前言

今天这篇文章，讲解linux系统的全局环境变量以及当前用户环境变量的配置。

以前在没有遇到实际问题时，基本忽略掉思考，所有配置的环境变量都配成全局生效。这样是有弊端的，在某一次偶然的机会，我沉思的一小会，意识到全局环境变量与当前用户环境变量还是区分为好，不应一概而论。

直到后来我在linux服务器从Oracle11g过度到Oracle19c时。需要创建普通用户例如oracle才能安装，这时我才恍然大悟。这大概是linux操作系统的魅力所在。

当时遇到的国产银河麒麟操作系统，把不同用户权限区分的明明白白的，三权分立。管理员用户root、安全用户secure、普通用户user，三者之间均无法访问别人的home目录，这时我才明白原来那个root用户是假的。泾渭分明，这样做当然有优势。好处在于运维人员只需要分心维护管路员用户和安全用户。普通用户权限比较小，想对来说不会造成较为严重的后果。

正文

配置环境变量，并不一定需要非得配置成所有用全局的或者当前用户全局的。如果使用的中间件有配置文件，你甚至可以在当前的中间件配置文件中指定JDK安装的路径，我做测试时一般会这样配置，为了不干扰原有的环境配置而做出让步。这样会显得繁琐一些，配置全局则更为方便管理。

一、配置全局环境变量

1、Redhat系列配置环境变量

注意：**需要root用户权限**，才能编辑 `/etc/profile` 文件，默认配置如下图所示

```
# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

# It's NOT a good idea to change this file unless you know what you
# are doing. It's much better to create a custom.sh shell script in
# /etc/profile.d/ to make custom changes to your environment, as this
# will prevent the need for merging in future updates.

pathmunge () {
    case ":${PATH}:" in
        *:"$1":*)
            ;;
        *)
            if [ "$2" = "after" ] ; then
                PATH=$PATH:$1
            else
                PATH=$1:$PATH
            fi
    esac
}


if [ -x /usr/bin/id ]; then
```

1,1 顶端

1.1、永久生效

1.1.1、配置全局环境变量，针对所有用户

首先需要切换到root用户模式，然后编辑全局系统环境变量配置文件，所有用户都生效，永久生效。

```
#切换到root用户下
su root
#编辑全局系统环境变量存储配置文件，所有用户都生效，永久生效。
vim /etc/profile
```

例如加入配置文件，对应你安装JDK所在路径

```
#例如加入JDK环境变量
export JAVA_HOME=/usr/local/jdk1.8
```

1.2.1、source命令

执行source命令读取并执行shell脚本配置文件，无需重启服务器。

```
#执行source命令使其生效
source /etc/profile
```

1.2、临时生效

临时设置这种方式一般很少采取，大多数情况下都是直接编辑环境变量配置文件进行设置。

```
#例如配置ORACLE_HOME
export ORACLE_HOME=$ORACLE_BASE/product/19c/dbhome_1;
```

2、Ubuntu配置环境变量

2.1、永久生效

2.1.1、全局环境变量，针对所有用户

首先需要切换到root用户模式，然后编辑全局系统环境变量配置文件，所有用户都生效，永久生效。

```
#切换到root用户下
su root
#编辑全局系统环境变量存储配置文件，所有用户都生效，永久生效。
vim /etc/profile
```

