

Linux 下安装JDK1.8

原文链接:<https://www.cnblogs.com/xuliangxing/p/7066913.html>

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，下面笔者就和大家一起分享如何jdk1.8的过程吧。

一、安装环境

操作系统：Red Hat Enterprise Linux 6 64 位(版本号6.6)

JDK版本：1.8

工具：Xshell5、Xftp5

说明：本文是通过Xshell5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

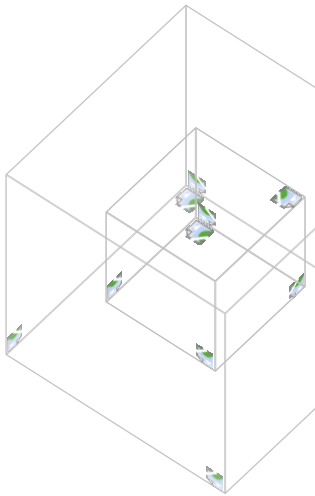
第一步：下载安装包

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

由于我的Linux是64位的，因此我下载[jdk-8u131-linux-x64.tar.gz](#)。

如下图所示：

公告



希望大家多多支持，觉得文章好可以点个赞，动动你的鼠标加下关注哦

我的访客量统计：Start From 2018.1.1 19:00



昵称：枫树湾河桥
园龄：2年3个月
粉丝：24
关注：17
[+加关注](#)

- 一、安装环境
- 二、安装步骤

OverviewDownloadsDocumentationCommunityTechnologiesTraining

Fork me on GitHub

Java SE Development Kit 8 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- Java Developer Newsletter: From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to **Java**.
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK 8u131 checksum

<http://www.cnblogs.com/xuliangxing>

Java SE Development Kit 8u131

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

☐ Accept License Agreement☒ Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.87 MB	jdk-8u131-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.81 MB	jdk-8u131-linux-arm64-vfp-hflt.tar.gz
Linux x86	164.66 MB	jdk-8u131-linux-i586.rpm
Linux x86	179.39 MB	jdk-8u131-linux-i586.tar.gz
Linux x64	162.11 MB	jdk-8u131-linux-x64.rpm
Linux x64	176.95 MB	jdk-8u131-linux-x64.tar.gz
Mac OS X	226.57 MB	jdk-8u131-macosx-x64.dmg
Solaris SPARC 64-bit	139.79 MB	jdk-8u131-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.13 MB	jdk-8u131-solaris-sparcv9.tar.gz
Solaris x64	140.51 MB	jdk-8u131-solaris-x64.tar.Z
Solaris x64	96.96 MB	jdk-8u131-solaris-x64.tar.gz
Windows x86	191.22 MB	jdk-8u131-windows-i586.exe
Windows x64	198.03 MB	jdk-8u131-windows-x64.exe

积分 - 115018
排名 - 5675

杭州
23°

随笔分类

- dubbo(3)
- error(16)
- idea(8)
- java基础(14)
- linux(5)
- mongodb(7)
- MyBatis(8)
- mysql(19)
- springboot(21)
- 并发安全(15)
- 分布式(25)
- 集合(7)
- 计算机网络(3)
- 面试(5)
- 前端(2)
- 设计模式(8)
- 数据结构与算法(1)

随笔档案

- 2020年4月(2)
- 2020年3月(3)
- 2020年2月(1)
- 2020年1月(14)
- 2019年12月(8)
- 2019年11月(13)
- 2019年10月(4)
- 2019年9月(2)
- 2019年8月(1)
- 2019年7月(12)
- 2019年6月(3)
- 2019年5月(1)
- 2019年4月(8)
- 2019年1月(1)
- 2018年12月(2)
- 2018年11月(7)
- 2018年10月(10)
- 2018年9月(16)
- 2018年8月(27)
- 2018年7月(1)
- 2018年6月(10)
- 2018年5月(4)
- 2018年4月(10)
- 2018年3月(10)

