视频服务器安装配置手册

**目录**

[一、Java环境安装和部署 2](#_Toc527038242)

[1.1、JDK安装 2](#_Toc527038243)

[1.1.1、上传安装包 2](#_Toc527038244)

[1.1.3、环境变量配置 2](#_Toc527038245)

[1.2、Tomcat安装 3](#_Toc527038246)

[1.2.1、上传安装包 3](#_Toc527038247)

[1.2.2、解压tomcat包 3](#_Toc527038248)

[1.2.3、软件部署 3](#_Toc527038249)

[1.2.4、获取权限 3](#_Toc527038250)

[1.2.5、启动服务 4](#_Toc527038251)

[1.2.6、检测服务启动成功 4](#_Toc527038252)

[1.2.7、关闭tomcat 4](#_Toc527038253)

[1.2.8、删除RtmpServer.war包 4](#_Toc527038254)

[1.2.9、tomcat下RtmpServer服务系统配置 4](#_Toc527038255)

[二、C++程序安装部署 4](#_Toc527038256)

[2.1、gcc和gcc++安装 4](#_Toc527038257)

[2.1.1、解压文件 5](#_Toc527038258)

[2.1.2、下载、配置和安装 5](#_Toc527038259)

[2.1.3、查看最新gcc版本 6](#_Toc527038260)

[2.2、运行程序 6](#_Toc527038261)

[2.2.1、上传程序文件 6](#_Toc527038262)

[2.2.2、环境配置 6](#_Toc527038263)

[2.2.3、glibc安装 7](#_Toc527038264)

[2.2.3、启动服务 7](#_Toc527038265)

[三、流媒体服务安装 8](#_Toc527038266)

[3.1、流媒体主服务安装 8](#_Toc527038267)

[3.1.1、上传文件 8](#_Toc527038268)

[3.1.2、解压文件 8](#_Toc527038269)

[3.1.3、安装文件（centos6.5系统跳过安装步骤） 9](#_Toc527038270)

[3.1.4、流媒体服务获取文件的读写权限 9](#_Toc527038271)

[3.1.5、文件上传 9](#_Toc527038272)

[3.1.6、启动src服务 9](#_Toc527038273)

[3.1.7、检查srs服务器有没有启动成功 9](#_Toc527038274)

[3.2、流媒体采集环境安装 10](#_Toc527038275)

[3.2.1、文件上传 10](#_Toc527038276)

[3.2.2、解压文件 10](#_Toc527038277)

[3.2.3、安装yasm 10](#_Toc527038278)

[3.2.4、安装ffmpeg 10](#_Toc527038279)

[3.3、流媒体服务器防火墙端口开放 10](#_Toc527038280)

[四、memcached安装 11](#_Toc527038281)

[4.1、gcc和gcc++安装 11](#_Toc527038282)

[4.2、libevent的安装 11](#_Toc527038283)

[4.3、memcache安装 11](#_Toc527038284)

[4.4、memcached的启动 11](#_Toc527038285)

[五、注意事项 12](#_Toc527038286)

[5.1、视频服务器启动顺序 12](#_Toc527038287)

[5.2、FAQ 12](#_Toc527038288)

[5.3、流媒体所有服务设置开机自启动 12](#_Toc527038289)

[5.3.1、流媒体服务 12](#_Toc527038290)

[5.3.2、memcached服务 12](#_Toc527038291)

[5.3.3、C++服务开机自启动 14](#_Toc527038292)

[5.3.4、tomcat服务开机自启动 15](#_Toc527038293)

