**描　　述**

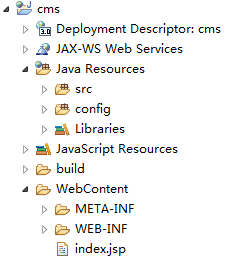
要开发一个”cms系统“，有2个人分别是：晓飞， 盈月。要求使用Git来进行版本控制。

**项目信息**

版本控制：Git

项目名称：cms

开发人员：xiaofei，yingyue



**概　　要**

1、服务器上安装Git依赖及Git

2、创建Git用户及所属组

3、安装Git客户端并生成公钥

4、创建证书登录

5、服务器上初始化Git仓库

6、使用Git Bash克隆服务器上的空仓库

7、将本地库项目推送到服务器

8、导入本地Git库的项目到eclipse

## ****1、服务器上安装Git依赖及Git：****

安装Git依赖

注：yum install curl-devel expat-devel gettext-devel openssl-devel zlib-devel perl-devel是一次性安装也可以一个一个安装

[root@localhost zhoujianxu]# yum install curl-devel expat-devel gettext-devel openssl-devel zlib-devel perl-devel

安装Git

　如果不知道你要安装包的具体名称，可以使用yum provides \*/name 进行查找系统自带软件包的信息（这一步不是必须）；

   [root@localhost zhoujianxu]# yum provides \*/git

安装git 执行命令 yum install -y git （-y表示安装过程中不需要手动确认可以不加那么安装过程中需要输入y手动回车确认）

  [root@localhost zhoujianxu]# yum install -y git

完成；

### **2、创建”用户组“和”用户“，用来运行git服务。**

　创建用户组

   [root@localhost zhoujianxu]# groupadd java\_team

  创建用户并yingyue添加到名为”java\_team“的组中

   [root@localhost zhoujianxu]# adduser yingyue -g java\_team

   为用户名为yingyue的用户设置密码

　[root@localhost zhoujianxu]# passwd yingyue  
   Changing password for user git.  
   New password:   
   Retype new password:   
   passwd: all authentication tokens updated successfully.  
   [root@localhost zhoujianxu]#

   创建第二个用户xiaofei

   [root@localhost zhoujianxu]# groupadd java\_team

  将xiaofei”java\_team“的组中

   [root@localhost zhoujianxu]# adduser xiaofei -g java\_team

   [root@localhost zhoujianxu]# passwd xiaofei

   Changing password for user xiaofei.  
   New password:   
   Retype new password:   
   passwd: all authentication tokens updated successfully.  
   [root@localhost zhoujianxu]#

### **3、安装Git客户端并生成公钥：**

安装Git客户端（开发人员安装）

下载地址https://git-scm.com/downloads 目前最新版本为：Git-2.10.2-64-bit.exe

IMG_256

生成公钥

方式一：图形界面Git GUI方式

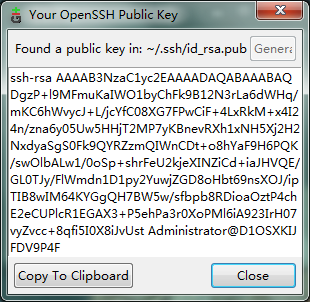
右键Git GUI here



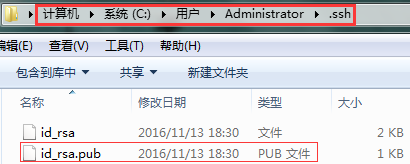
点击Help--》Generate Key生成SSHkey



如下

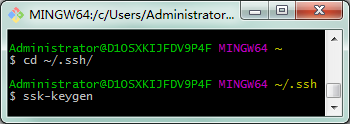


这时.ssh目录下就有了公钥(id\_rsa.pub)和私钥(id\_rsa)



方式二：使用Git Bash通过ssh-keygen命令生成， ssh-keygen -t rsa   参数-t rsa 意思是以rsa的加密方式生成秘钥对, 不是必须。

打开Git Bash进入.ssh目录如果没有可以手动建



### **4、创建证书登录**

4.1、在用户yingyue的.ssh目录下创建authorized\_keys文件

收集所有需要登录的用户的公钥，公钥位于id\_rsa.pub文件中，把我们的公钥导入到/home/yingyue/.ssh/authorized\_keys文件里，一行一个。

如果没有该文件创建它（.ssh为隐藏文件夹可以使用ls -la 或 ll 命令查看）：

[root@localhost zhoujianxu]# cd /home/yingyue/

[root@localhost yingyue]# ls

[root@localhost yingyue]# mkdir .ssh

[root@localhost yingyue]# ls

[root@localhost yingyue]# ls -la

total 32

drwx------. 5 yingyue java\_team 4096 Nov 12 22:48 .

drwxr-xr-x. 4 root root 4096 Nov 12 22:12 ..

