使用下面的命令检查是否安装有MySQL Server

rpm -qa | grep mysql

卸载软件

rpm -e mysql //普通删除模式

rpm -e --nodeps mysql // 强力删除模式，如果使用上面命令删除时，提示有依赖的其它文件，则用该命令可以对其进行强力删除

chkconfig --level 2345 memcached on 执行该命令将memcached加入linux服务。添加成功以后，你可以使用service memcached start|stop|restart等命令来启动、关闭、重启memcached服务。

# Aapche

Apache依赖apr和apr-util包，需要先安装这俩个包，apr-util又依赖apr包，所以要先安装apr

1. 先下载apr和apr-util包



2.编译安装apr



3.编译安装apr-util

./configure --prefix=/usr/local/apr-util-1.5.4 --with-apr=/usr/local/apr



4.编译安装Apache

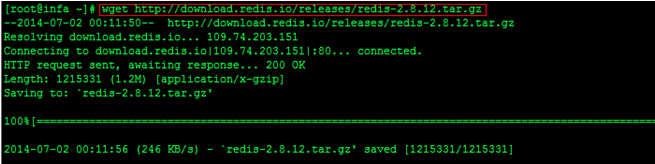
./configure --prefix=/usr/local/apache/ --with-apr=/usr/local/apr --with-apr-util=/usr/local/apr-util-1.5.4 



# 编译安装redis

我用的系统是：redhat

[root@infa ~]# wget <http://download.redis.io/releases/redis-2.8.12.tar.gz>



tar xzf redis-2.8.12.tar.gz

[root@infa ~]# tar xzf redis-2.8.12.tar.gz

[root@infa ~]# cd redis-2.8.12

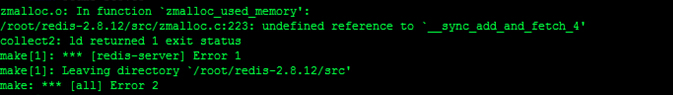
[root@infa redis-2.8.12]# ls

00-RELEASENOTES  BUGS  CONTRIBUTING  COPYING  deps  INSTALL  Makefile  MANIFESTO  README  redis.conf  runtest  runtest-sentinel  sentinel.conf  src  tests  utils

[root@infa redis-2.8.12]# make

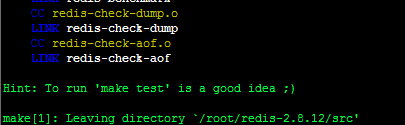
但是这样子编译会报错，貌似是因为系统的原因(注：我centos没报错)；

报错如下：



解决方法，就是加上编译参数设置，如下：

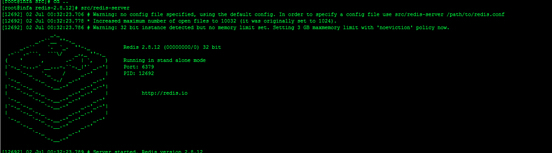
[root@infa redis-2.8.12]# make CFLAGS="-march=i686"



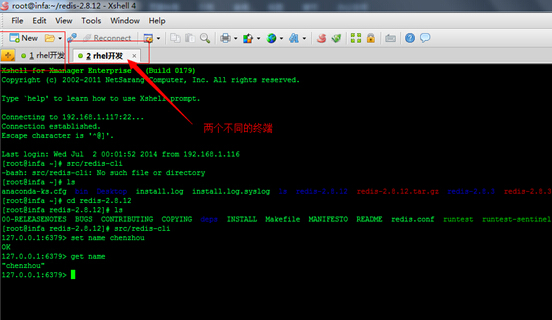
编译成功；

启动并运行redis：

[root@infa redis-2.8.12]# src/redis-server



打开另外一个终端运行，redis客户端



[root@infa redis-2.8.12]# src/redis-cli

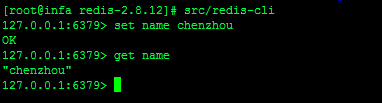
127.0.0.1:6379> set name chenzhou

OK

127.0.0.1:6379> get name

"chenzhou"

127.0.0.1:6379>

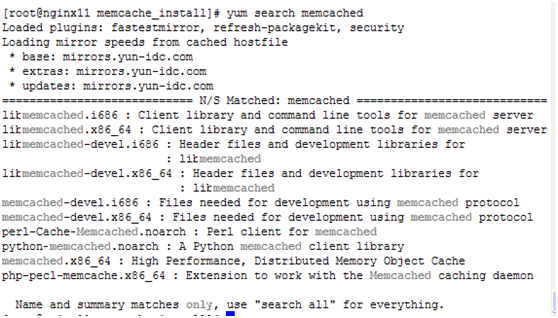


# Yum安装Memcache

## 查找memcached

                   yum  search  memcached

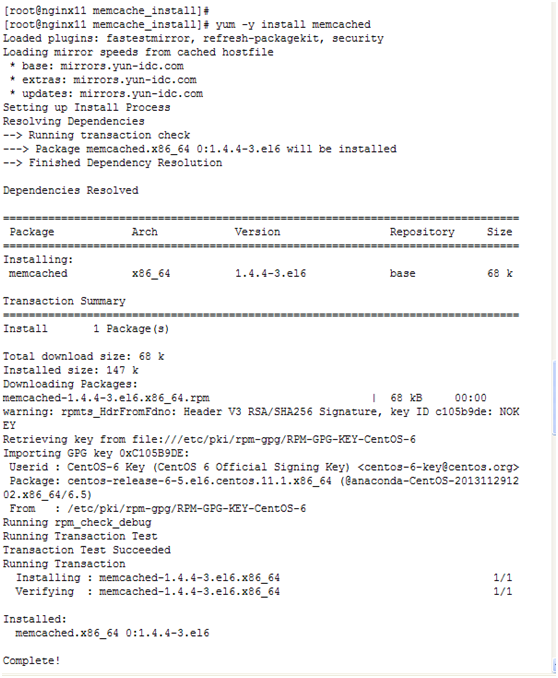
         该命令可以查询yum库中有关memcached的安装包信息，以下是搜寻结果截图：



## 安装 memcached

         yum  -y install memcached

该命令用来安装memcached，不必关心其中各个安装包之间的依赖。命令执行结果如下图所示：



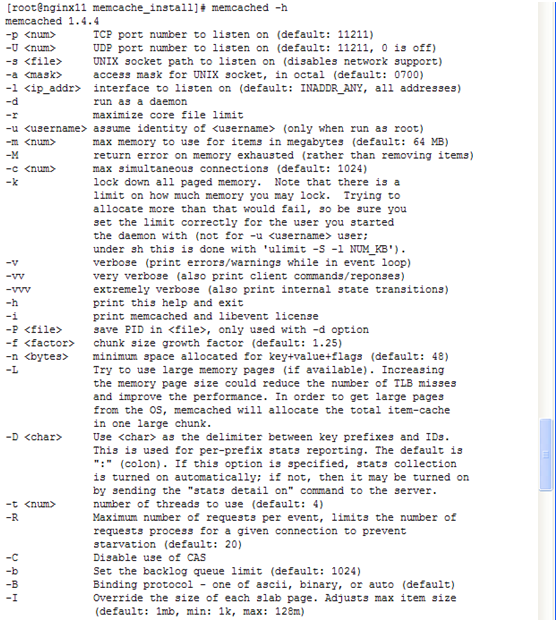
         （备注：如果执行yum –y install memcached报网络解析异常，则说明当前无法联网，或者说明dns服务器没有正确配置，如果因为dns没有配置原因，可以通过 vi /etc/resolv.conf