一、安装环境
二、安装步骤

如果Linux本身连接到互联网，我们可以直接通过wget命令直接把JDK安装包下载下来，如图所示：

```
[root@localhost ~]# wget http://download.oracle.com/otn-pub/java/jdk/8u131-b11/d54c1d5a695b4f2b6667095fa0163/jdk-8u131-linux-x64.tar.gz?AuthParam=1499844927_cc3bd7a296f188e960197848c4b7d7
--2017-07-12 08:33:37-- http://download.oracle.com/otn-pub/java/jdk/8u131-b11/d54c1d5a695b4f2b6667095fa0163/jdk-8u131-linux-x64.tar.gz?AuthParam=1499844927_cc3bd7a296f188e960197848c4b7d7
Resolving download.oracle.com... 111.108.54.30, 111.108.54.17
Connecting to download.oracle.com[111.108.54.10]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 185540433 (177M) [application/x-gzip]
Saving to: 'jdk-8u131-linux-x64.tar.gz?AuthParam=1499844927_cc3bd7a296f188e960197848c4b7d7'
100% [*****] 24.913K 125 147K/s eta 7h 20s
```

要是没有外网的环境，还是安装上面的方法下载安装包，然后上传到服务器当中

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

192.168.8.178

/home/cmfchina

名称	大小	类型	修改时间	属性	所有者
..					
Desktop		文件夹	2017-06-20, 13:56	drwxr-xr-x	cmfchina
Documents		文件夹	2017-06-20, 13:56	drwxr-xr-x	cmfchina
Downloads		文件夹	2017-06-20, 13:56	drwxr-xr-x	cmfchina
Music		文件夹	2017-06-20, 13:56	drwxr-xr-x	cmfchina
openssl-1.0.2l		文件夹	2017-07-10, 16:38	drwxr-xr-x	root
opt		文件夹	2017-06-20, 14:27	drwxrwxr-x	cmfchina
Pictures		文件夹	2017-06-20, 13:56	drwxr-xr-x	cmfchina
Public		文件夹	2017-06-20, 13:56	drwxr-xr-x	cmfchina
redis		文件夹	2017-07-10, 13:44	drwxrwxr-x	cmfchina
ruby-2.4.1		文件夹	2017-03-22, 13:56	drwxr-xr-x	1001
rubygems-2.6.12		文件夹	2017-05-01, 7:28	drwxr-xr-x	501
Templates		文件夹	2017-07-12, 15:43	drwxr-xr-x	cmfchina
Videos		文件夹	2017-06-20, 13:56	drwxr-xr-x	cmfchina
zlib-1.2.11		文件夹	2017-07-10, 15:59	drwxr-xr-x	501
jdk-8u131-linux-x64.tar.gz	176.95MB	gz Archive	2017-06-22, 14:55	-rw-rw-r--	cmfchina

<http://www.cnblogs.com/xuliangxing>

解压命令进行解压

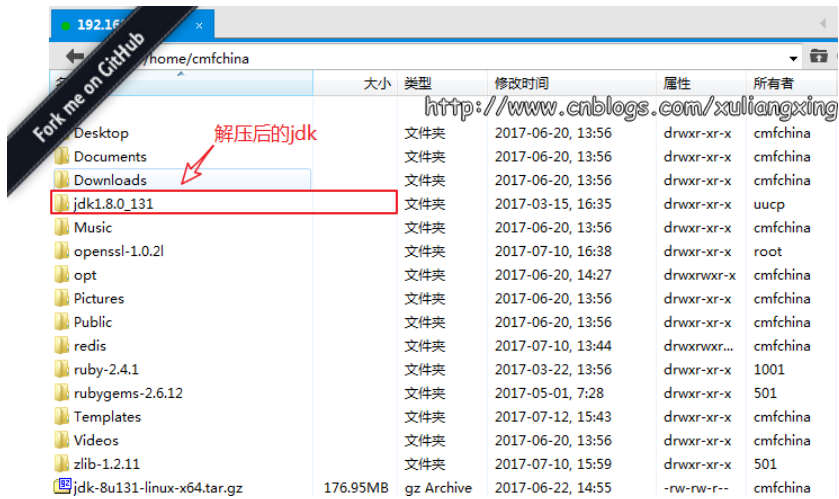
```
1 $ cd /home/cmfchina
2 $ tar -zxvf jdk-8u131-linux-x64.tar.gz

[root@localhost cmfchina]# cd /home/cmfchina
[root@localhost cmfchina]# tar -zxvf jdk-8u131-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_131】的目录，里面存放的是相关文件

https://www.cnblogs.com/fswhq/p/10713429.html

2/5



我们要将解压后的【jdk1.8.0_131】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

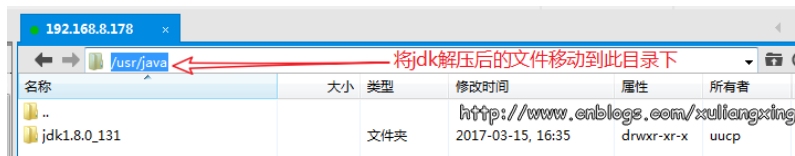
```
mkdir /usr/java
```