# 一、Java环境安装和部署

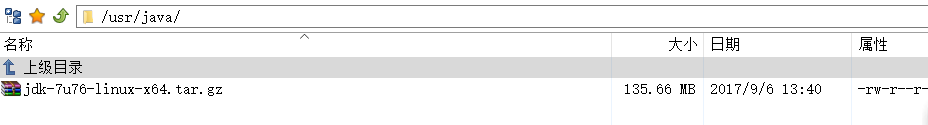
## 1.1、JDK安装

### 1.1.1、上传安装包

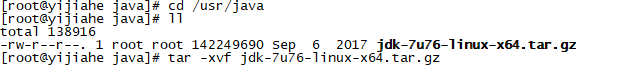
创建目录 /usr/java/

mkdir /usr/java

上传安装包到此目录



进入上传目录 tar -zxvf 解压jdk包



### 1.1.3、环境变量配置

使用vi /etc/profile 编辑profile文件 输入： vi /etc/profile

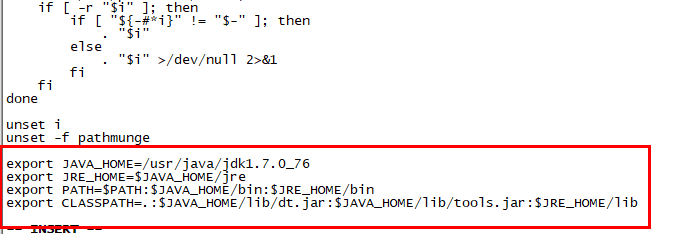
向文件最后添加以下内容：

export JAVA\_HOME=/usr/java/jdk1.7.0\_76

export JRE\_HOME=$JAVA\_HOME/jre

export PATH=$PATH:$JAVA\_HOME/bin:$JRE\_HOME/bin

export CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar:$JRE\_HOME/lib



#使其环境变量生效

# source /etc/profile

#验证环境变量是否生效

#java –version



安装完成

## 1.2、Tomcat安装

### 1.2.1、上传安装包

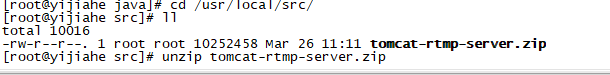
将tomcat-rtmp-server.zip 上传 Linux服务器 "/usr/local/srs" 目录