添加         nameserver     192.98.18.20(dns服务器地址) 来配置dns映射）

## 验证安装memcached

         memcached –h

该命令主要的功能是提供memcached的帮助信息，罗列memcached命令有哪些接口以及接口参数。具体信息参考截图：



## 修改memcached配置文件

         chkconfig --level 2345 memcached on 执行该命令将memcached加入linux服务。添加成功以后，你可以使用service memcached start|stop|restart等命令来启动、关闭、重启memcached服务。

         修改memcached的参数，主要涉及memcached启动绑定的端口、用户、最大连接、最大内存等，命令：vi /etc/sysconfig/memcached 命令响应结果如截图：



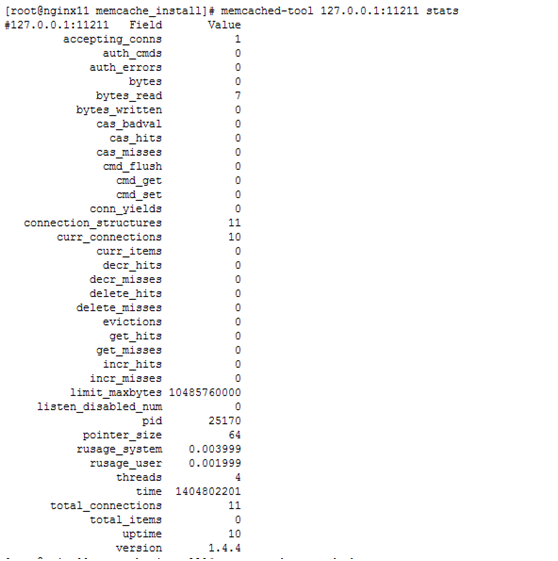
         通过vi命令即可修改上面各个参数。PORT,服务启动端口。USER,服务启动用户。MAXCONN,服务配置支持最大连接数。CACHESIZE,服务支持的最大内存，以M为单位。

## 重启memcached服务

         修改上面的参数以后，需要重启memcached服务，不然参数是无法立即生效的。重启服务，只需执行：service memcached restart。服务重启以后，即可使用memcached-tool来检测memcached服务状态。

## 使用memcached-tool检测memcached服务

         执行memcached-tool 127.0.0.1:11211 stats即可检测memcached的服务状态。如果报错，连接不上，那么说明memcached没有正确安装或者没有启动。执行正常，内容如截图所示：



标签: [NoSQL](http://www.cnblogs.com/likehua/tag/NoSQL/), [Memcached](http://www.cnblogs.com/likehua/tag/Memcached/)

安装tomcat

1. 先安装jdk环境



wget <http://download.oracle.com/otn-pub/java/jdk/8u60-b27/jdk-8u60-linux-x64.tar.gz>

# 安装nginx

下载nginx（目前稳定版）<http://nginx.org/download/nginx-1.5.13.tar.gz>

Nginx依赖于zlib zlib-devel和pcre包

下载pcre（支持nginx伪静态）<http://sourceforge.net/projects/pcre/files/pcre/8.35/pcre-8.35.tar.gz>

安装PCRE

tar zxvf pcre-8.35.tar.gz

cd pcre-8.35

./configure --prefix=/usr/local/pcre

make

make install

安装nginx

cd /usr/local/src

groupadd www

useradd -g www www -s /bin/false

tar zxvf nginx-1.5.13.tar.gz

cd nginx-1.5.13

. ./configure --prefix=/usr/local/nginx \

--with-http\_stub\_status\_module \

--with-http\_ssl\_module \

--with-http\_gzip\_static\_module \

--with-http\_realip\_module \

--http-client-body-temp-path=/var/tmp/nginx/client \

--http-proxy-temp-path=/var/tmp/nginx/proxy/ \

--http-fastcgi-temp-path=/var/tmp/nginx/fcgi/

问题：

./configure: error: the HTTP rewrite module requires the PCRE library.

You can either disable the module by using --without-http\_rewrite\_module

option, or install the PCRE library into the system, or build the PCRE library

statically from the source with nginx by using --with-pcre=<path> option.

解决办法：

安装pcre-devel解决问题

yum -y install pcre-devel

重新编译

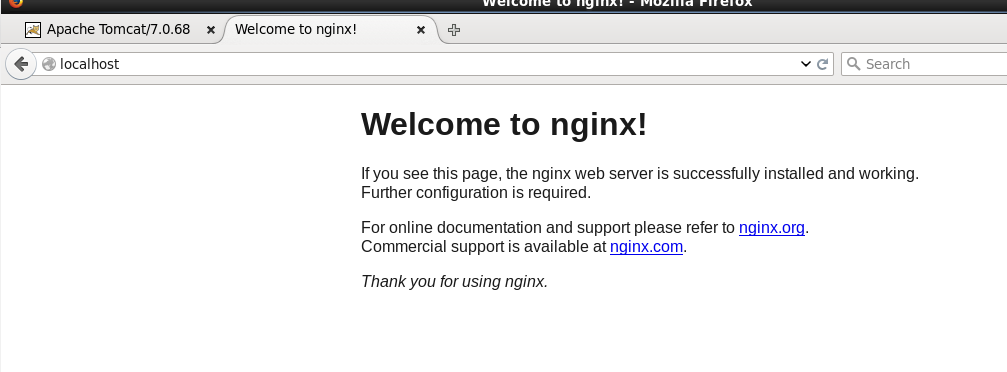
make

make install

测试安装是否成功

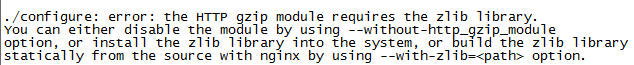
进入/nginx/sbin目录， ./nginx启动nginx，然后输入localhost，出现下面网页表示安装成功





开机自启动：Chekconfig --level 2345 nginx on

注：编译遇到下面错误，需要安装zlib-devel包



安装即可



如果遇到如下错误：

./configure: error: the HTTP rewrite module requires the PCRE library.

You can either disable the module by using --without-http\_rewrite\_module

option, or install the PCRE library into the system, or build the PCRE library

statically from the source with nginx by using --with-pcre= option.

用以下命令解决

*# yum install pcre-devel*

看到错误：

/configure: error: SSL modules require the OpenSSL library.  
You can either do not enable the modules, or install the OpenSSL library  
into the system, or build the OpenSSL library statically from the source  
with nginx by using --with-openssl= option.  
  