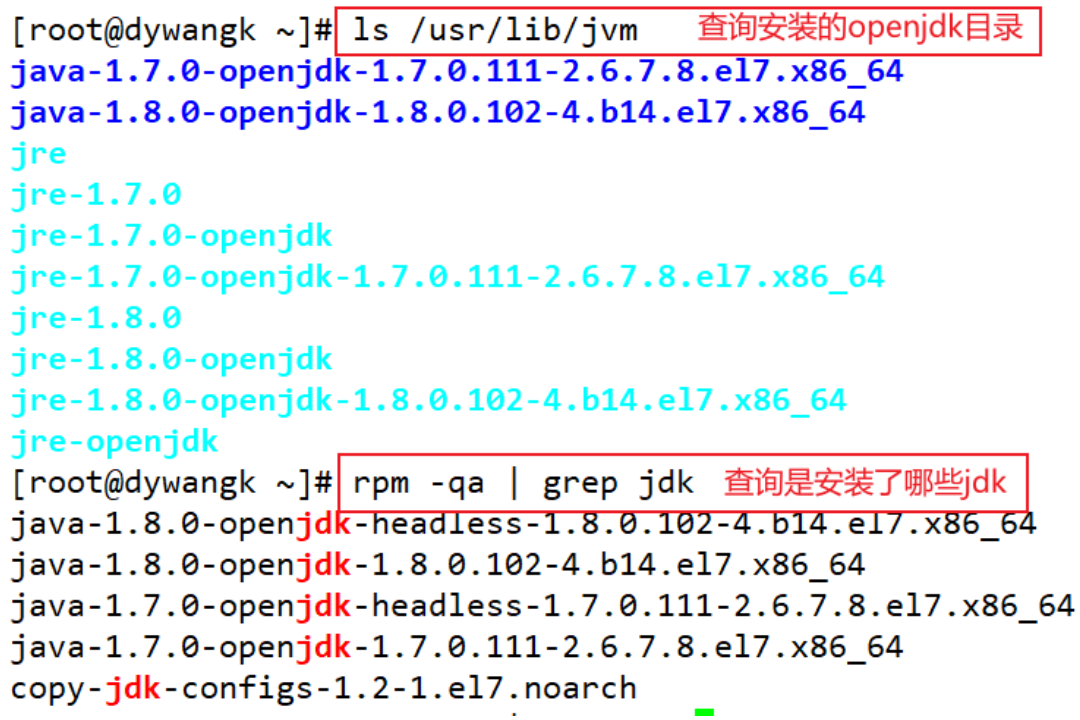
2.1.2、当前用户环境变量

```
#编辑当前用户环境变量
vim .bashrc
```

加入环境变量配置，这是以源码包安装`JDK`为例子

```
#以最常见的JDK为例子,真实安装的JDK目录，一般源码包安装会这样设置
export JAVA_HOME=/usr/local/java
#OPENJDK，一般情况下openjdk所在目录
export JAVA_HOME=/usr/lib/jvm
```

既然谈到了openjdk，我就简单的列举下**Centos7.3默认安装openjdk的目录**，如下图所示



如果你是开发人员，你可能还需要加入如下配置CLASS_PATH的配置，在lib目录下的dt.jar和tools.jar。在JDK5以后，貌似可以不用加此项，但是加了比较保险。

```
#CLASS_PATH的配置，在lib目录下的dt.jar和tools.jar
export
CLASS_PATH=.:$JAVA_HOME/jre/lib/rt.jar:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tool
s.jar
```

最后配置PATH，找到bin目录

```
#PATH的设置，找到bin目录
export PATH=$PATH:$JAVA_HOME/bin
```

二、配置当前用户环境变量

1、切换到当前用户

1.1、Redhat系列

```
#切换至当前用户的home目录
cd ~
#编辑当前用户的系统环境变量
vim .bash_profile
```

1.2、Ubuntu系列

```
#切换至当前用户的home目录
cd ~
#编辑当前用户的系统环境变量
vim .bashrc
```

2、Oracle为例配置环境变量

下面列举的例子是新增了oracle用户，并且安装了oracle19c，然后配置oracle用户的环境变量。

2.1、永久生效

编辑当前用户的环境变量配置文件`.bash_profile`，并加入如下配置，执行`source .bash_profile`生效。

```
#加入环境变量
ORACLE_BASE=/opt/oracle;
ORACLE_HOME=$ORACLE_BASE/product/19c/dbhome_1;
PATH=$ORACLE_HOME/bin:$PATH;
ORACLE_SID=ORCLCDB;

export ORACLE_SID ORACLE_BASE ORACLE_HOME PATH
```

2.2、临时生效

此方法在当前用户下配置环境变量只是临时生效，如果服务器不重启那可能一直有效。万一重启了，就失效了，相当于一次性的设置。建议在当前用户或者全局配置文件中写入相应的配置，效果更佳。

```
#例如配置ORACLE_HOME
export ORACLE_BASE=/opt/oracle;
export ORACLE_HOME=$ORACLE_BASE/product/19c/dbhome_1;
```