-rw-r--r--. 1 yingyue java\_team 18 Feb 21 2013 .bash\_logout

-rw-r--r--. 1 yingyue java\_team 176 Feb 21 2013 .bash\_profile

-rw-r--r--. 1 yingyue java\_team 124 Feb 21 2013 .bashrc

drwxr-xr-x. 2 yingyue java\_team 4096 Nov 11 2010 .gnome2

drwxr-xr-x. 4 yingyue java\_team 4096 Nov 12 19:25 .mozilla

drwxr-xr-x. 2 root root 4096 Nov 12 22:48 .ssh

[root@localhost yingyue]# chmod 700 .ssh

[root@localhost yingyue]# touch .ssh/authorized\_keys

[root@localhost yingyue]# chmod 600 .ssh/authorized\_keys  
注：chmod命令详解

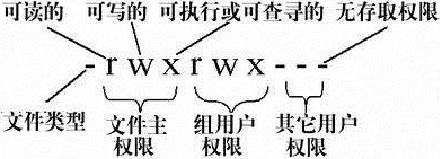
在linux系统中，我们经常会通过chmod命令来更改文档所属的权限，格式为：chmod abc file

其中a,b,c各为一个数字，分别表示User、Group、及Other的权限。

在解释上述命令之前，我们先来了解一下unix系统家族中，对用户权限的划分在yingyue账户下新建一个文件，test.txt 然后我们用ls -a 来查看

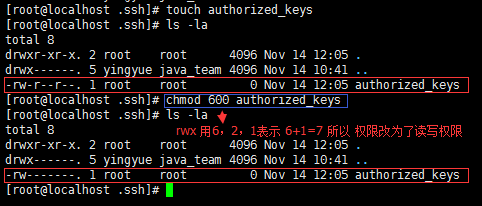
[root@localhost yingyue]# ls -la test.txt   
drwxr-xr-x.2 root root 0 Nov 14 10:16 test.txt   
可以看到，之前有-rwxr-xr-x这样一串字符。第一位为文件类型，之后每三位在一起分别代表用户的权限（User）、组的权限（group）、其他用户的权限（Other）

其结构如下图所示：



r=4，w=2，x=1   
若要rwx属性则4+2+1=7   
若要rw-属性则4+2=6  
若要r-x属性则4+1=5  
如下图：

chmod 600 .ssh/authorized\_keys 表示只有所属用户对.ssh目录下的authorized\_keys具有读写权限没有执行权限，Group和Other没有权限。



-rw------- (600) -- 只有属主有读写权限。   
-rw-r--r-- (644) -- 只有属主有读写权限；而属组用户和其他用户只有读权限。   
-rwx------ (700) -- 只有属主有读、写、执行权限。   
-rwxr-xr-x (755) -- 属主有读、写、执行权限；而属组用户和其他用户只有读、执行权限。   
-rwx--x--x (711) -- 属主有读、写、执行权限；而属组用户和其他用户只有执行权限。   
-rw-rw-rw- (666) -- 所有用户都有文件读、写权限。这种做法不可取。   
-rwxrwxrwx (777) -- 所有用户都有读、写、执行权限。更不可取的做法。   
以下是对目录的两个普通设定:

drwx------ (700) - 只有属主可在目录中读、写。   
drwxr-xr-x (755) - 所有用户可读该目录，但只有属主才能改变目录中的内容。

Linux/Unix 的档案调用权限分为三级 : 档案拥有者、群组、其他。利用 chmod 可以藉以控制档案如何被他人所调用。

4.2、将id\_rsa\_yingyue.pub 公钥添加到authorized\_keys文件中

收集所有需要登录的用户（即：开发人员 xiaofei，yingyue）的公钥，把我们的公钥导入到/home/yingyue/.ssh/authorized\_keys文件里，一行一个。（注：公钥可以发给管理员管理员通过Xshell或文件传输工具传到服务器通过 vi 或 cat 命令进行合并即可，cat控制台查看  >> 或 tee 表示输入到目标文件中）如下：

将yingyue的公钥添加到authorized\_key中：

[root@localhost yingyue]# cat /home/pubkeys/id\_rsa\_yingyue.pub >> .ssh/authorized\_keys

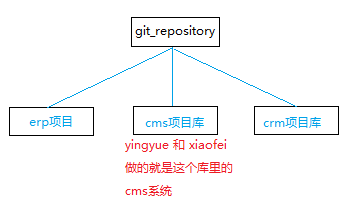
将yingyue的公钥添加到authorized\_key中：

[root@localhost xiaofen]# cat /home/pubkeys/id\_rsa\_xiaofei.pub | tee .ssh/authorized\_keys

