TCM跨进程通信协议

用于全双工信道tcm::Pipe

1. 信道区分服务端和客户端，数据载体由服务方提供。

信道功能分化发生在连接方式确定时，使用Listen方法后自动将自身作为服务端。使用Connect方法后自动将自身作为客户端。

1. 信道具有一个内存映射区段，由相邻的5部分组成。

以下将“发送”方向作为基准方向。

第1部分是公共段，用于广播信道公共状态。

第2部分是S2C协议段，用于服务端到客户端的通信状态控制。

第3部分是C2S协议段，用于客户端到服务端的通信状态控制。

第4部分是S2C数据段，用于服务端到客户端的数据。

第5部分是C2S数据段，用于客户端到服务端的数据。

当前分段方案的优点：数据段长度的变化不影响协议段的分析。

1. 公共段共12字节。

公共段的第1字节是信道开关状态，0x00为关闭，其他为开放。

公共段的其他字节保留用于扩展。

1. 每个协议段共10字节。

协议段的第1字节是传送通知。0x01表示已发送，0x00表示已读。

协议段的第2～5字节（共4个字节）是有效数据长度。

协议段的其他字节保留用于扩展。

1. 发送方发送数据需要在对应的数据段，写入完成后需要将通知置为发送。

接收方接收数据需要在对应的数据段，读取完成后需要将通知置位已读。

TCM Xproc协议

宿主程序解析任务描述文件后，根据不同的Xproc扩展模式，需要产生对应的响应。

1. Lua脚本扩展模式

Lua脚本扩展模式仅能用于TCM Host

在这种模式下，开发人员需要为组件准备Lua脚本。

脚本中必须具备以下函数：

TcmBeforeRun { param null ; return int }

TcmRunning { param [stage] , [prog] ; return int }

TcmAfterRun { param [retcode] ; return null }

这些函数分别用于任务阶段A、B和C（运行前、运行中和运行后）

TcmBeforeRun 的返回值表示任务开发前是否取消任务，0表示不取消

TcmRunning 的输入参数为运行状态和进度，返回值表示控制码

TcmAfterRun 的输入参数为执行器返回值

在Lua脚本中可以使用以下数据：

tcm\_param 为任务参数，具体的索引由任务描述文件映射

tcm\_tag 为任务标签，具体的索引由任务描述文件映射

tcm\_debug 为调试信息，目前仅支持exception字段（用于获取组件异常的详细信息）

1. Pipe扩展模式

Pipe扩展模式可以用于TCM Host和TCM xHost

具体的响应流程如下：

1. 调用方需要以监听方式启动Pipe，生成任务描述文件，调用宿主并等待宿主消息。
2. 宿主启动，解析任务描述文件，加载组件，会以连接方式启动Pipe。
3. 宿主会向调用方发送单字节 ’A’（1 byte），表示执行任务阶段A，然后等待调用方回执阶段A执行状态。
4. 调用方接收到 ’A’，执行运行前操作，根据实际情况返回取消指令，0表示不取消。

然后监听任务阶段B的消息。

1. 宿主确认调用方不取消任务时，开始运行任务的功能。

持续发送 ’B’（1 byte）+ state（4 bytes）+ prog（4 bytes）

1. 调用方接收消息后，反馈控制码

当state转换成IDLE后，等待宿主发送’C’

1. 宿主完成任务，发送’C’（1 byte）+ response\_xml（n bytes），等待调用方接收完成
2. 调用方接收到’C’，执行运行后操作
3. 宿主监测到调用方已接收，断开连接，退出

response\_xml:

<response>

<params>

<param id=”{0}”>[opt:CDATA]{1}</param>

[REPEAT: param]

</params>

<debug>

<exception>[opt:CDATA]{2}</exception>

</debug>

</response>

{0}: 参数的标识，整数

{1}: 参数的值，字符串，可选使用CDATA，要求符合XML基本规范

{2}: 异常内容，字符串，可选使用CDATA，要求符合XML基本规范

1. 回调TCM组件模式