

深入浅出编译原理-7-编译器前端实验

原创

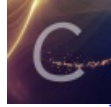
Rill 于 2012-10-04 18:35:15 发布 4596 收藏 1

分类专栏: [编译原理](#) 文章标签: [编译器](#) [java](#) [jdk](#) [path](#) [shell](#) [makefile](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/rill_zhen/article/details/8039922

版权



[编译原理](#) 专栏收录该内容

7 篇文章 2 订阅

订阅专栏

引言

还是那句话, 编译原理比较乏味, 所以有一个感性的, 直观的认识, 很重要。上一节没有做这个事情, 这一小节, 就以《编译原理》(龙书)里的那个前端的源码为例, 感受一下, 一个真正的, 完整的, 我们的手可以摸得到的, 编译器前端的工作情况和结果。

分三部分内容:

- 1》linux下安装JDK, 用来编译front。
- 2》编译front, front的代码, 我已经传上来了, 在上一篇博客里有具体的下载地址。
- 3》看结果, 小试牛刀。

7.1安装JDK

参考:

<http://blog.csdn.net/rital/article/details/5486021>

内容如下。

注意: 原文中有问题, 版本前后不一, 本文中已做更正。

Linux下JDK的安装步骤:

第一步: 改权限

```
[root@wxr java]# pwd
```

```
/usr/java
```

```
[root@wxr java]# chmod 755 jdk-6u19-linux-i586.bin
```

首先更改操作权限, 加入写入和执行的权限。

第二步：安装

执行这个jdk-6u19-linux-i586.bin二进制可执行程序，实现安装：

```
[root@wxr java]# ./jdk-6u19-linux-i586.bin
```

安装过程提示如下：

Do you agree to the above license terms? [yes or no] 按yes键确定。

之后会出现下面的画面：

```
Checksumming...
```

```
0
```

```
0
```

```
Extracting...
```

结束时提示画面：

For more information on what data Registration collects and

how it is managed and used, see:

<http://java.sun.com/javase/registration/JDKRegistrationPrivacy.html>

Press Enter to continue.....

Done.

装完就会看到/usr/java下多了一个目录：jdk1.6.0_19

第三步：设置环境变量

总体来说就是让运行的JAVA命令能找到JAVA的目录路径,知道它安装到哪了。

通常都喜欢用export命令直接在shell下设置：

```
[root@wxr java]# export JAVA_HOME=/usr/java/jdk1.6.0_19
```

```
[root@wxr java]# export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
```

```
[root@wxr java]# export PATH=$PATH:$JAVA_HOME/bin
```

的变量。所以最好的方法还是修改.bashrc文件。（/root目录中的.bashrc是个隐含文件。需要ls -a来看）

当然这样设置环境变量是可以生效的，但是只对当前shell生效。如果从另外一个shell登陆，将不能使用刚才设置

以root用户身份修改.bashrc文件，文件末尾增加蓝色字体内容：

```
[root@wxr ~]# vi .bashrc
```

```
# User specific aliases and functions
```

```
alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

export JAVA_HOME=/usr/java/jdk1.6.0_19
export CLASSPATH=$CLASSPATH:$JAVA_HOME/lib:$JAVA_HOME/jre/lib
export PATH=$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$PATH:$HOMR/bin
```

当然也可以通过更改/etc/profile来实现，不过不推荐这么做，因为这样的设置将对所有用户的shell都生效，对系统安全会产生影响。方法如下：

```
[root@localhost etc]# more /etc/profile
```

在文件最后加上以下内容即可：

```
unset i
unset pathmunge
export JAVA_HOME=/usr/java/jdk1.6.0_19
export PATH=$JAVA_HOME/bin:$PATH
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
```

下面来验证一下变量设置是否生效，设置完环境变量最好重起一下，不然看不到效果。

```
[root@wxr ~]# reboot
[root@wxr ~]# echo $JAVA_HOME
/usr/java/jdk1.6.0_19
```

这样，说明已经设置好啦~

第四步：测试

```
[root@wxr test_temp]# pwd
/usr/test_temp
```

```
[root@wxr test_temp]# vi test.java
class test
{
    public static void main(String[] args)
    {
        System.out.println("welcome to my JAVA environment!");
    }
}
```

保存退出，下面来编译、执行；

```
[root@wxr test_temp]# javac test.java
```

```
[root@wxr test_temp]# java test
```

```
welcome to my JAVA environment!
```

至此，全部JDK安装成功！

7.2 编译

1》 cd front

2》 make

7.3 牛刀小试

1》 把你想要测试的代码放到tests目录下，比如rill.t。以书上P628页的例子为例，如下：

□

2》 执行 `java main.Main <tests/rill.t >tmp/rill.i`

说明：上面命令中'<'，是将rill.t的内容作为输入，给main.Main

上面命令中'>'，是将执行结果输出到rill.i

总之，一句话，重定向，或者叫管道。

3》 `cat rill.i` 查看处理结果

结果如下：

□

4》 如果想同时测试多个文件就执行以下`make test`，这个看一下Makefile就会明白的。

7.4 小结

可见和书上给出的结果一样。下一步就是试着改一下测试代码，或者改一下这个前端，看一下有什么不同。我相信，编译原理这个东西，这样学起来，或许会更有趣一些。

试一下吧。