### **5、服务器上初始化Git仓库（即：在服务器上创建一个空的仓库）**

首先选定一个目录作为Git仓库假设是git\_respository那么我的想法是这样的这这个文件夹下可以创建多个git仓库通过不同的命名来区分仓库作用。

结构图如下：



从上图可知cms项目在cms\_repo.git这个库，路径为/home/git\_repository/cms\_repo.git，下面就在git\_repository目录下初始化一个空的仓库cms\_repo.git

[IMG_259](http://www.cnblogs.com/YingYue/p/javascript:void(0);)

[root@localhost yingyue]# cd /home

[root@localhost home]# mkdir git\_repository

[root@localhost home]# ls

git\_repository

[root@localhost home]# chown yingyue:java\_team git\_repository

[root@localhost home]# cd git\_repository/

[root@localhost git\_repository]# git init --bare cms\_repo.git

Initialized empty Git repository in /home/git\_repository/cms\_repo.git/

[root@localhost git\_repository]# lscms\_repo.git

[root@localhost git\_repository]#

[IMG_260](http://www.cnblogs.com/YingYue/p/javascript:void(0);)

chow 命令详解：

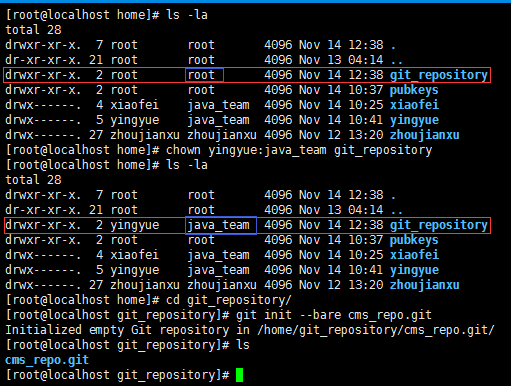
chown将指定文件的拥有者改为指定的用户或组（即：修改文件目录属主），用户可以是用户名或者用户ID；组可以是组名或者组ID；文件是以空格分开的要改变权限的文件列表，支持通配符。系统管理员经常使用chown命令，在将文件拷贝到另一个用户的名录下之后，让用户拥有使用该文件的权限。

　　1．命令格式：chown [选项]... [所有者][:[组]] 文件...

　　2．命令功能：  
　　　　通过chown改变文件的拥有者和群组。在更改文件的所有者或所属群组时，可以使用用户名称和用户识别码设置。普通用户不能将自己的文件改变成其他的拥有者。其操作权限一般为管理员。

如下图：

chown yingyue:java\_team git\_repository 表示将git\_repository目录的权限更改为当前所属用户为yingyue所属组为java\_team，如下图：

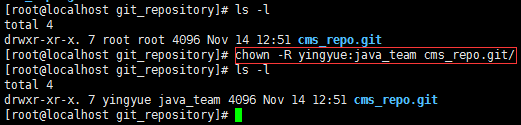


chown命令的参数-R表示向下递归改变目录的所属用户和所属组，chown -R yingyue:java\_team cms\_repo.git 表示将cms\_repo.git目录的权限向下递归更改为所属用户为yingyue所属组为java\_team。

[root@localhost git\_repository]# chown -R yingyue:java\_team cms\_repo.git

[root@localhost git\_repository]#

如下图：



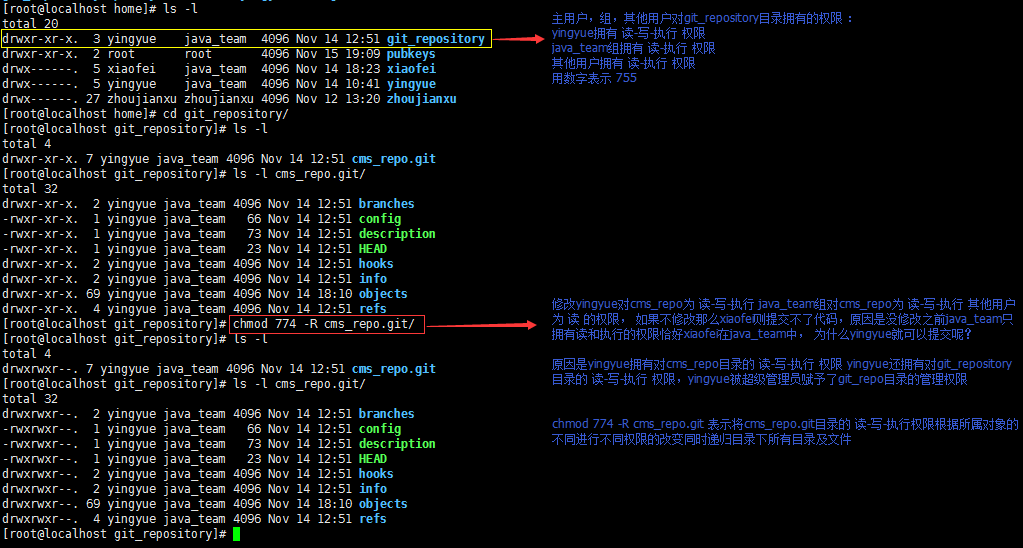
**这一步很重要**

给yingyue对git\_repository目录 可读，可写，可执行 的权限；对cms\_repo.git目录及子目录 可读，可写，可执行的权限。（rwx用数字7表示）