### 1.2.2、解压tomcat包

# cd /usr/local/src/

# unzip tomcat-rtmp-server.zip



mv tomcat-rtmp-server /usr/

### 1.2.3、软件部署

将RtmpServer.war包上传至 Linux服务器 "/usr/tomcat-rtmp-server/webapps" 目录

### 1.2.4、获取权限

cd /usr/tomcat-rtmp-server/bin

chmod 777 \*

### 1.2.5、启动服务

cd /usr/tomcat-rtmp-server/bin

./startup.sh

### 1.2.6、检测服务启动成功

ps -ef | grep tomcat 出现 tomcat的进程

### 1.2.7、关闭tomcat

cd /usr/tomcat-rtmp-server/bin

./shutdown.sh

### 1.2.8、删除RtmpServer.war包

cd /usr/tomcat-rtmp-server/webapps/

rm RtmpServer.war

### 1.2.9、tomcat下RtmpServer服务系统配置

修改红色标注的地方

Jdbc.properties

#驱动oracle

driverClass=oracle.jdbc.driver.OracleDriver

#jdbc地址

jdbcUrl=jdbc:oracle:thin:@192.168.2.40:1521:orcl

#用户名

user=ZSTEST

#密码

password=ZSTEST

红色部分为工区数据库账号和密码

# 二、C++程序安装部署

## 2.1、gcc和gcc++安装

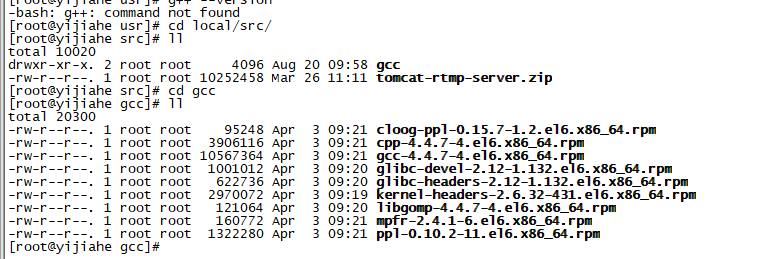
# gcc --version

# g++ --version



检查当前服务器的版本，如果没有安装或者不是4.8.5版本需要手动安装gcc和gcc-c++

上传以下包至/usr/local/src/gcc目录下



安装顺序：

ppl,clog-ppl,mpfr,cpp,gcc,libstdc++-devel,gcc-c++

cd /usr/local

rpm -ivh 上面的包的名字

按照顺序来安装

安装完成后查看gcc和gcc-c++的版本应该为4.4.7

Centos6.5最高是4.4.7，需要手动下载gcc更高版本包进行升级

升级gcc版本至4.8.5，如果当前版本是该版本，则忽略此步骤。

上传gcc-4.8.5.tar.gz到/usr/local目录下

### 2.1.1、解压文件

cd /usr/local

tar -zxvf gcc-4.8.5.tar.gz

### 2.1.2、下载、配置和安装

进入到压缩目录执行命令：

# cd /usr/local/gcc-4.8.5

上传gmp-4.3.2.tar,mpc-0.8.1.tar,mpfr-2.4.2.tar到目录/usr/local/gcc-4.8.5下

上传文件download\_prerequisites到/usr/local/gcc-4.8.5/contrib/目录下，覆盖原文件

# cd /usr/local/gcc-4.8.5

# ./contrib/download\_prerequisites

在当前目录下创建一个目录，存放编译后的文件：

# mkdir gcc-build-4.8.5

执行命令：

# cd gcc-build-4.8.5

生成Makefile文件，准备编译整个项目（gcc4.8.5）：

# ../configure -enable-checking=release -enable-languages=c,c++ -disable-multilib

（注意：..表示会进入到上一层目录执行configure。所以gcc-build-4.8.5目录必须在gcc-4.8.5里面创建，也就是我们这样创建的。如果你想自己定义gcc-build-4.8.5目录的位置，你要处理好这里的路径问题。）