```
[root@localhost usr]# mkdir /usr/java
```

将【jdk1.8.0_131】里的数据拷贝至java目录下

```
mv /home/cmfchina/jdk1.8.0_131 /usr/java
```

```
[root@localhost usr]# mv /home/cmfchina/jdk1.8.0_131/ /usr/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
```

```
[root@localhost jdk1.8.0_131]# 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）：

```
1 export JAVA_HOME=/usr/java/jdk1.8.0_131
2 export JRE_HOME=${JAVA_HOME}/jre
3 export
  CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib:$CLASSPATH
4 export JAVA_PATH=${JAVA_HOME}/bin:${JRE_HOME}/bin
5 export PATH=$PATH:${JAVA_PATH}
```

如图所示：

4. Re:AOP拦截日志报错
IllegalStateException: It is illegal
method if the current request is

asynchronous mode
博客UI太棒啦

--MartinDon

5. Re:SpringBoot浏览器直接访问html

@ 甘劭网上一大堆...

--枫树湾河桥

6. Re:SpringBoot浏览器直接访问html

你的博客太炫彩了，有模版吗，借鉴借鉴

--甘劭

7. Re:单列索引和联合索引区别

博客很好看，差点忘了我点进来是为了学习了

--Frankie_Zhen

8. Re:java volatile详解

@ 枫树湾河桥你这个妹子控件也太强了...

--黑夜下雨

9. Re:SpringBoot实现JWT保护前后端分离
RESTful API

@ 青春不止、奋斗不息private boolean

isExceededUrl(HttpServletRequest
request){return pathMatcher.match("/log...

--枫树湾河桥

10. Re:SpringBoot实现JWT保护前后端分离
RESTful API

isExceededUrl 过滤器就根本没这方法

--青春不止、奋斗不息

阅读排行榜

1. Expression #1 of SELECT list is not in GR
OUP BY clause and contains nonaggre(250
97)
2. SpringBoot浏览器直接访问html(17157)
3. SpringBoot实现JWT保护前后端分离RES
Tful API(17132)
4. jpa batch批量操作save和persist比较(875
0)
5. 23种设计模式详解(8369)
6. AOP拦截日志报错IllegalStateException: It
is illegal to call this method if the current re
quest is not in asynchronous mode(7413)
7. mybatis 日期查询datetime(6713)
8. java volatile详解(6598)
9. UML之时序图详解(6551)
10. Linux 下安装JDK1.8(6217)
11. springboot jpa mongodb 多条件分页查

一、安装环境

二、安装步骤

Fork me on GitHub

```
if [ "$HISTCONTROL" = "ignoreblank" ]; then
    export HISTCONTROL=ignoreboth
else
    export HISTCONTROL=ignoreblanks
fi

# By default, we want umask to get set. This sets it for login shell
current threshold for system reserved uid/gids is 200
# You could check uidgid reservation validity in
# /usr/share/doc/setup-*/uidgid file
if [ $(id -u) -gt 199 ] && [ "$(id -gn)" = "$(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
export JAVA_HOME=/usr/java/jdk1.8.0_131
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}:${SCALA_HOME}/bin:${SPARK_HOME}/bin:${SPARK_HOME}/sbin:/usr/local/ruby/bin

-- INSERT --
```

后面这些是其他程序的环境变量，这是之前系统设置了其他程序环境变量，我们不要动他们，我们以：分隔就可以了

设置jdk的环境变量

<http://www.cnblogs.com/xuliangxing>

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

```
[root@localhost jdk1.8.0_131]# source /etc/profile
```