给xiaofei对cms\_repo.git目录及子目录 可读，可写，可执行的权限，xiaofei在java\_team里所以该组的权限即可。              ( rwx用数字7表示 )

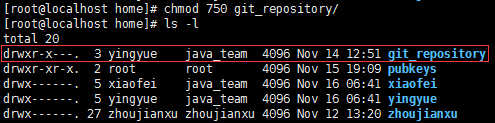
给其他用户对cms\_repo.git目录及子目录 可读 的权限。　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　( r - -用数字4表示 )

chmod 774 -R cms\_repo.git

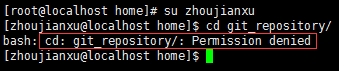


 落下一步，将git\_repository目录权限改为 yingyue 拥有 可读，可写，可执行 权限；java\_team组 拥有 可读，可执行 权限；其他用户没有权限；

chmod 750 git\_repository



这时用户名为zhoujianxu的用户对git\_repository没有任何权限了

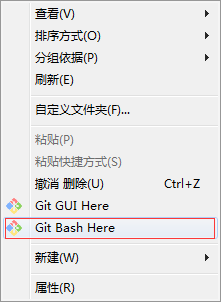


### **6、使用Git Bash克隆服务器上的空仓库**

仓库地址：git@192.168.42.129:/home/git\_repository/cms\_repo.git

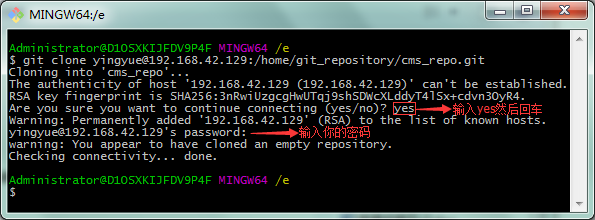
 现在Git服务器上已经有了空仓库cms\_repo.git 那么yingyue想把eclipse中的cms项目传到仓库中，步骤如下：

进入本地磁盘F盘右键Git Bash Here

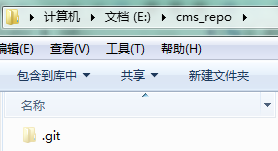


格式git clone [user@]example.com:/\*/\*.git/

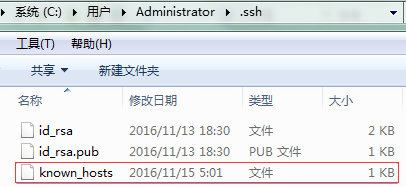
执行 git clone yingyue@192.168.42.129:/home/git\_repository/cms\_repo.git



仓库已经克隆到yingyue的本地磁盘F



同时在本地磁盘的.ssh目录下多了个文件known\_hosts



这个文件是纪录连到对方时，对方给的 host key。每次连线时都会检查目前对方给的 host key 与纪录的 host key 是否相同，可以简单验证连结是否又被诈骗等相关事宜。

注：克隆版本库的时候，所使用的远程主机自动被Git命名为origin。如果想用其他的主机名，需要用git clone命令的-o选项指定（可以使用git remote命令查看，添加，修改，删除远程主机）。

例如：  
$ git clone -o jQuery https://github.com/jquery/jquery.git

$ git remote

jQuery  
表示克隆的时候，指定远程主机叫做jQuery

   git clone支持多种协议，除了HTTP(s)以外，还支持SSH、Git、本地文件协议等，下面是一些例子。

$ git clone http[s]://example.com/\*/\*/\*.git/

$ git clone ssh://example.com/\*/\*/\*.git/

$ git clone git://example.com/\*/\*/\*.git/

$ git clone /opt/git/project.git

$ git clone file:///opt/git/project.git

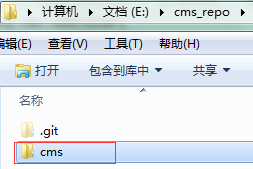
$ git clone ftp[s]://example.com/\*/\*/\*.git/