编译gcc4.8.5的源码，非常耗时，时间决定于你的机器的性能：

# make

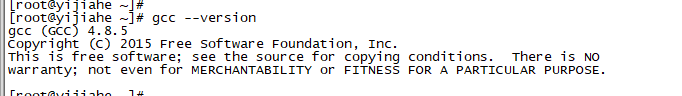
观察是否出现error，如果没出现error，进入下一步

安装gcc4.8.5：

#make install

### 2.1.3、查看最新gcc版本

# gcc --version



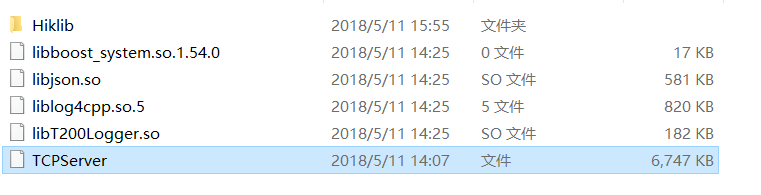
如果版本为gcc-4.8.5，说明安装成功

如果不是则需要重新启动服务器，在查看版本。

## 2.2、运行程序

### 2.2.1、上传程序文件

将以下文件上传至/usr/local/srs/文件夹下



### 2.2.2、环境配置

执行命令：

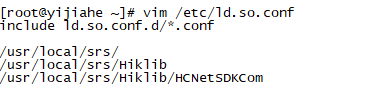
# vim /etc/ld.so.conf

添加如下内容

/usr/local/srs/

/usr/local/srs/Hiklib

/usr/local/srs/Hiklib/HCNetSDKCom



cd /usr/local/srs/Hiklib/HCNetSDKCom

mv libiconv.so.2 libiconv

ln -s libiconv libiconv.so.2

执行命令：

# ldconfig

使得配置生效。

### 2.2.3、glibc安装

如果系统的glibc版本不是2.14，则需要重新安装glibc-2.14

上传glibc-2.14.tar.gz至目录/root下

# cd /root

解压文件

# tar -zxvf glibc-2.14.tar.gz

# cd glibc-2.14

# mkdir build

# cd build

# ../configure --prefix=/root/glibc-2.14

# make ------时间比较长

# make install

更新系统的链接

cp /usr/local/src/ glibc-2.14/ build/libc.so /lib64

mv /lib64/libc.so /lib64/libc-2.14.so

rm -rf /lib64/libc.so.6

ln -s /lib64/libc-2.14.so /lib64/libc.so.6

LD\_PRELOAD=/lib64/libc-2.14.so ln -s /lib64/libc-2.14.so  /lib64/libc.so.6

**注意：**

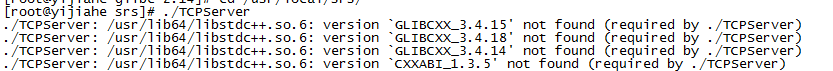
strings /lib64/libc.so.6|grep GLIBC -----查询版本

### 2.2.3、启动服务

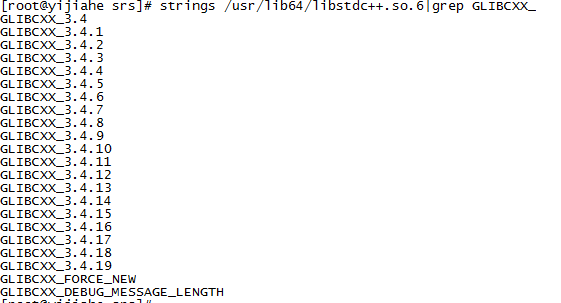
# cd /usr/local/srs

# ./TCPServer

如果报如下错误



缺少GLIBCXX\_3.4.15版本，或是更高的版本。  
strings /usr/local/lib64/libstdc++.so.6|grep GLIBCXX\_