这样解决：

*#yum install openssl openssl-devel*

看到下面的情形一般表示成功了！

configuration summary  
  + using system PCRE library  
  + using system OpenSSL library  
  + md5: using OpenSSL library  
  + sha1: using OpenSSL library  
  + using system zlib library  
  
  nginx path prefix: "/webserver/nginx"  
  nginx binary file: "/webserver/nginx/sbin/nginx"  
  nginx configuration prefix: "/webserver/nginx/conf"  
  nginx configuration file: "/webserver/nginx/conf/nginx.conf"  
  nginx pid file: "/webserver/nginx/logs/nginx.pid"  
  nginx error log file: "/webserver/nginx/logs/error.log"  
  nginx http access log file: "/webserver/nginx/logs/access.log"  
  nginx http client request body temporary files: "client\_body\_temp"  
  nginx http proxy temporary files: "proxy\_temp"  
  nginx http fastcgi temporary files: "fastcgi\_temp"  
  nginx http uwsgi temporary files: "uwsgi\_temp"  
  nginx http scgi temporary files: "scgi\_temp"

# yum安装tomcat

yum安装tomcat会自动安装相关的软件，如jre环境等，因此不需要单独安装jre



默认配置文件在/etc/tomcat6/下面

# 编译安装mysql

<http://jingyan.baidu.com/article/c1465413ac2fb30bfcfc4c3e.html>

mysql依赖包

yum -y install make gcc-c++ cmake bison-devel ncurses-devel

卸载旧版本mysql

使用下面的命令检查是否安装有MySQL Server

rpm -qa | grep mysql

有的话通过下面的命令来卸载掉

rpm -e mysql //普通删除模式

rpm -e --nodeps mysql // 强力删除模式，如果使用上面命令删除时，提示有依赖的其它文件，则用该命令可以对其进行强力删除

先安装cmake 解压编译安装

## 解压mysql



## 切到mysql目录，执行编译



**cmake  \**

**-DCMAKE\_INSTALL\_PREFIX=/usr/local/mysql    \**

**-DMYSQL\_DATADIR=/usr/local/mysql/data    \**

**-DSYSCONFDIR=/etc   \**

**-DWITH\_MYISAM\_STORAGE\_ENGINE=1 \**

**-DWITH\_INNOBASE\_STORAGE\_ENGINE=1 \**

**-DWITH\_MEMORY\_STORAGE\_ENGINE=1 \**

**-DWITH\_READLINE=1 \**

**-DMYSQL\_UNIX\_ADDR=/var/lib/mysql/mysql.sock \**

**-DMYSQL\_TCP\_PORT=3306 \**

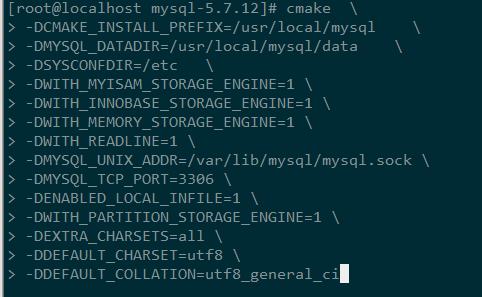
**-DENABLED\_LOCAL\_INFILE=1 \**

**-DWITH\_PARTITION\_STORAGE\_ENGINE=1 \**

**-DEXTRA\_CHARSETS=all \**

**-DDEFAULT\_CHARSET=utf8 \**

**-DDEFAULT\_COLLATION=utf8\_general\_ci**



### 编译参数解释：

-DCMAKE\_INSTALL\_PREFIX= 指向mysql安装目录

-DINSTALL\_SBINDIR=sbin 指向可执行文件目录（prefix/sbin）

-DMYSQL\_DATADIR=/var/lib/mysql 指向mysql数据文件目录（/var/lib/mysql）

-DSYSCONFDIR=/etc/mysql 指向mysql配置文件目录（/etc/mysql）

-DINSTALL\_PLUGINDIR=lib/mysql/plugin 指向插件目录（prefix/lib/mysql/plugin）

-DINSTALL\_MANDIR=share/man 指向man文档目录（prefix/share/man）

-DINSTALL\_SHAREDIR=share 指向aclocal/mysql.m4安装目录（prefix/share）

-DINSTALL\_LIBDIR=lib/mysql 指向对象代码库目录（prefix/lib/mysql）

-DINSTALL\_INCLUDEDIR=include/mysql 指向头文件目录（prefix/include/mysql）

-DINSTALL\_INFODIR=share/info 指向info文档存放目录（prefix/share/info）

prefix官方推荐设为/usr

Storage Engine相关

类型csv,myisam,myisammrg,heap,innobase,archive,blackhole

若想启用某个引擎的支持：-DWITH\_<ENGINE>\_STORAGE\_ENGINE=1

如：

-DWITH\_INNOBASE\_STORAGE\_ENGINE=1

-DWITH\_ARCHIVE\_STORAGE\_ENGINE=1

-DWITH\_BLACKHOLE\_STORAGE\_ENGINE=1

若想禁用某个引擎的支持：-DWITHOUT\_<ENGINE>\_STORAGE\_ENGINE=1

如：

-DWITHOUT\_EXAMPLE\_STORAGE\_ENGINE=1

-DWITHOUT\_FEDERATED\_STORAGE\_ENGINE=1

-DWITHOUT\_PARTITION\_STORAGE\_ENGINE=1

Library相关

-DWITH\_READLINE=1 启用readline库支持（提供可编辑的命令行）

-DWITH\_SSL=system 启用ssl库支持（安全套接层）

-DWITH\_ZLIB=system 启用libz库支持（zib、gzib相关）

-DWTIH\_LIBWRAP=0 禁用libwrap库（实现了通用TCP包装的功能，为网络服务守护进程使用）

-DMYSQL\_TCP\_PORT=3306 指定TCP端口为3306

-DMYSQL\_UNIX\_ADDR=/tmp/mysqld.sock 指定mysql.sock路径

-DENABLED\_LOCAL\_INFILE=1 启用本地数据导入支持

-DEXTRA\_CHARSETS=all 启用额外的字符集类型（默认为all）

-DDEFAULT\_CHARSET=utf8 指定默认的字符集为utf8

-DDEFAULT\_COLLATION=utf8\_general\_ci 设定默认排序规则（utf8\_general\_ci快速/utf8\_unicode\_ci准确）

-DWITH\_EMBEDDED\_SERVER=1 编译嵌入式服务器支持

-DMYSQL\_USER=mysql 指定mysql用户(默认为mysql)

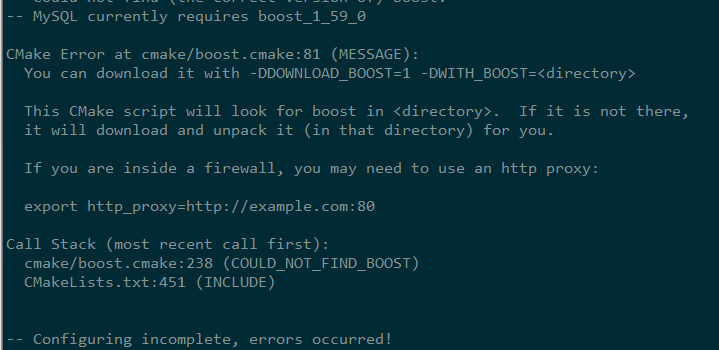
-DWITH\_DEBUG=0 禁用debug（默认为禁用）

-DENABLE\_PROFILING=0 禁用Profiling分析（默认为开启）

-DWITH\_COMMENT='string' 一个关于编译环境的描述性注释

### 编译出错解决

如果编译时提示下面错误，说明未安装boost包，MySQL5.7以及更高的版本时，都需要下载并引用或者直接安装boost库



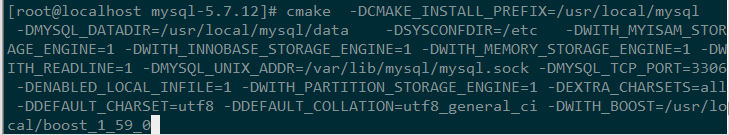
解决方法：

下载一个boost包，放到/usr/local/boost目录下，然后在cmake后面加选项-DWITH\_BOOST=/usr/local/boost

**注意：编译时提示需要boost哪个版本，只能去下载哪个版本，否则还会出错，且每次重新Cmake编译的时候，需要删除CMakeCache.txt文件。（# rm -f CMakeCache.txt）**



**下载好boost后，重新编译**



**[root@localhost mysql-5.7.12]# cmake -DCMAKE\_INSTALL\_PREFIX=/usr/local/mysql -DMYSQL\_DATADIR=/usr/local/mysql/data -DSYSCONFDIR=/etc -DWITH\_MYISAM\_STORAGE\_ENGINE=1 -DWITH\_INNOBASE\_STORAGE\_ENGINE=1 -DWITH\_MEMORY\_STORAGE\_ENGINE=1 -DWITH\_READLINE=1 -DMYSQL\_UNIX\_ADDR=/var/lib/mysql/mysql.sock -DMYSQL\_TCP\_PORT=3306 -DENABLED\_LOCAL\_INFILE=1 -DWITH\_PARTITION\_STORAGE\_ENGINE=1 -DEXTRA\_CHARSETS=all -DDEFAULT\_CHARSET=utf8 -DDEFAULT\_COLLATION=utf8\_general\_ci -DWITH\_BOOST=/usr/local/boost\_1\_59\_0**