$ git clone rsync://example.com/\*/\*/\*.git/  
  
ssh协议还有另一种写法

$ git clone [user@]example.com:\*/\*/\*.git/  
  
来试一下用户zhoujianxu可以克隆吗？上面已经设置设置了对git\_repository目录的访问权限，所以zhoujianxu是没法克隆的 如下图；  


### **7、将本地库项目推送到服务器**

 将eclipse中的cms项目复制到cms\_repo目录



进入cms\_repo目录并查看远程主机名（默认为origin）

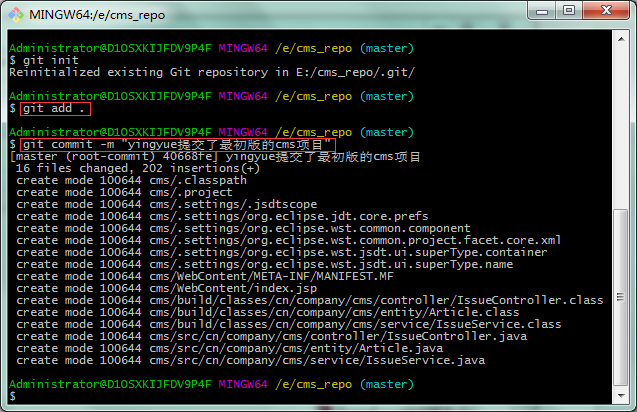


可以使用git remote rename命令用于远程主机的改名，格式为：

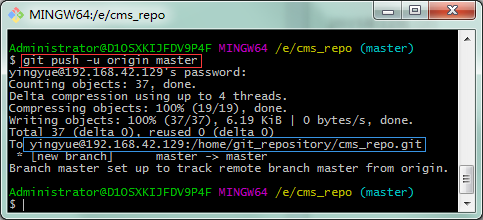
$ git remote rename <原主机名> <新主机名>

 不改了就用默认。

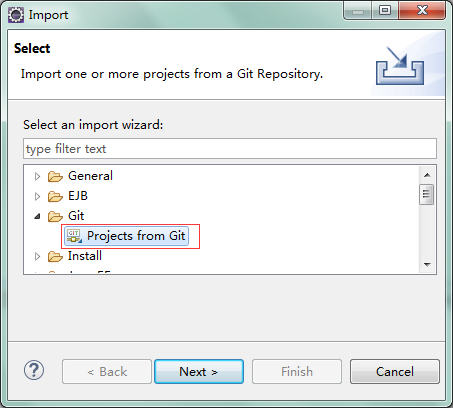
 执行git add .把整个cms项目内文件的信息添加到索引库中，使用git commit命令提交。（git将依据索引库中的内容来进行文件的提交）



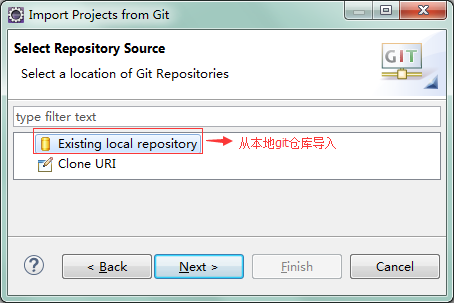
执行git push -u origin master命令将本地cms\_repository库的文件提交到远程服务器



