# 项目交接说明文档

## 环境说明

编译环境：172.16.10.201

代码位置/home/lxj/work/Thumbs/tcpserver

/home/lxj/work为挂载的远程目录。如果201主机重启，需要以root运行/home/lxj/mount.sh脚本进行挂载。

项目位置：E:\workspace：//172.16.10.98/workspace 可通过smbuser/smbuser来进行远程访问。

## tcpserver基本情况

以下的几个项目基于nginx tcpserver开发。tcpserver是基于tcpproxy改造的一个tcp服务器插件。核心代码在E:\workspace\Thumbs\tcpserver\src\tcpserver目录下。该目录下有一些子目录，是相应的业务模块：

qmv 是 QMV+项目

msgstorage 是 消息中心存储服务

msgonline 是 消息中心在线服务

stat 是 统计系统的，已经废弃。

thumb 是缩略图上传服务的，用于云播放的缩略图。

proxy 是 tfs上传代理服务。

目前正在开发中的项目：

### qmv+项目

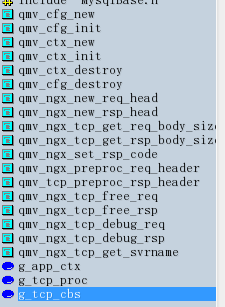
代码为：E:\workspace\Thumbs\tcpserver\src\tcpserver\qmv

config-qmv config用的配置文件。

Makefile 为测试工具的编译用，不是本服务编译用的。

qmv\_protocol.h 一些协议头，指令，请求响应消息相关的定义

qmv\_impl.h|.cpp 是实现了tcpserver的一些回调，包含：



qmv\_cfg\_\* 配置解析。

qmv\_ctx\_\* 上下文初始化。

其它一些函数是实现相关协议解析的回调。这些回调，都被设置到g\_app\_ctx, g\_tcp\_proc,g\_tcp\_cbs这计个函数指针上了。

g\_app\_ctx 是设置配置回调，及上下文初始化相关回调

g\_tcp\_proc 是设置业务处理的回调。

g\_tcp\_cbs 是设置协议解析相关的回调。

qmv\_service.cpp 是处理 具体业务请求的相关实现。

其中 qmv\_proc 是主方法，被设置到g\_tcp\_proc函数指针上了。其中会根据具体的命令字调用相应的业务处理函数。

qmv\_pub.h|.cpp 是访问数据库的相关实现。

qmv\_client.h|.cpp Testqmv.cpp 是测试代码。

config命令：

./configure --prefix=/usr/local/qmvsvr --sbin-path=sbin/qmvsvr --builddir=qmvsvr-objs \

--conf-path=conf/qmvsvr.conf --add-module=src/tcpserver --mod-config=qmv/config-qmv \

--with-debug --without-pcre --without-http

其中有安装位置，配置文件名。编译目录为：qmvsvr-objs。config之后，如果没有添加的新的源文件，没有修改编译用的配置文件(qmv/config-qmv) 不用每次重新config。

编译命令：

make –f qmvsvr-objs/Makefile

编译完成后，qmvsvr-objs下会有一个名为nginx的程序。

运行：

可以直接运行编译目录下的程序：

qmvsvr-objs/nginx –t 测试配置文件。#也可用于查看当前使用的是哪个配置文件。

qmvsvr-objs/nginx 运行。

或者

cp qmvsvr-objs/nginx /usr/local/qmvsvr/sbin/qmvsvr

然后执行：

/usr/local/qmvsvr/sbin/qmvsvr

日志文件在：/usr/local/qmvsvr/logs/ 目录下。

### 消息存储msgstorage

代码为：E:\workspace\Thumbs\tcpserver\src\tcpserver\msgstorage

config-msgstorage config用的配置文件。

msg\_protocol.h 一些协议头，指令，请求响应消息相关的定义

msg\_impl.h|.cpp 是实现了tcpserver的一些回调，包含：

msg\_cfg\_\* 配置解析。

msg\_ctx\_\* 上下文初始化。

其它一些函数是实现相关协议解析的回调。这些回调，都被设置到g\_app\_ctx, g\_tcp\_proc,g\_tcp\_cbs这计个函数指针上了。

g\_app\_ctx 是设置配置回调，及上下文初始化相关回调

g\_tcp\_proc 是设置业务处理的回调。

g\_tcp\_cbs 是设置协议解析相关的回调。

msg\_service.cpp是处理 具体业务请求的相关实现。msgstorage\_proc是主方法，里面根据具体的命令字请求相关业务处理方法。

mongo\_dao/SequenceMgr.h|.cpp 生成，维护mongo序列的类。

mongo\_dao/msg\_dao.h |.cpp 操作mongo db的相关函数。

mongo\_dao/mongo\_pool.h mongo连接池封装类。

mongo\_dao/BCBitmap.h|.cpp 广播用的位图实现类。

client 目录下为测试程序。

config命令：

./configure --prefix=/usr/local/msgstorage --sbin-path=sbin/msgstorage --builddir=msgstorage-objs \

--conf-path=conf/msgstorage.conf --add-module=src/tcpserver --mod-config=msgstorage/config-msgstorage \