### 编译好后make

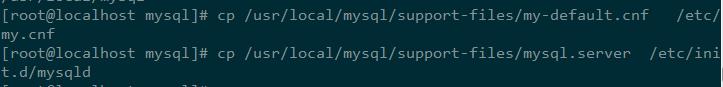


## 部署文件

复制配置文件和服务控制脚本到相应位置

**# cp /usr/local/mysql/support-files/my-default.cnf   /etc/my.cnf**

**# cp /usr/local/mysql/support-files/mysql.server  /etc/init.d/mysqld**



编辑配置文件 /etc/my.cnf，添加：

**basedir = /usr/local/mysql**

**datadir = /usr/local/mysql/data**

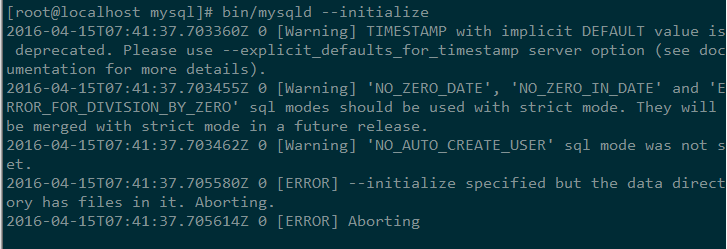
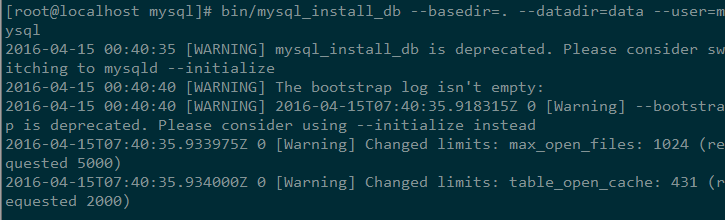
指定安装目录和数据库目录就可以了，其他参数先不用管。如图一所示

接下来就可以初始化数据库了

**cd /usr/local/mysql**

**bin/mysql\_install\_db --basedir=. --datadir=data --user=mysql**

**bin/mysqld --initialize**



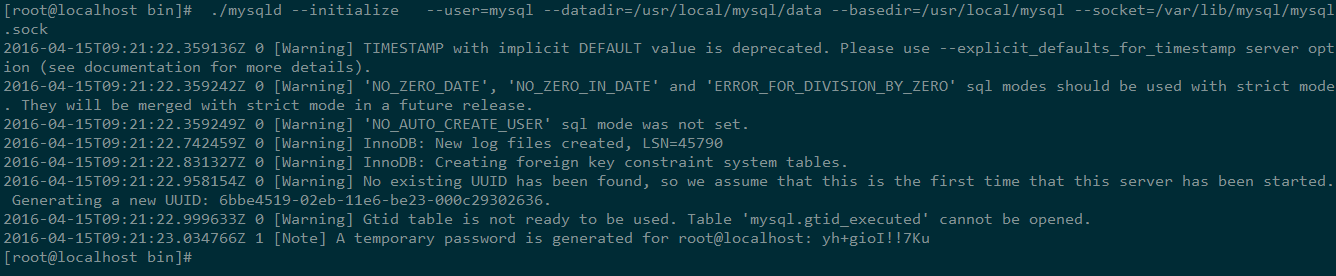
mysql\_install\_db

之前版本mysql\_install\_db是在mysql\_basedir/script下，5.7放在了mysql\_install\_db/bin目录下,且已被废弃。所以执行下面命令会出错。

**# scripts/mysql\_install\_db --user=mysql**

### 初始化数据库

**[root@localhost bin]# ./mysqld --initialize --user=mysql --datadir=/usr/local/mysql/data --basedir=/usr/local/mysql --socket=/var/lib/mysql/mysql.sock**



### 初始化密码

mysql5.7会生成一个初始化密码，而在之前的版本首次登陆不需要登录。

shell> cat /root/.mysql\_secret

# Password set for user 'root@localhost' at 2015-04-22 22:13:23

?G5W&tz1z.cN

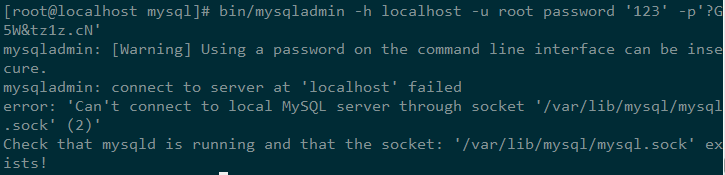
修改默认密码

shell> bin/mysqladmin -h localhost -u root password '123' -p'?G5W&tz1z.cN'

mysqladmin: [Warning] Using a password on the command line interface can be insecure.

Warning: Since password will be sent to server in plain text, use ssl connection to ensure password safety.

**有时执行修改密码时会报下面错误：**

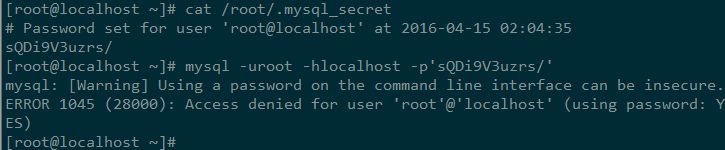


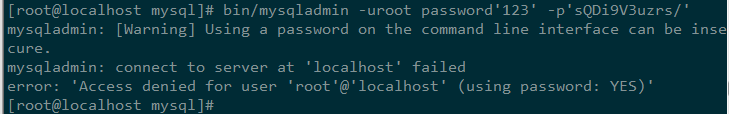
### 启动数据库



## 忘记密码

有时候使用默认密码会出错，初始化密码也会报错





此时就只好初始化密码。方法如下：

**1.修改my.cnf**

 vi /etc/my.cfn

在末尾新起一行加入：

skip-grant-tables

**2.重启数据库**

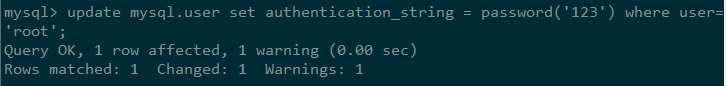
service mysqld restart

**3.连接数据库**

mysql -uroot

**4.修改mysql root 用户的密码**

mysql> update mysql.user set authentication\_string = password('123') where user='root';



mysql> flush privilegs;

