

本文主要介绍的是如何在Linux环境下安装JDK的，因为Linux环境下，很多时候也离不开Java的，下面就和大家一起分享如何安装jdk1.8的过程吧。

一、安装环境

操作系统：Centos6.8

JDK版本：1.8

工具：Xshell6、WinSCP

说明：本文是通过Xshell6工具远程连接Linux操作，如果是直接在Linux可视化界面操作那就更方便了，原理一样。

二、安装步骤

第一步：下载安装包

下载Linux环境下的jdk1.8，请去（[官网](#)）中下载jdk的安装文件；

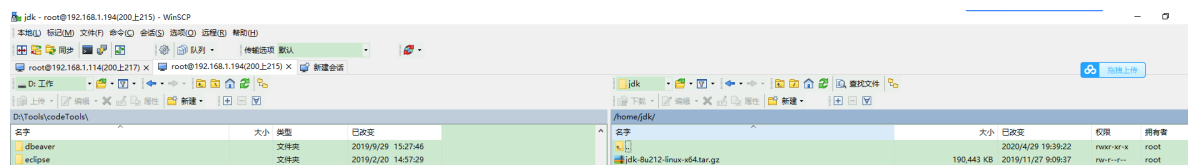
jdk-8u212-linux-x64.tar.gz百度云下载地址：

链接：https://pan.baidu.com/s/1lSa_vlonXCg65PG6UgYrNw

提取码：azqs

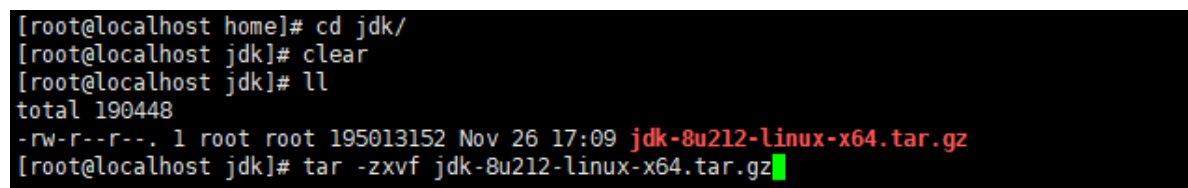
第二步：解压安装包

将我们下载好的JDK安装包上传到服务器，进行解压

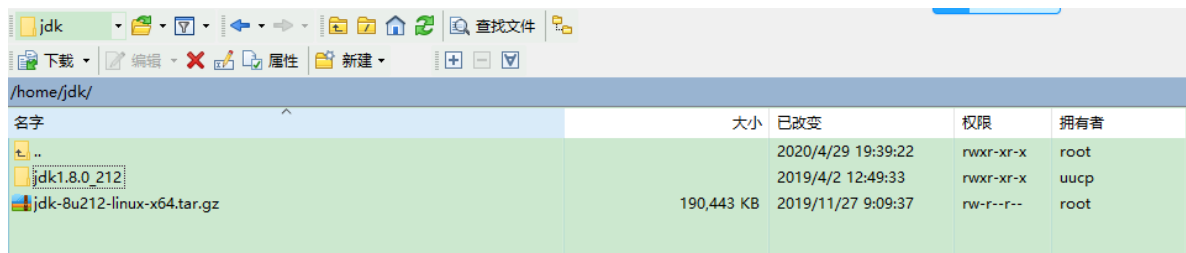


解压命令进行解压：

```
tar -zxvf jdk-8u212-linux-x64.tar.gz
```



解压完成之后，可以在当前目录下看到一个名字为【jdk1.8.0_212】的目录，里面存放的是相关文件。



名字	大小	已改变	权限	拥有者
..		2020/4/29 19:39:22	rw-r--r--	root
jdk1.8.0_212		2019/4/2 12:49:33	rw-r--r--	uucp
jdk-8u212-linux-x64.tar.gz	190,443 KB	2019/11/27 9:09:37	rw-r--r--	root

我们要将解压后的【jdk1.8.0_212】里面的所有数据移动到我们需要安装的文件夹当中，我们打算将jdk安装在/usr/java当中，我们在usr目录下新建一个java文件夹。

```
mkdir /usr/java
```

将【jdk1.8.0_212】里的数据拷贝至java目录下。

```
mv /home/jdk/jdk1.8.0_212 /usr/java
```

```
[root@localhost jdk]# mkdir /usr/java
[root@localhost jdk]# mv /home/jdk/jdk1.8.0_212 /usr/java
[root@localhost jdk]# cd /usr/java/
[root@localhost java]# ll
total 4
drwxr-xr-x. 7 uucp 143 4096 Apr  1 2019 jdk1.8.0_212
[root@localhost java]#
```

第三步、修改环境变量

至此，我们最后需要修改环境变量，通过命令：

```
vim /etc/profile
```

用vim编辑器来编辑profile文件，在文件末尾添加一下内容（按“i”进入编辑）：

```
export JAVA_HOME=/usr/java/jdk1.8.0_212
export JRE_HOME=${JAVA_HOME}/jre
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib:${CLASSPATH}
export JAVA_PATH=${JAVA_HOME}/bin:${JRE_HOME}/bin
export PATH=$PATH:${JAVA_PATH}
```