这里该有的都有了，把这份软链到正确的地方

ls -ls /usr/lib64/libstdc++.so\*

ls -ls /usr/local/lib64/libstdc++.so\*

cp /usr/local/lib64/libstdc++.so.6.0.19 /usr/lib64/

cd /usr/lib64

rm -rf libstdc++.so.6

ln -s libstdc++.so.6.0.19 libstdc++.so.6

ls -l libstdc++\*

重新进入/usr/local/srs目录下，执行命令

nohup ./TCPServer > server.txt &

查看进程

ps -ef|grep TCPServer



# 三、流媒体服务安装

## 3.1、流媒体主服务安装

### 3.1.1、上传文件

将SRS-CentOS6-x86\_64-2.0.243.zip上传至Linux服务器 "/root" 目录

### 3.1.2、解压文件

#cd /root

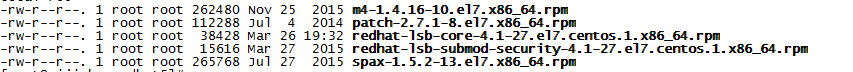
#unzip SRS-CentOS6-x86\_64-2.0.243.zip

### 3.1.3、安装文件（centos6.5系统跳过安装步骤）

将redhat5.zip上传至/root目录下

unzip /root/redhat5.zip

cd /root/redhat5



安装这五个包

rpm -ivh 包名

如果遇到无法安装

rpm -ivh 包名 --nodeps –force

进行强制安装

### 3.1.4、流媒体服务获取文件的读写权限

cd /root/SRS-CentOS6-x86\_64-2.0.243/

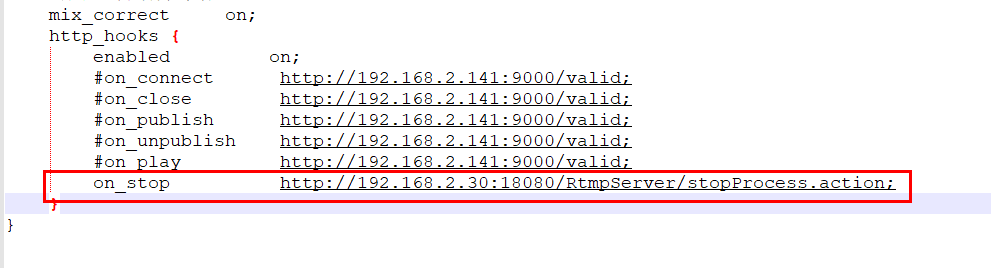
sudo bash INSTALL

chmod 777 /usr/local/srs/objs/srs

### 3.1.5、文件上传

将srs-yijiahe.conf上传至Linux服务器 "/usr/local/srs/conf" 目录

修改里面的内容，将图片中红框部分改为现场服务器的地址端口



将start-srs-with-YJH-conf上传至Linux服务器 "/usr/local/srs" 目录

### 3.1.6、启动src服务

cd /usr/local/srs

bash start-srs-with-YJH-conf

### 3.1.7、检查srs服务器有没有启动成功

重新开一个页面

ps -ef | grep srs

出现 srs的进程中包含 ./objs/srs -c ./conf/srs.conf 就启动成功了

## 3.2、流媒体采集环境安装

### 3.2.1、文件上传

将ffmpeg-3.4.2.tar.bz2上传至Linux服务器 "/root" 目录

将yasm-1.3.0.tar.gz上传至Linux服务器 "/root" 目录

### 3.2.2、解压文件

cd /root

tar -xjvf ffmpeg-3.4.2.tar.bz2

tar -zxvf yasm-1.3.0.tar.gz

### 3.2.3、安装yasm

cd /root/yasm-1.3.0

./configure

make

make install

### 3.2.4、安装ffmpeg

cd /root/ffmpeg-3.4.2

./configure

make

make install

安装过程很长，请耐心等待

## 3.3、流媒体服务器防火墙端口开放

查看防火墙的状态

# /etc/init.d/iptables status

8888-流媒体，8080-tomcat，11211-缓存，10002-socket

LINUX通过下面的命令可以开启允许对外访问的网络端口：

/sbin/iptables -I INPUT -p tcp --dport 8888 -j ACCEPT #开启8888端口

/sbin/iptables -I INPUT -p tcp --dport 8080 -j ACCEPT #开启8080端口

/sbin/iptables -I INPUT -p tcp --dport 11211 -j ACCEPT #开启11211端口

/sbin/iptables -I INPUT -p tcp --dport 10002 -j ACCEPT #开启10002端口

/etc/rc.d/init.d/iptables save #保存配置

/etc/rc.d/init.d/iptables restart #重启服务

# 四、memcached安装

## 4.1、gcc和gcc++安装

同3.1，如果已经安装则不需要重新安装

gcc --version

gcc-c++ --version

查看是否安装

## 4.2、libevent的安装

将libevent-2.0.21-stable.tar包上传至/usr目录下

# cd /usr

# tar -zxvf libevent-2.0.21-stable.tar.gz

# cd libevent-2.0.21-stable

#./configure --prefix /usr/local/libevent/

#make

#make install

安装完后可以查看下/usr/lib是否有libevent等文件

# ls -al /usr/lib | grep libevent

## 4.3、memcache安装

把安装包memcached-1.4.25.tar上传至 /usr目录下

# cd /usr

# tar -zxvf memcached-1.4.25.tar.gz

# cd memcached-1.4.25

#./configure --prefix /usr/local/memcached/ --with-libevent=/usr/local/libevent/

#make

#make install

安装完检查是否安装完成

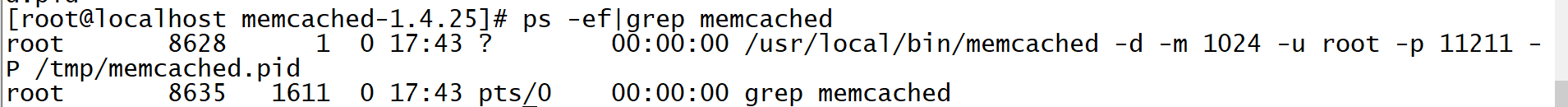
# ls -al /usr/local/bin/memcached

## 4.4、memcached的启动

# memcached -d -m 256 -u root -p 11211 -c 1024

查看是否有该进程，启动成功

# ps -ef|grep memcached



# 五、注意事项

## 5.1、视频服务器启动顺序

流媒体服务——>memcached ——>C++程序服务——>tomcat服务

## 5.2、FAQ

1、在启动服务的过程中如果发现端口被占用，并且之前的进程已经杀掉时，使用命令：netstat -nlp | grep 端口号| awk '{print $7}' | awk -F "/" '{print $1}'，查看端口占用的进程号，然后杀掉进程号。

