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 |
| 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 |
| 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 | U116 | 2 | 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 |
| LEN | U16 | 2 | 数据包长度 | 数据包不含包头长度 |
| PAYLOAD | [] | LEN | 数据实体 |  |

1. **基本协议类型**

下表所示为所有合法的协议基础类型，用于填写包头TYPE字段

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 二进制数据 | 作用 | 标签 |
| LOGIN | 0 | 登入服务器 | 客户端发起  SYN后第一包 |
| LOGOFF | 1 | 登出服务器,通常可用于要求服务端主动关闭链接的场景 | 客户端发起  可选 |
| QUERY | 2 | 查询请求，需要服务端端给出数据应答 | 客户端发起 |
| INFORM | 3 | 通知请求，不需要给出应答 | 客户端发起 |
| INTERACT | 4 | 交互请求，对端可以给出应答，也可以不给出应答 | 不限发起方 |
| PUSH | 5 | 推送消息，不需要给出应答 | 服务端发起 |
| KEEPALIVE | 6 | 心跳保活 | 可选  不限发起方 |

备注:

1. 以发起方为维度， 严格区分不需要应答的协议类型， 可以有利于较大规模的互联后， 点到点通信无法区分CS的问题
2. 暂不考虑支持服务端发起推送，又要求客户端必须应答的场景
3. **协议体组成**

协议体(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 | 设定一些必备的类型+各种业务的具体组合 |
| 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 |  |

1. **CATEGROY分段**

0段: 保留，协议自身调节用，业务不使用

0x100段: creator 相关业务

0x200 段： davinic 相关业务

0x300 段: daemon主从通信业务

1. **0段明确占位符**

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 二进制数据 | 作用 | 类型(长度) |
| PID | 0 | 描述主调进程PID | LONG(4/8) |
| PSNAME | 1 | 作用进程名 |  |

1. **creator段占位符**

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 二进制数据 | 作用 | 类型(长度) |
| FAULT-REPORT | 0x100 | 故障信息上报 |  |
| REPAIR-ACTION | 0x101 | 修复动作下发 |  |

1. **示例**

举例组成一个客户端发出的命令包, 模拟传输如下结构体:

{

int reboot; // 要求重启

char reason[0]; // 原因描述

}

业务给定的“重启”业务CATEGROY假设为 0x101

报文如下:

48 64 74 70 (OP)

00 01 (SEQ)

00 00 (ERR)

00 03 (INFORM)

00 14 (LEN)

01 01 (CATEGROY)

00 10 (SECT-LEN)

00 00 00 01 (REBOOT)

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