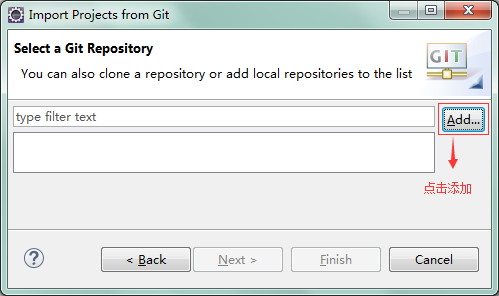
### **8、导入本地Git库的项目到eclipse**



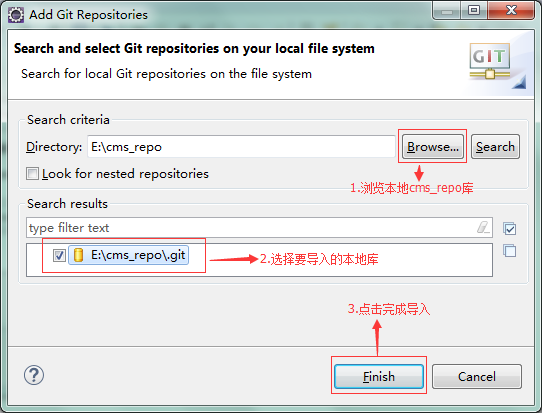
下一步



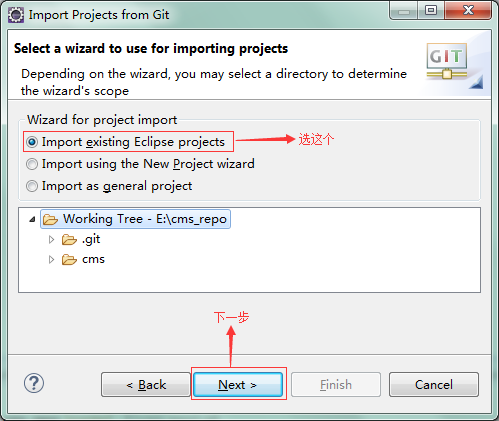
下一步



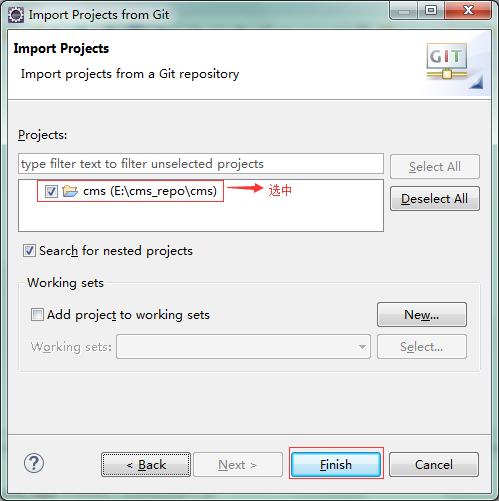
下一步



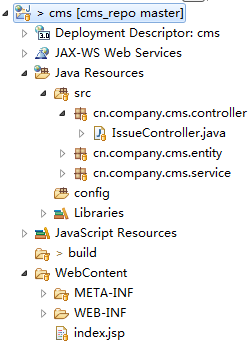
点击Finish完后选中cms\_repo点击Next



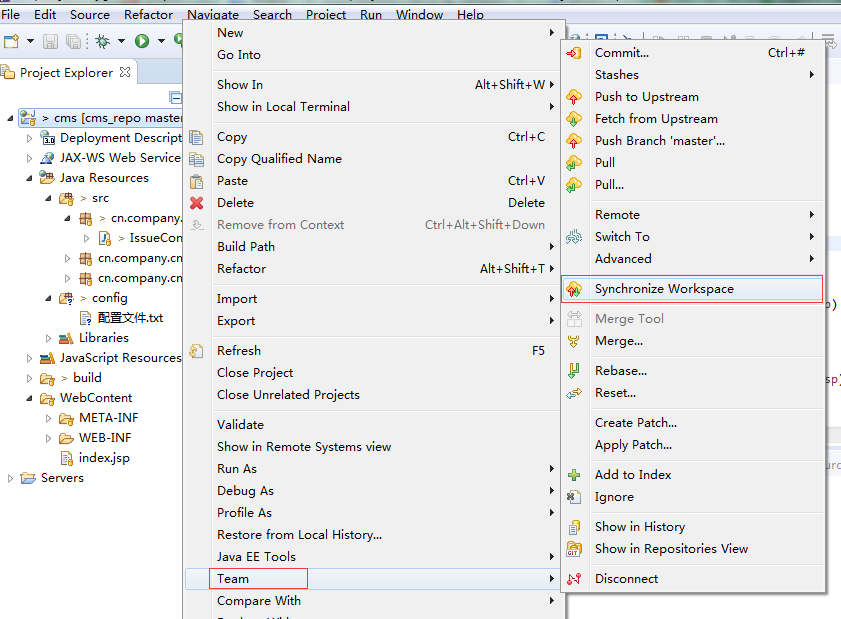
下一步



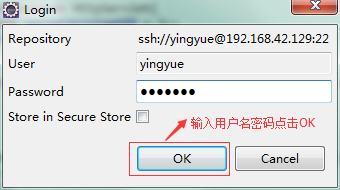
点击Finish完成，如下图项目已经成功导入到eclipse中。



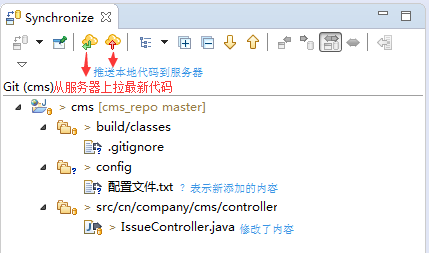
添加了新的文件需要提交，右键-->Team--Synchronize Workspace进行同步，类似svn的同步。