**5.恢复第一步中的修改**

vi /etc/my.cfn

去掉第一步中加入的那行代码

**6.再重新登录即可**



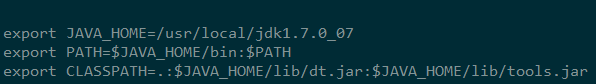
# Tar安装jdk tomcat

先解压



再配置环境变量 vi /etc/profile

在文件末尾输入以下内容



export JAVA\_HOME=/usr/local/jdk1.7.0\_07

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

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

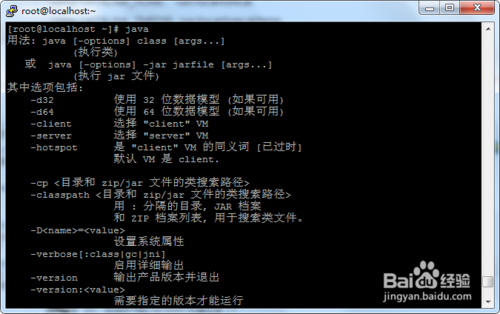
使环境变量立即生效，输入命令：source /etc/profile

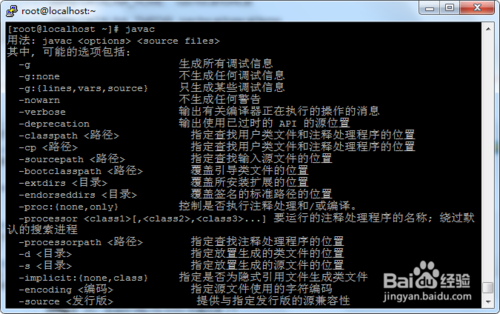
检验是否安装成功

输入命令：java    #检测jdk bin环境

接着输入：javac  #检测jdk classpath环境

成功，你会看到如下java命令参数使用说明界面；

[](http://jingyan.baidu.com/album/ce09321b7c111f2bff858fea.html?picindex=6)

[[](http://jingyan.baidu.com/album/ce09321b7c111f2bff858fea.html?picindex=7)](http://jingyan.baidu.com/album/ce09321b7c111f2bff858fea.html?picindex=7)

安装tomcat

解压tomcat 二进制包，免编译

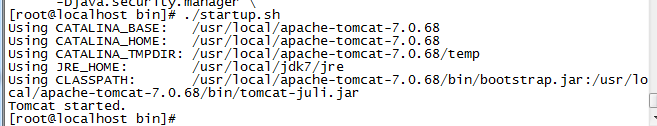
[root@localhost ~]# tar zxvf apache-tomcat-7.0.68.tar.gz

 [root@localhost tomcat7]# cp -r apache-tomcat-7.0.68 tomcat-1

启动tomcat

# cd /usr/local/tomcat7/bin

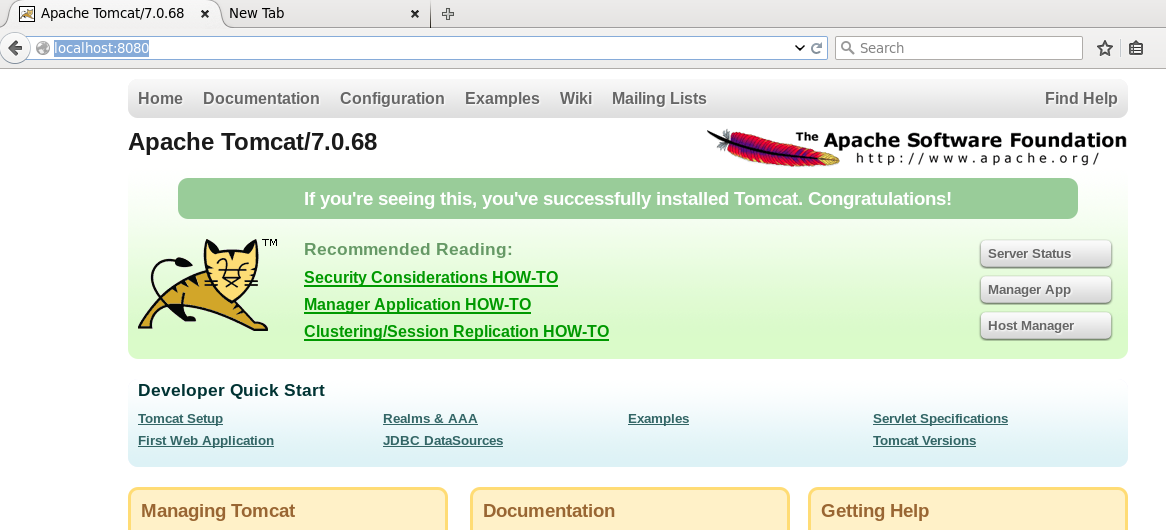
#./startup.sh



测试tomcat

在浏览器输入<http://localhost:8080/>

出现以下画面，说明安装成功



停止Tomcat  
    # ./shutdown.sh

开机自启动

4.配置web管理帐号  
   修改文件conf/tomcat-users.xml，在<tomcat-users>元素中添加帐号密码，需要指定角色.  
   # vi /usr/local/tomcat/server/conf/tomcat-users.xml  
     <tomcat-users>  
       <user name="admin" password="admin" roles="admin-gui,manager-gui" />  
     </tomcat-users>  
  
  
6.配置https安全连接(ssl加密连接)  
   https连接需要用到数字证书与数字签名(MD5算法)  
   网站https连接首先需要申请数字证书，配置加密连接器，浏览器安装证书.  
   ·使用java的工具keytool产生数字证书  
     # keytool -genkey -alias tomcat -keyalg RSA  
     生成文件.keystore  
     注意:CN为主机名称，本机可用localhost  
   ·将文件.keystore放到Tomcat服务器的conf目录下  
     # cp .keystore /usr/local/tomcat/server/conf/  
   ·修改conf/server.xml文件，修改加密连接器，添加keystoreFile与keystorePass  
    <Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"  
               maxThreads="150" scheme="https" secure="true"  
               clientAuth="false" sslProtocol="TLS"   
               keystoreFile="conf/.keystore" keystorePass="123456"/>      
   ·重新启动tomcat.浏览器输入https://localhost:8443访问,并安装证书.

# 编译安装SVN

一，下载相关软件：

[root@youxia201 test]# wge thttp://subversion.tigris.org/downloads/subversion-1.6.1.tar.gz

[root@youxia201 test]# wget http://subversion.tigris.org/downloads/subversion-deps-1.6.1.tar.gz

二，安装及配置：

1，解压，要在同一个目录下：

[root@youxia201 opt]# tar-zxvf subversion-1.6.1.tar.gz

[root@youxia201 opt]# tar-zxvf subversion-deps-1.6.1.tar.gz

2，编译及安装：

[root@localhost ~]# cdsubversion-1.6.1

[root@localhostsubversion-1.6.1]# ./configure  先编译，不指定路径，看是否报错

报错一：

configure: error: in`/root/subversion-1.6.1':

configure: error: no acceptable C compilerfound in $PATH   此错误的意思是没有安装gcc语言

解决方法：yum install gcc  ,安装后再次执行./configure看是否报错

报错二：

configure: error: We requireOpenSSL; try --with-openssl  没有安openssl openssl-devel

解决方法：yum install opensslopenssl-devel, 安装后再次执行./configure看是否报错

报错三：

configure: error: no XML parser was found: expat or libxml 2.x required

解决方法：yum install expat, 如果装不上就源码装安装后再次执行./configure看是否报错,

如果还有错误就根据提示一直解决，不会的百度，直到出现下列正确提示：

Berkeley DB here:

http://www.oracle.com/technology/software/products/berkeley-db/index.html

成功后执行下列操作

[root@youxia201subversion-1.6.1]# ./configure --prefix=/usr/local/svn/

[root@youxia201subversion-1.6.1]# make && make install

3，把svn相关的命令添加到环境变量中：

[root@youxia201subversion-1.6.1]# echo "export PATH=$PATH:/usr/local/svn/bin/">> /etc/profile

