

## 一、libhv安装

### 下载

git clone <https://gitee.com/libhv/libhv.git>

### 编译安装

<https://hewei.blog.csdn.net/article/details/113704737>

进入到libhv目录下，执行以下三个命令

对于类Unix系统平台来说，推荐使用 **Makefile三部曲**

```
1 | ./configure
2 | make
3 | sudo make install
```

### 链库与使用

一般情况下就可以直接使用libhv网络库了，如果无法导入，请参照下面的文档修改

<https://hewei.blog.csdn.net/article/details/113706378>

## 二、Hiredis安装

为了配合libhv网络库，hiredis中有专门为libhv封装的一些接口

### 下载hiredis.h

git clone <https://github.com/redis/hiredis>

### 配置环境

进入hiredis目录

```

1 # 进入 hiredis 目录
2 cd hiredis
3 # 执行 make && make install
4 sudo make
5 sudo make install
6 # 如果权限不够先执行 su 指令
7 su

```

以 redis.cpp 为例，编译命令如下

```
g++ redis.cpp o redis -/usr/local/lib -lhiredis
```

在执行的时候如果出现动态库无法加载，那么需要进行如下配置在/etc/ld.so.conf.d/ 目录下新建文件 usr-libs.conf，内容是 /usr/local/lib

```
sudo vim /etc/ld.so.conf.d/usr-libs.conf
```

然后使用命令 sbin/ldconfig 更新一下配置即可

使用后，如果认为 Hirdis 使用起来较为繁琐，可以根据自己需求进行封装或用其他方式与 Redis 进行交互。

### 三、libhv目录解析

examples: 一些示例文件，各个用例的源文件

include: 各个用例的头文件

lib: 存放了一个静态库、一个动态库，后面调用直接从这里拿就可以

bin: 用例的可执行程序存储在这里

项目用到的示例是 protorpc

**编译一下 protorpc**

```
make protorpc
```

进入libhv/examples/protorpc目录，找到protorpc的实例文件

```

weihong@weihong:~/libhv/examples/protorpc$ ls
generated handler proto protorpc.c protorpc_client.cpp protorpc.h protorpc_server.cpp router.h
weihong@weihong:~/libhv/examples/protorpc$

```

一个客户端 一个服务器

libhv解析proto的结构体

由一个头部结构体和一个body组成，头部是一个12字节的结构体

```

typedef struct {
    protorpc_head head;
    protorpc_body body;
} protorpc_message;

```

**头部结构**

```

// protocol:4bytes + version:1byte + flags:1byte + reserved:2bytes + length:4bytes = 12bytes
#define PROTORPC_HEAD_LENGTH 12
#define PROTORPC_HEAD_LENGTH_FIELD_OFFSET 8

```

```
#define PROTORPC_HEAD_LENGTH_FIELD_BYTES    4
typedef struct {
    unsigned char    protocol[4];
    unsigned char    version;
    unsigned char    flags;
    unsigned char    reserved[2];
    unsigned int      length;
} protorpc_head;
```