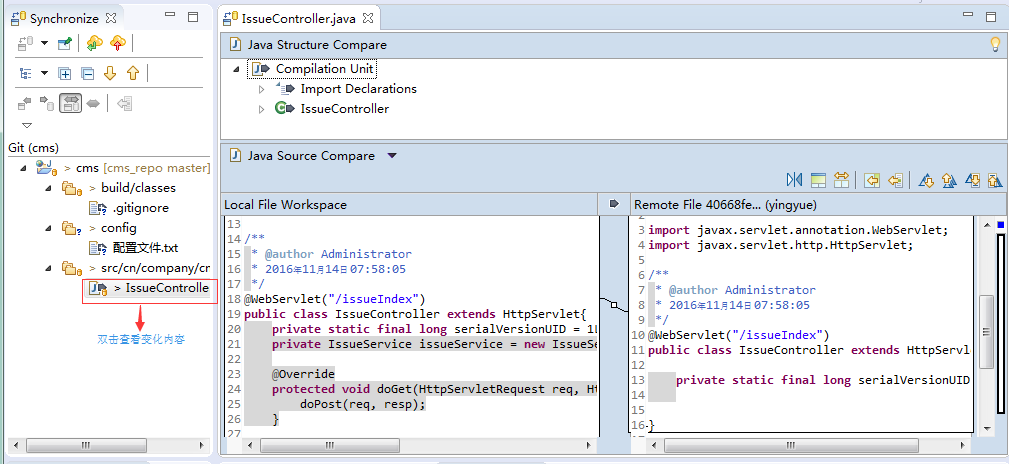
填写用户名密码



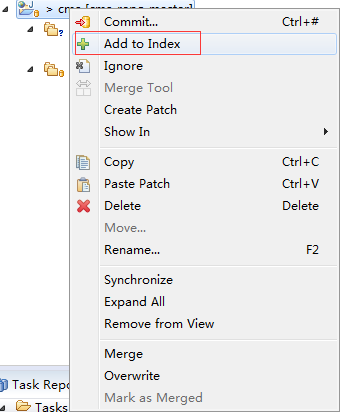
下一步



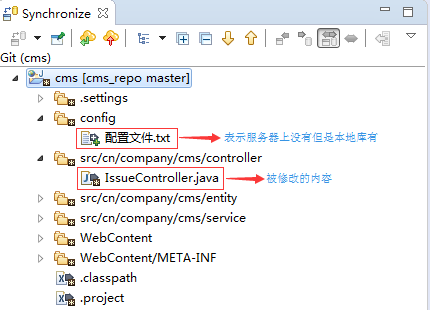
双击IssueController.java查看变化的内容



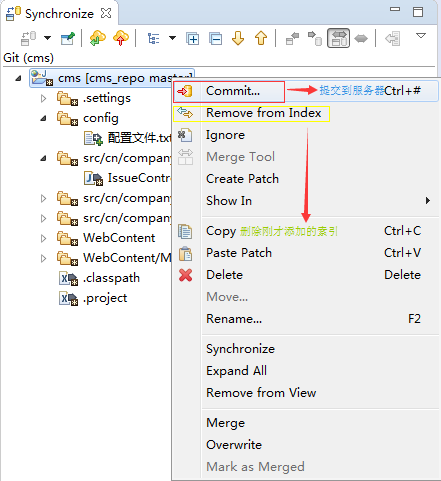
因为添加了新文件索引要为新文件添加索引选中项目右键-->Add to Index