--with-debug --without-pcre --without-http

# config的命令在 tcpserver根目录下有个config.sh中都有记录。

编译：

make –f msgstorage-objs/Makefile

编译完成后，msgstorage-objs目录下会生成一个nginx程序。

运行：

可以直接运行编译目录下的程序：

msgstorage-objs/nginx –t 测试配置文件。#也可用于查看当前使用的是哪个配置文件。

msgstorage-objs/nginx 运行。

或者

cp msgstorage-objs/nginx /usr/local/msgstorage/sbin/msgstorage

然后执行：

/usr/local/msgstorage/sbin/msgstorage

日志文件在/data/msgstorage目录下

### 消息中心在线服务

代码为：E:\workspace\Thumbs\tcpserver\src\tcpserver\msgonline

config-msgonline config用的配置文件。

online\_protocol.h 一些协议头，指令，请求响应消息相关的定义

online \_impl.h|.cpp 是实现了tcpserver的一些回调，包含：

online\_cfg\_\* 配置解析。

online\_ctx\_\* 上下文初始化。

其它一些函数是实现相关协议解析的回调。这些回调，都被设置到g\_app\_ctx, g\_tcp\_proc,g\_tcp\_cbs这计个函数指针上了。

g\_app\_ctx 是设置配置回调，及上下文初始化相关回调

g\_tcp\_proc 是设置业务处理的回调。

g\_tcp\_cbs 是设置协议解析相关的回调。

online\_service.cpp是处理 具体业务请求的相关实现。online\_proc 是主方法，里面根据具体的命令字请求相关业务处理方法。

SvrMgr.h|.cpp PC推送服务器管理类，服务器的状态会记录到共享内存中(文件mmap)。

StatusDao.h|.cpp 用户状态管理类，记录到共享内存中(文件mmap)。

config命令：

./configure --prefix=/usr/local/msgonline --sbin-path=sbin/msgonline --builddir=msgonline-objs \

--conf-path=conf/msgonline.conf --add-module=src/tcpserver --mod-config=msgonline/config-msgonline \

--with-debug --without-pcre --without-http

编译：

make –f msgonline-objs/Makefile

编译完成后，msgonline-objs目录下会生成一个nginx程序。

运行：

可以直接运行编译目录下的程序：

msgonline-objs/nginx –t 测试配置文件。#也可用于查看当前使用的是哪个配置文件。

msgonline-objs/nginx 运行。

或者

cp msgonline-objs/nginx /usr/local/msgonline/sbin/msgonline

然后执行：

/usr/local/msgonline/sbin/msgonline

配置：

msgonline有两个配置文件，一个是nginx的配置文件：

/usr/local/msgonline/conf/msgonline.conf

另外一个是应用自己相关的配置:

/usr/local/msgonline/conf/msgonlinecfg.ini

save\_dir=/data/msgonline #共享的文件保存位置。

save\_interval=60 #保存间隔。

ios\_expiretime=300 #ios 登录过期时间。

svr\_expiretime=3600 #服务器没心跳后的过期时间(过期即认为服务器已经死亡)

其它配置请见代码：online\_impl.cpp:: online\_cfg\_init方法。

日志文件在/data/msgonline目录下

依赖项：

以上几个项目都依赖于：/home/lxj/work/Thumbs/thumbpub/BaseLib

到该目录后， make 可以编译出baselib.a

## 监控：

在201主机的/etc/crontab中有心上三个程序及mongod的监控脚本：

\* \* \* \* \* root mon.sh /usr/local/msgonline/sbin/msgonline

\* \* \* \* \* root mon.sh /usr/local/qmvsvr/sbin/qmvsvr

\* \* \* \* \* root mon.sh /usr/local/msgstorage/sbin/msgstorage

\* \* \* \* \* root /usr/bin/mon\_mongo.sh

## 关于数据存储：

### qmv数据存储在mysql中，库为172.16.10.201 QMV\_LAB

mysql -uroot -p123456

服务器部署地址为：113.107.201.113:22222

远程帐号songhero 密码testtesttest

root密码qvod\_123

mysql root qmv@@qvod

### 消息系统数据存储在mongodb中，查询方法是直接使用mongo命令就可以连接上，然后

use msgcenter; #打开msgcenter库

show tables; #显示所有表。

db.msg\_broadcast.find() 查询

## 消息系统PC模拟测试程序：

/home/lxj/work/Thumbs/testmsg

./testmsg -?

./testmsg -h host -p port -t [threads] -r [request times] -f [func] -a bindaddr -b base -? help

-f func: 1: redirect 2: login, 3:all

-h 指定服务器地址。

-p 指定端口地址。

-t 线程数 (测试 功能 3， all时，该参数无效。)

-r 请求数。

-f 测试功能：

1，只测试重定向功能。

2，只测试登录功能(-p直接指定登录服务的端口), 及登出功能。测试完成后，会对所有的测试用户进行登出操作。

3，整体的测试，测试从重定向，到登录，心跳，接收消息所有功能。心跳是根据服务器返回的时间间隔自动发送。

-b userid起始位置，默认是0，如 请求数为100次，userid为0-100。

-a 绑定的本机地址，当要测试大于6W个连接时，可以绑定到多个IP实现。