2、发现C++程序一直在启动中时，需要一直查询被端口占用的进程，直到全部杀完为止。使用这两个命令：pkill ffmpeg、pkill tail，然后重新启动C++程序。

3、lsof -i:端口号

4、升级glibc库后，使用SecureCRT登录linux，提示:

Last login: Mon Feb 20 02:03:21 2017 from 192.168.4.187

-bash: warning: setlocale: LC\_CTYPE: cannot change locale (en\_US.UTF-8): No such file or directory

-bash: warning: setlocale: LC\_COLLATE: cannot change locale (en\_US.UTF-8): No such file or directory

-bash: warning: setlocale: LC\_MESSAGES: cannot change locale (en\_US.UTF-8): No such file or directory

-bash: warning: setlocale: LC\_NUMERIC: cannot change locale (en\_US.UTF-8): No such file or directory

-bash: warning: setlocale: LC\_TIME: cannot change locale (en\_US.UTF-8): No such file or directory

vi /etc/sysconfig/i18n

# 添加下面的

LC\_ALL=C

export LC\_ALL

# 再执行下面的语句，使之生效

source /etc/sysconfig/i18n

## 5.3、流媒体所有服务设置开机自启动

### 5.3.1、流媒体服务

srs服务目前已设置成自启动的形式，不需要做任何修改

### 5.3.2、memcached服务

在/etc/init.d/目录下新建一个脚本，名称为：memcached。内容如下：

#! /bin/bash

#

# memcached start/stop the memcached daemon

#

# chkconfig: 35 80 70

# description: memcached is a memory cache server.

#

prog="memcached"

exec=/usr/local/memcached/bin/memcached

lockfile=/var/lock/subsys/memcached

# source function library.

. /etc/rc.d/init.d/functions

start() {

if [ $UID -ne 0 ]; then

echo "User has insufficient privilege."

exit 4

fi

[ -x $exec ] || exit 5

echo -n $"starting $prog: "

daemon $exec -u root -d -P /var/run/memcached.pid

retval=$?

echo

[ $retval -eq 0 ] && touch $lockfile

}

stop() {

if [ $UID -ne 0 ]; then

echo "User has insufficient privilege."

exit 4

fi

echo -n $"Stopping $prog: "

if [ -n "`pidfileofproc $exec`" ]; then

killproc $exec

else

failure $"stopping $prog"

fi

retval=$?

echo

[ $retval -eq 0 ] && rm -f $lockfile

}

restart() {

stop

start

}

rh\_status() {

# run checks to determine if the service is running or use generic status

status $prog

}

rh\_status\_q() {

rh\_status >/dev/null 2>&1

}

case "$1" in

"start")

rh\_status\_q && exit 0

$1

;;

"stop")

rh\_status\_q || exit 0

$1

;;

"restart")

rh\_status\_q || exit 7

$1

;;

"status")

rh\_status

;;

\*)

echo $"Usage: $0 {start|stop|status|restart}"

exit 2

;;

esac

exit $?

给memcached赋予权限

chmod 777 /etc/init.d/memcached

启动memcached服务

service memcached start

设置memcached服务加入开机自启动

chkconfig --add memcached

### 5.3.3、C++服务开机自启动

在 /usr/local/srs/新建脚本TCPServer.sh内容如下：

export FFMPEG\_HOME=/usr/local

export PATH=$PATH:$FFMPEG\_HOME/bin

cd /usr/local/srs

nohup ./TCPServer > server.txt &

在/etc/rc.local中添加如下内容：

/usr/local/srs/TCPServer.sh

### 5.3.4、tomcat服务开机自启动

在/etc/rc.local中添加如下内容：

export JAVA\_HOME=/usr/java/jdk1.7.0\_76

/usr/tomcat-rtmp-server/bin/startup.sh start