量Offset = LenRemove，数据长度是 len(PackHead)

添加body，根据包头的Uri，对应相应解包流程，对消息message进行解包操作，

解包偏移量Offset = LenRemove + len(PackHead)，数据长度是 Length - len(PackHead)