[root@youxia201subversion-1.6.1]# source /etc/profile

[root@localhost subversion-1.6.1]# cd /usr/local/svn/

[root@localhost svn]# ls  看到下面有内容，说明安装成功

bin  build-1 include  lib  share

[root@localhost ~]# /usr/local/svn/bin/svnserve  --version 查看SVN版本信息，出现版本信息为正确

三，建立测试仓库：

1，建立SVN的根目录，研发中心有多个项目部：

[root@youxia201subversion-1.6.1]# mkdir -p /opt/svn/

2，建立一个测试仓库：

[root@youxia201subversion-1.6.1]# mkdir -p /opt/svn/svntest/

[root@youxia201subversion-1.6.1]# svnadmin create /opt/svn/svntest/

3，修改配置文件：

[root@localhost ~]# cd /opt/svn/svntest/

[root@localhost svntest]# ls

conf  db  format hooks  locks  README.txt

[root@youxia201 svntest]# cd /opt/svn/svntest/conf/

[root@youxia201 conf]# ll

总计 24

-rw-r--r-- 1 root root 710 08-25 09:40 authz

-rw-r--r-- 1 root root 325 08-25 09:38 passwd

-rw-r--r-- 1 root root 144908-25 09:36 svnserve.conf

[root@youxia201 conf]# visvnserve.conf   修改svn配置文件

[general]

anon-access = none # 使非授权用户无法访问  
auth-access = write # 使授权用户有写权限  
password-db = password#密码数据库的路径  
authz-db = authz   # 访问控制文件  
以上语句都必须顶格写, 左侧不能留空格, 否则会出错.

[root@admin conf]# vim passwd  
修改passwd为以下内容：

[users]  
# harry = harryssecret  
# sally = sallyssecret

hello=123

aaa = 123

www = 123

用户名=密码

这样我们就建立了hello用户， 123密码

以上语句都必须顶格写, 左侧不能留空格, 否则会出错.

[root@youxia201 conf]# viauthz   指定svn用户拥有的目录权限

[groups]

admin = hello，www

[svntest:/]

@admin = rw

aaa = r

将用户添加到组，利用组来设置权限，也可以不需要租，直接在版本库下设置用户，例如

aaa = r ,意思是aaa用户对svntest测试库下所有的目录有读权限,而hello和www有读写权限。  
如果是自己用，就直接是读写吧。

以上语句都必须顶格写, 左侧不能留空格, 否则会出错.

[root@localhost conf]# svnserve -d -r /opt/svn/svntest/   启动SVN测试库

如果已经有svn在运行，可以换一个端口运行  
svnserve -d -r /opt/svn/ --listen-port 3391

四，启服务器及测试：

1，启SVN服务，并指定SVN的根目录：

[root@youxia201 test]#svnserve -d -r /opt/svn/

在这里特别的要注意，/opt/svn是仓库的根目录，不要和[svntest:/]目录重叠了。如果重叠是会提示错误

[root@localhost conf]# svn co svn://127.0.0.1   在本机上测试svn是否生效

|  |
| --- |
| 认证领域: <svn://127.0.0.1:3690>ed4c06a0-3c8f-11e3-b7ad-e704fa0b2aec  用户名: hello  “hello”的密码:  -----------------------------------------------------------------------  注意!  你的密码，对于认证域:   <svn://127.0.0.1:3690> ed4c06a0-3c8f-11e3-b7ad-e704fa0b2aec  只能明文保存在磁盘上! 如果可能的话，请考虑配置你的系统，让 Subversion  可以保存加密后的密码。请参阅文档以获得详细信息。  你可以通过在“/root/.subversion/servers”中设置选项“store-plaintext-passwords”为“yes”或“no”，  来避免再次出现此警告。  -----------------------------------------------------------------------  保存未加密的密码(yes/no)?y  请输入 'yes' 或 'no': yes  取出版本 1。 |

不让它每次输入都提示yes/on的办法

编辑vim /root/.subversion/servers  找到下行去掉#号将on改为yes，上面绿色部分已提示 store-plaintext-passwords = no

2，服务已经正常起来：

[root@localhost conf]# netstat -anpl |grep svn

tcp        0     0 0.0.0.0:3690               0.0.0.0:\*                   LISTEN      318/svnserve

---------------------------------------------------------------------------------------------------------------------

到此，SVN服务端就配置完成了

**Linux svn客户端配置**

**检查已安装版本**

#检查是否安装了低版本的SVN  
[root@localhost /]#rpm -qa subversion

#如果存储旧版本，卸载旧版本SVN  
[root@localhost modules]#yum remove subversion

**安装SVN**

[root@localhost modules]#yum install subversion

**验证安装**

检验已经安装的SVN版本信息  
[root@localhost modules]# svnserve --version

svnserve，版本 1.6.11 (r934486)  
编译于 Jun 23 2012，00:44:03

版权所有 (C) 2000-2009 CollabNet。  
Subversion 是开放源代码软件，请参阅 http://subversion.tigris.org/ 站点。  
此产品包含由 CollabNet(http://www.Collab.Net/) 开发的软件。

下列版本库后端(FS) 模块可用:

\* fs\_base : 模块只能操作BDB版本库。  
\* fs\_fs : 模块与文本文件(FSFS)版本库一起工作。

Cyrus SASL 认证可用。

**SVN客户端下载文件**

[root@localhost ~]# mkdir /svn

[root@localhost ~]# cd /svn/        #将svn服务上提取的文件放在此目录下

[root@localhost svn]# svn checkoutsvn://192.168.233.203/repo              #svn服务器地址

输入用户名密码：hello/123

注:可以使用带用户名和密码的访问:

svn co --username hello --password 123svn://192.168.233.203/repo

**SVN版本库更新**

svn up /svn/repo

**上传文件到服务器**

[root@localhost ]# touch /svn/repo/test.php        #所有的动作必须在repo版本库中完成

[root@localhost ]# svn add /svn/repo/test.php           #上传test.php文件到svn服务器

**删除服务器上的文件**

[root@localhost ]# svn rm /svn/repo/test.php --force

**查看版本库内容**

[root@localhost ~]# svn list /svn/repo/

------------------------------------------------------------------------------------------------------------------------

到此，SVN客户端就配置完成了

**创建版本库：**  
终端输入：svnadmin create [路径]  
该路径可以是相对路径，例如，当前在home/user/svn/目录下，输入svnadmin create myproject，则该版本库会被创建在home/user/svn/目录下，新建一个文件夹名为myproject，其中便是版本库的文件了。  
  
**将文件导入版本库：**  
终端输入：svn import [源路径] [目标版本库路径] -m [日志信息]  
源路径可以是相对路径，导入时会递归导入源路径下的所有文件和文件夹，目标版本库路径需要绝对目录（反正我试验是这样的），例如版本库的目录是：home/user/svn/myproject/。则应该这样写：file:///home/user/svn/myproject/  
例如将当前目录导入版本库myproject：  
svn import . file:///home/user/svn/myproject -m"导入文件"  
  
**从版本库导出：**  
导入后原文件并未被纳入版本管理，若想获得受版本控制的文件，就需要从版本库导出  
终端输入：svn co [版本库路径]   
例如将myproject库中的文件导出到当前目录：svn co file:///home/user/svn/myproject .

