1. **协议总览**
2. 主机内通信，统一使用domain-socket模拟TCP方式
3. 跨SOC通信，使用短链接TCP方式
4. 协议所述字段字节序均为**小端**
5. 协议所述对齐方法均为1字节对齐
6. **包头定义**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 长度(字节) | 用途 | 备注 |
| OP | I32 | 4 | 协议起始标志 | 固定值小端’Hdtp’, 寓意为:Hik Daemon Transport Protocol |
| LEN | U16 | 2 | 数据包长度 | 数据包不含包头长度 |
| SEQ | U16 | 2 | 信令序列号 | 用于强制同步, 无需应答的请求可以填0 |
| TYPE | I32 | 4 | 协议基本类型 | 请求类型遵从(三)之定义，应答类型在(三)之定义基础上二进制或0x80000000—最高位置1 |
| ERR | I32 | 4 | 应答错误码 | 负数为错误，0为正常，正数为警告，尽量保持和errno含义一致 |
| PAYLOAD | [] | LEN | 数据实体 |  |

1. **协议节**

协议节(即：包头所示PAYLOAD字段中的全部数据) 全部以TLB（Type-Length-Block）方式分段描述, 每一个TLB的整块数据称之为节(SECTION), 节的整体结构布局如下

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 长度(字节) | 用途 | 备注 |
| CATEGROY | U16 | 2 | 节数据类型 | 预设的，具备含义的类型标记符，类似于json中的key |
| SECT-LEN | U16 | 2 | 节数据长度 | 不含节头的数据长度 |
| PAYLOAD | [] | SECT-LEN | 数据实体 | 类似键值模式中的value |

1. **业务协议区间(该区间同时约束TYPE和CATEGROY的定义区间)**

[0,0xFF]段: 保留，协议自身调节用，或固定的可引用字段

[0x100,0x1FF]段: creator 相关业务

[0x200,0x2FF] 段： davinci 相关业务

[0x300,0x3FF] 段: daemon主从通信业务

1. **业务类型(包头TYPE)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 二进制 | 业务用途 | 预设参数字段 | 数据流向 |
| LOGON | 0 | 登入服务端 | PID  PSNAME | C->D |
| LOGOFF | 1 | 登出服务器 | PID | C->D |
| KEEPALIVE | 2 | 心跳 | TIMESTAMP\_MONOTONIC |  |
|  |  |  |  |  |
| CREATOR\_FAULT\_REPORT | 0x100 | Creator上报故障信息给daemon | FAULT\_CODE  TIMESTAMP\_EPOCH  FAULT\_REASON | C->D |

1. **节类型定义表(节CATEGROY)**

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 二进制数据 | 作用 | 类型 |
| PID | 0 | 进程ID | U32 |
| PSNAME | 1 | 进程名 | string |
| TIMESTAMP\_MONOTONIC | 2 | 时间戳(墙钟,100ns) | U64 |
| TIMESTAMP\_EPOCH | 3 | 时间戳(纪元,100ns) | U64 |
|  |  |  |  |
| FAULT\_CODE | 0x100 | 故障代码 | I32 |
| FAULT\_REASON | 0x101 | 故障描述 | string |
|  |  |  |  |

蓝色所示部分为creator和daemon具体通信协议的字段定义区域，请帮忙完善

目前这个设计思路，参考json的方式，以key-value对作为存储和传输媒介，扩展性类似于json, 但是解析复杂度低于json,可以保证:

1. 不因为缺少字段而触发解析错位
2. 不会发生数据类型失配

比较麻烦的是，需要针对各个字段，或者是某些恒定的组合进行类型定义

不过这个工作量，并不比将当下工作转换为json来得重

同时，不一定每个简单字段都必须定义一个节，有一些比较常用且固定的结构，也可以定义为一个节

1. **示例**

举例： 当发生进程崩溃，creator报告此信息给daemon报文如下:

48 64 74 70 (OP)

00 00 (ERR)

01 00 (SEQ)

00 01 (TYPE:CREATOR\_FAULT\_REPORT)

24 00 (LEN)

00 01 (CATEGROY:FAULT\_CODE)

04 00 (SECT-LEN)

00 00 00 11 (SECT-DATA)

00 01 (CATEGROY: TIMESTAMP\_EPOCH)

08 00 (SECT-LEN)

66 cb 01 99 26 01 00 00(SECT-DATA)

01 01 (CATEGROY:FAULT\_REASON)

0B 00 (SECT-LEN)

6d 65 6d 6f 72 79 20 6c 65 61 6b 00 (‘memory leak’)