如图所示：

```
fi
HOSTNAME="/bin/hostname 2>/dev/null"
HISTSIZE=1000
if [ "$HISTCONTROL" = "ignoreboth" ]; then
    export HISTCONTROL=ignoreboth
else
    export HISTCONTROL=ignorespaces
fi
export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE HISTCONTROL

# By default, we want umask to get set. This sets it for login shell
# current threshold for system reserved uids/gids is 200
# You could check uidgid reservation validity in
# /usr/share/doc/setup-*/uidgid file
if [ $(SUID -gt 100) ] && [ "$id -gn" = "" -o "$id -un" = "" ]; then
    umask 002
else
    umask 022
fi
for i in /etc/profile.d/*.sh; do
    if [ -r "$i" ]; then
        if [ "${i%#}" != "$i" ]; then
            . "$i"
        else
            "$i" >/dev/null 2>&1
        fi
    fi
done
unset i
unset -f pathmunge
if [ $USER = "oracle" ]; then
    if [ $SHELL = "/bin/ksh" ]; then
        ulimit -p 16384
        ulimit -n 65536
    else
        ulimit -u 16384 -n 65536
    fi
fi
fi
export JAVA_HOME=/usr/java/jdk1.8.0_212
export JRE_HOME=${JAVA_HOME}/jre
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib:$CLASSPATH
export JAVA_PATH=${JAVA_HOME}/bin:${JRE_HOME}/bin
export PATH=$PATH:${JAVA_PATH}
-- INSERT --
```

然后，保存并退出(按：wq!)

保存完之后，我们还需要让这个环境变量配置信息里面生效，要不然只能重启电脑生效了。

通过命令source /etc/profile让profile文件立即生效，如图所示：

```
source /etc/profile
```

```
[root@localhost jdk1.8.0_212]# vim /etc/profile
[root@localhost jdk1.8.0_212]#
[root@localhost jdk1.8.0_212]# source /etc/profile
[root@localhost jdk1.8.0_212]#
```

第四步、测试是否安装成功

- ①、使用javac命令，不会出现command not found错误
- ②、使用java -version，出现版本为java version "1.8.0_212"
- ③、echo \$PATH，看看自己刚刚设置的的环境变量配置是否都正确

如图所示：

```
[root@localhost jdk1.8.0_212]# javac
Usage: javac <options> <source files>
where possible options include:
-g Generate all debugging info
-g:none Generate no debugging info
-g:{lines,vars,source} Generate only some debugging info
-nowarn Generate no warnings
-verbose Output messages about what the compiler is doing
-deprecation Output source locations where deprecated APIs are used
-classpath <path> Specify where to find user class files and annotation processors
-cp <path> Specify where to find user class files and annotation processors
-sourcepath <path> Specify where to find input source files
-bootclasspath <path> Override location of bootstrap class files
-extdirs <dirs> Override location of installed extensions
-endorseddirs <dirs> Override location of endorsed standards path
-processor <{none,only}> Control whether annotation processing and/or compilation is done.
-processor <class1>[,<class2>,<class3>...] Names of the annotation processors to run; bypasses default discovery process
-processorpath <path> Specify where to find annotation processors
-parameters Generate metadata for reflection on method parameters
-d <directory> Specify where to place generated class files
-s <directory> Specify where to place generated source files
-h <directory> Specify where to place generated native header files
-implicit:{none,class} Specify whether or not to generate class files for implicitly referenced files
-encoding <encoding> Specify character encoding used by source files
-source <release> Provide source compatibility with specified release
-target <release> Generate class files for specific VM version
-profile <profile> Check that API used is available in the specified profile
-version Version information
-help Print a synopsis of standard options
-Akey[=value] Options to pass to annotation processors
-X Print a synopsis of nonstandard options
-J<flag> Pass <flag> directly to the runtime system
-Werror Terminate compilation if warnings occur
@filename Read options and filenames from file

[root@localhost jdk1.8.0_212]# java -version
java version "1.8.0_212"
Java(TM) SE Runtime Environment (build 1.8.0_212-b10)
Java HotSpot(TM) 64-Bit Server VM (build 25.212-b10, mixed mode)
[root@localhost jdk1.8.0_212]# echo $PATH
/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin:/usr/java/jdk1.8.0_212/bin:/usr/java/jdk1.8.0_212/jre/bin
[root@localhost jdk1.8.0_212]#
```

至此，安装结束。

三、Linux下切换使用两个版本的JDK

我这里原来已经配置好过一个1.7版本的jdk。

输出命令: `java -version`

```
[root@localhost ~]# java -version
java version "1.7.0_79"
Java(TM) SE Runtime Environment (build 1.7.0_79-b15)
Java HotSpot(TM) 64-Bit Server VM (build 24.79-b02, mixed mode)
```

java JDK1.8_212安装过后，会发现jdk仍然是1.7的版本。用如下命令查看已配置好的jdk：

```
alternatives --config java
```

```
1 /home/bigdata/jdk1.7.0_79/bin/java
```

会发现jdk1.8_212没有配置，ctrl + c 退出，使用如下命令,将刚安装的jdk1.8_212加入配置：

```
alternatives --install /usr/bin/java java /usr/java/jdk1.8.0_212/bin/java 2
```

如果添加错误可以使用以下命令删除：

```
alternatives --remove java usr/java/jdk1.8.0_212/bin
```

然后使用以下命令切换两个JDK的使用，选择刚配置的jdk1.8_212，输入配置时的编号2即可：

```
alternatives --config java
```

然后再输入命令，即可查看到jdk已经切换了：

```
java -version
```

注意：

如果切换完成后发现版本不能改变时，你可以将 /etc/profile 中的JAVA_HOME注释掉，然后重新加载一下配置文件：source /etc/profile。

发现可以了，java版本切换成功了。

然后你再把配置文件改回来，重新加载，然后就可以了。