五，在WindowsXP上安装svn客户端：TortoiseSVN(SVN客户端)

1，英文版下载地址：

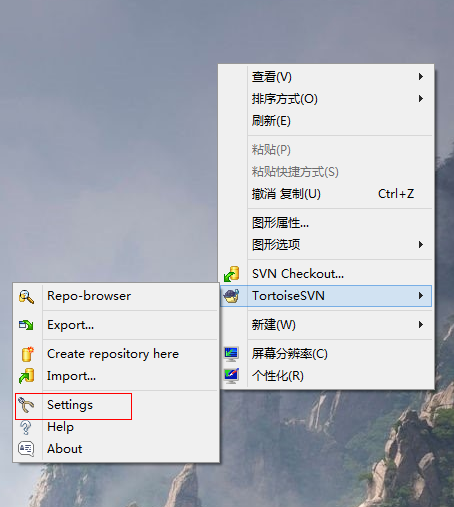
http://tortoisesvn.net/downloads.html

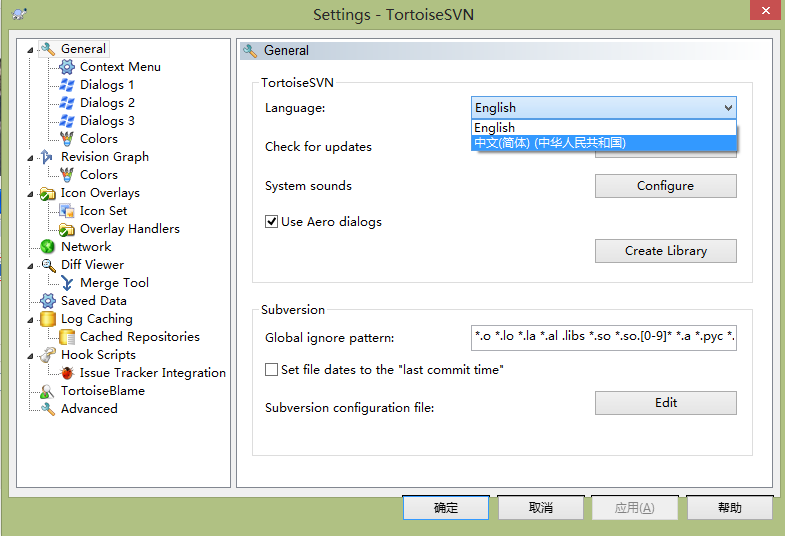
2，中文版安装

下载完成后，直接next安装即可，安装完成后需要重启生效（自行选择32位于64位版本）。，

http://www.centoscn.com/uploads/allimg/20140202/13913388739687.png

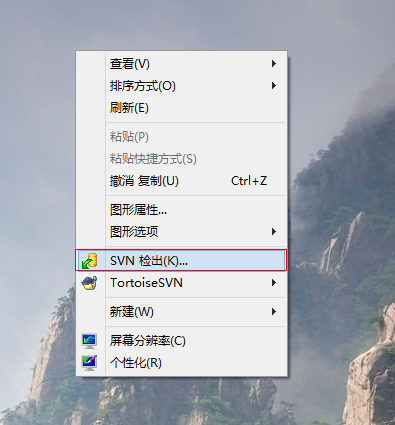
安装中文汉化包后会出现中文版选择项，进行下面设置成中文版



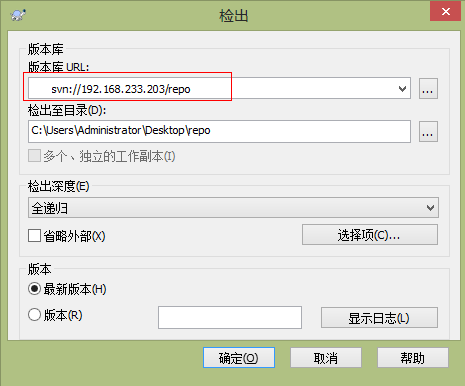


3，测试：

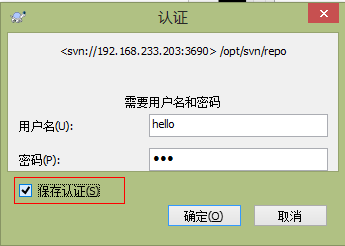
在桌面上新建一个名称为svntest的目录，在此目录上点击右键，选择Checkout，在首行填写svn服务器的IP地址及仓库名称



输入svn服务器ip地址和要连接的版本库，输出目录可以根据自己的爱好更改路径

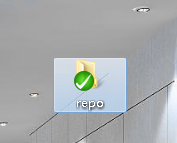


输入连接服务器端的用户名称及密码后就可以使用了

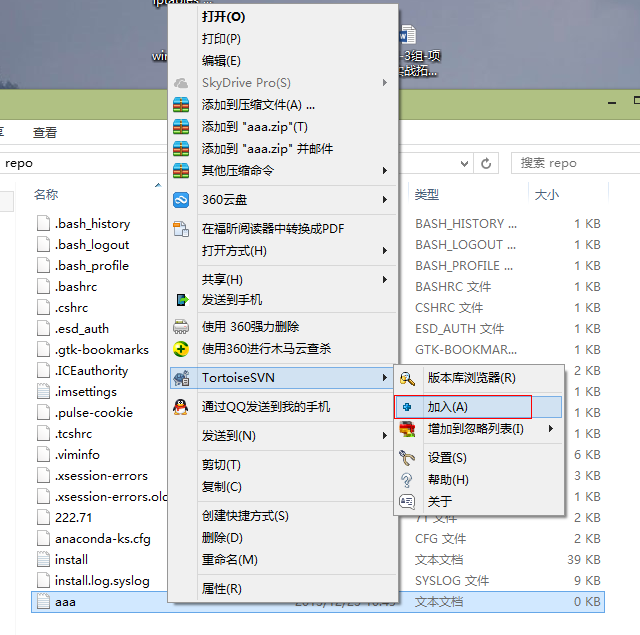




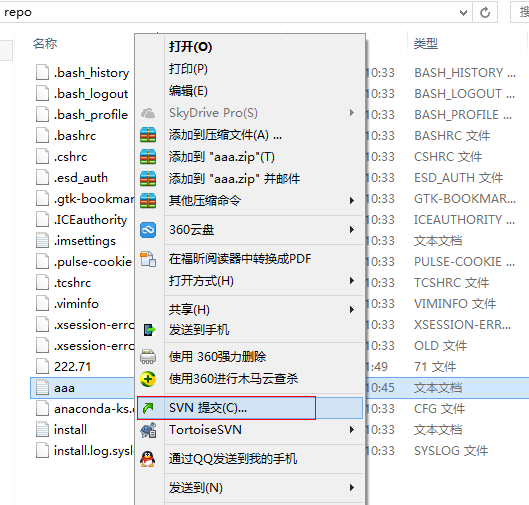
检查后的文件夹样子



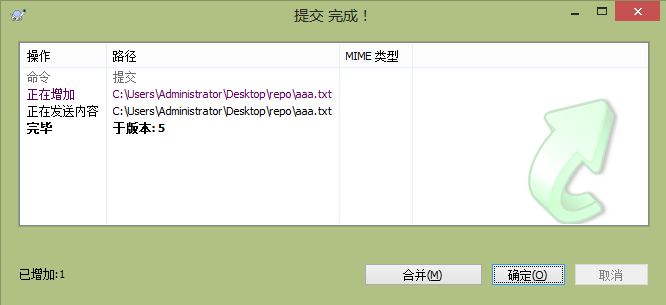
将windows上要添加的文件移动至repo文件夹进行上传，上传哪个文件就点击右键—添加