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_131"
- 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：

```
v[root@localhost ~]# 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
  -proc:{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
```

javac信息

java版本信息

<http://www.cnblogs.com/xuliangxing>

分类: linux

好文要顶 关注我 收藏该文



枫树湾河桥

关注 - 17

粉丝 - 24

+加关注

- « 上一篇: INSERT IGNORE与INSERT INTO的区别
- » 下一篇: maven中archetype(原型)的使用

posted @ 2019-04-15 21:42 枫树湾河桥 阅读(6217) 评论(0) 编辑 收藏

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 登录 或 注册， 访问 网站首页。

- 24. ConcurrentHashMap get方法
- 25. 打扮IDEA更换主题(1860)
- 26. jdbcType DATE和TIMESTAMP区别(1765)
- 27. BigDecimal 比较equals与compareTo(1548)
- 28. mysql 保存23:59:59 自动加一秒(1461)
- 29. java根据模板生成pdf(1409)
- 30. SpringBoot Maven多模块整合MyBatis打包jar(1368)
- 31. java枚举enum equal与==(1346)
- 32. springboot打成jar后获取classpath下文件失败(1245)
- 33. springboot dubbo filter之依赖注入null(1239)
- 34. 空间索引详解(1105)
- 35. dubbo filter实现接口认证springboot idea(1096)
- 36. springboot jpa mongodb 整合mysql Field in required a bean of type that could not be found Failed to load ApplicationContext(1089)
- 37. 携程Apollo统一配置中心的搭建和使用(1022)
- 38. HashMap resize导致死循环(958)
- 39. 非maven项目 idea project structure(899)
- 40. Java使用icepdf转高清图(895)

评论排行榜

- 1. SpringBoot实现JWT保护前后端分离RESTful API(5)
- 2. SpringBoot整合MyBatis之xml配置(4)
- 3. java volatile详解(3)
- 4. SpringBoot浏览器直接访问html(2)
- 5. UML之类图详解(2)
- 6. SpringBoot Maven多模块整合MyBatis 打包jar(2)
- 7. 23种设计模式详解(1)
- 8. 单列索引和联合索引区别(1)
- 9. AOP拦截日志报错IllegalStateException: It is illegal to call this method if the current request is not in asynchronous mode(1)
- 10. UML之时序图详解(1)

推荐排行榜

- 一、安装环境
- 二、安装步骤

历史上一天:

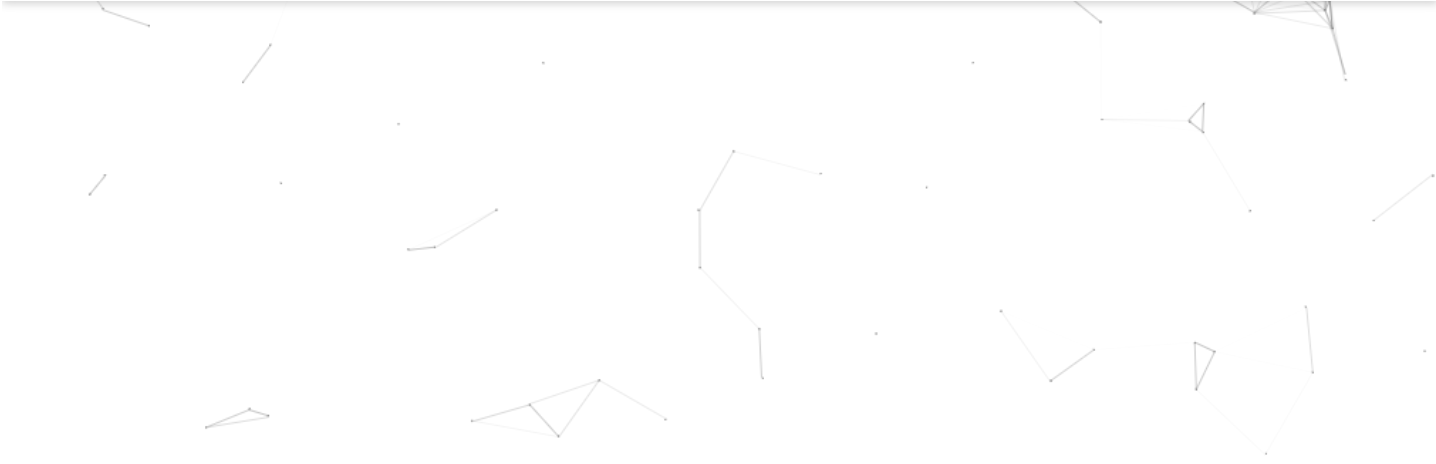
2020-04-15 INSERT IGNORE 与INSERT INTO的区别

2020-07-04-15 linux安装redis 完整步骤

Fork me on GitHub

杭州
23°

Copyright © 2020 枫树湾河桥
Powered by .NET Core on Kubernetes



- 一、安装环境
- 二、安装步骤