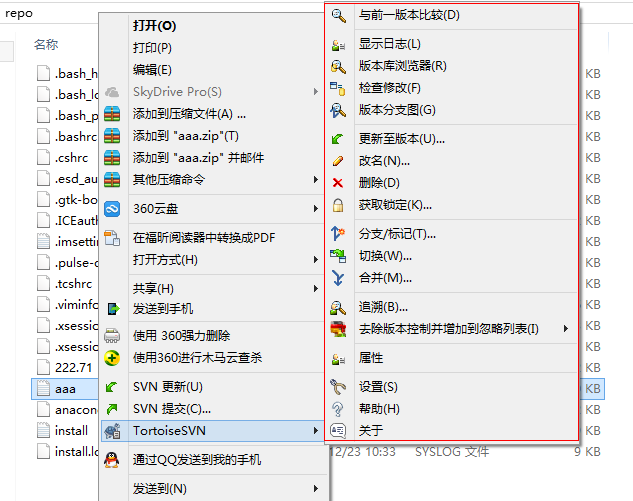
添加完成后，再次右键—提交







提交后可以查看一下这个文件都可以进行哪些操作



Windows svn客户端到此就结束了，其余功能自己去摸索吧，再次不在赘述了

# 查看端口被哪个进程占用

netstat -an | grep 8080





Listen就是被占用

netstat -lnp|grep 8080可以看出被哪个程序和进程占用



# 删除软件

rpm -e --nodeps java-1.8.0-openjdk-1.8.0.71-1.b15.el6\_7.i686

删除文件夹

rm -rf /etc/tomcat6

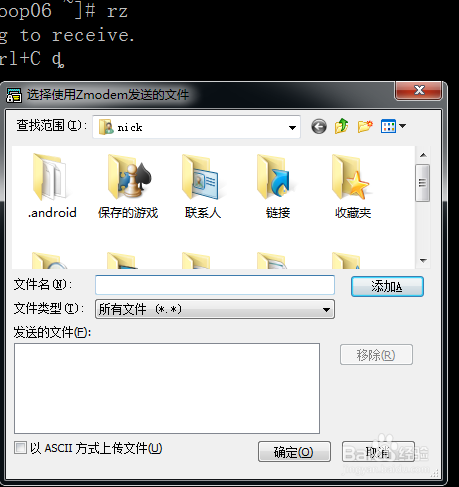
java wget  http://211.149.198.47/data/main/jdk-7u7-linux-i586.tar.gz

# SecureCRT中给linux上传和下载文件

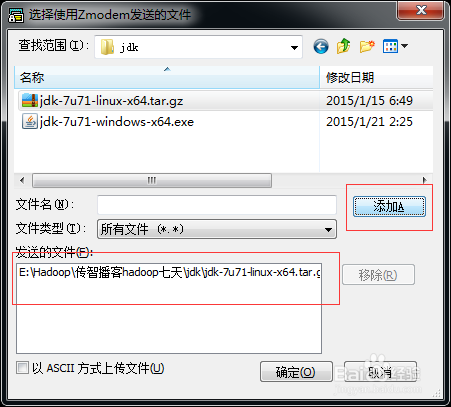
1. 需要上传或者下载，需要使用rz和sz命令。如果linux上没有这两个命令工具，则需要先安装。可以使用yum安装。运行命令yum install lrzsz。

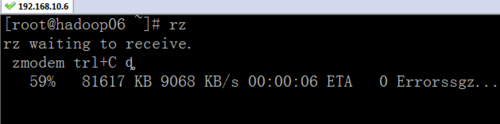


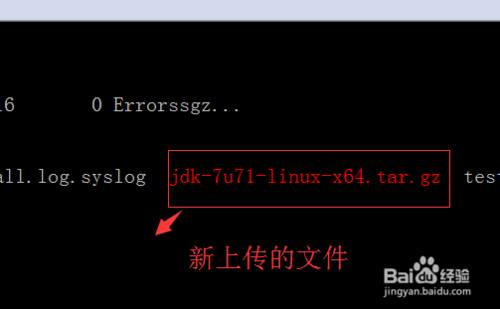
1. 安装完成后就可以使用rz命令上传本地的文件到linux平台。运行rz命令，会弹出选择文件的对话框。

[](http://jingyan.baidu.com/album/1709ad8096e7934634c4f020.html?picindex=2)

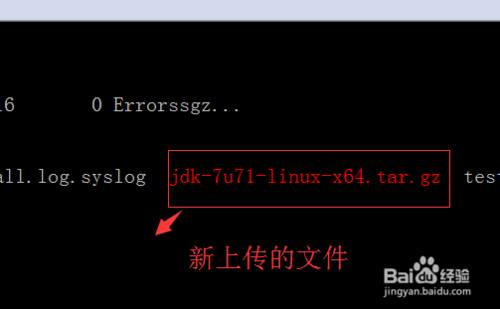
1. 选择需要上传的文件所在的文件夹，然后点击文件。然后再点击添加按钮，“发送的文件”框中就会显示需要上传的文件。点击确定按钮，就能完成上传。

[](http://jingyan.baidu.com/album/1709ad8096e7934634c4f020.html?picindex=3)

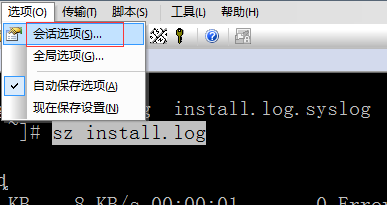
[](http://jingyan.baidu.com/album/1709ad8096e7934634c4f020.html?picindex=4)

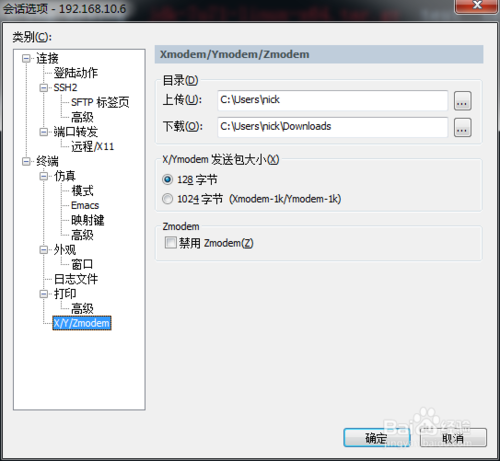
[](http://jingyan.baidu.com/album/1709ad8096e7934634c4f020.html?picindex=5)

1. 如果需要下载linux上的文件，使用sz命令。sz加上文件完整路径，比如：sz install.log。这样就下载到了本地一个默认的文件夹。

[](http://jingyan.baidu.com/album/1709ad8096e7934634c4f020.html?picindex=6)

1. 使用sz上传的时候，选择框会事先打开在是一个默认的目录。使用sz下载时，也会下载到一个默认目录。这个默认目录时可以设置的。
2. 需要设置上传和下载的默认目录，点击SecureCRT的菜单栏中的 选项=》会话选项。打开会话选项对话框，点击分类中的 终端=》X/Y/Zmodem。

[](http://jingyan.baidu.com/album/1709ad8096e7934634c4f020.html?picindex=7)

[](http://jingyan.baidu.com/album/1709ad8096e7934634c4f020.html?picindex=8)

1. 在对话框的右边有设置目录，这个就是上传和下载的默认的目录。修改成需要的目录后，点击确定即可。

[](http://jingyan.baidu.com/album/1709ad8096e7934634c4f020.html?picindex=9)

END

## 注意事项

* 上传的时候，如果上传到的linux目录有同名的文件，是无法上传的，需要先删掉linux上的同名文件。