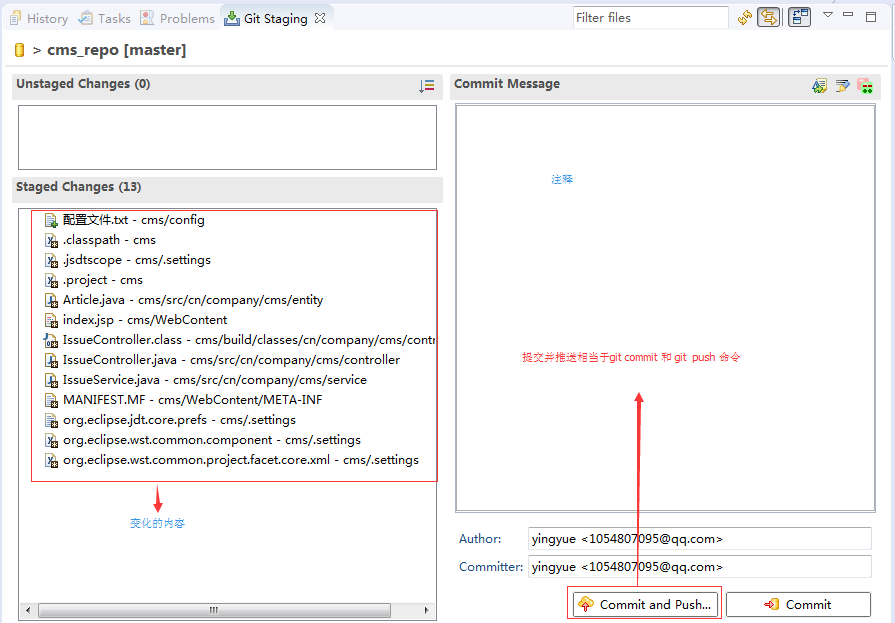
看看图标都变了



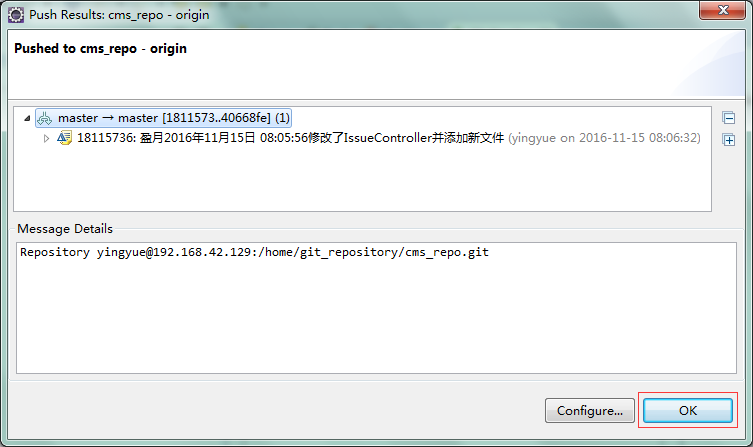
提交项目



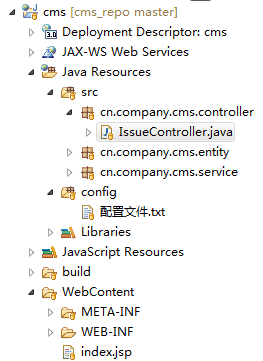
下一步Git Staging（Git工作台）



下一步



点击OK将本地Git库代码推送到服务器如下图是提交后的项目：



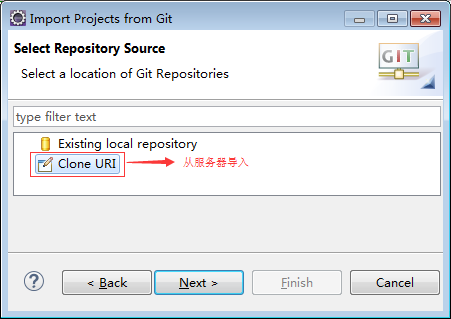
yingyue的eclipse导入cms项目完成；

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

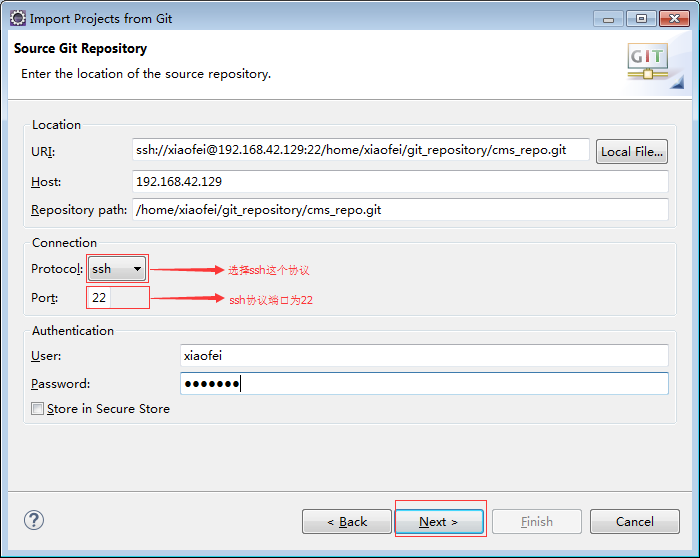
### ****下面xiaofei开始导入项目到eclipse（xiaofei使用的是Windows XP系统）****

 注意：因为xiaofei是普通的开发人员而yingyue既是开发人员又是Linux系统管理员，所以xiaofei只需要将公钥给系统管理员然后系统管理员将xiaofei的公钥放到/home/xiaofei/.ssh/authorized\_keys即可

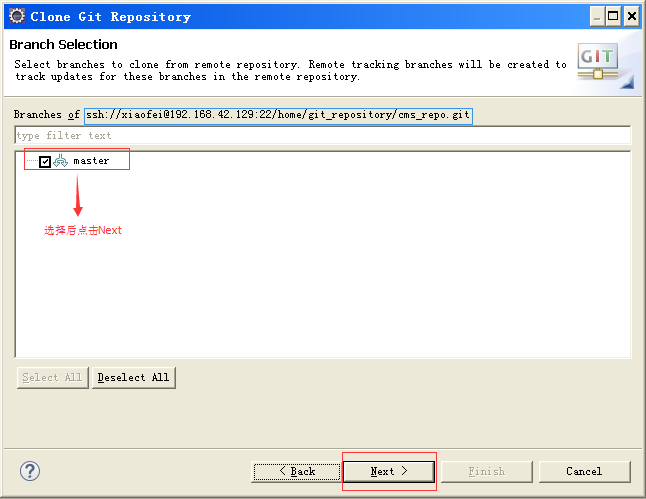
xiaofei选择Clone URI从服务器上直接导入如下图：



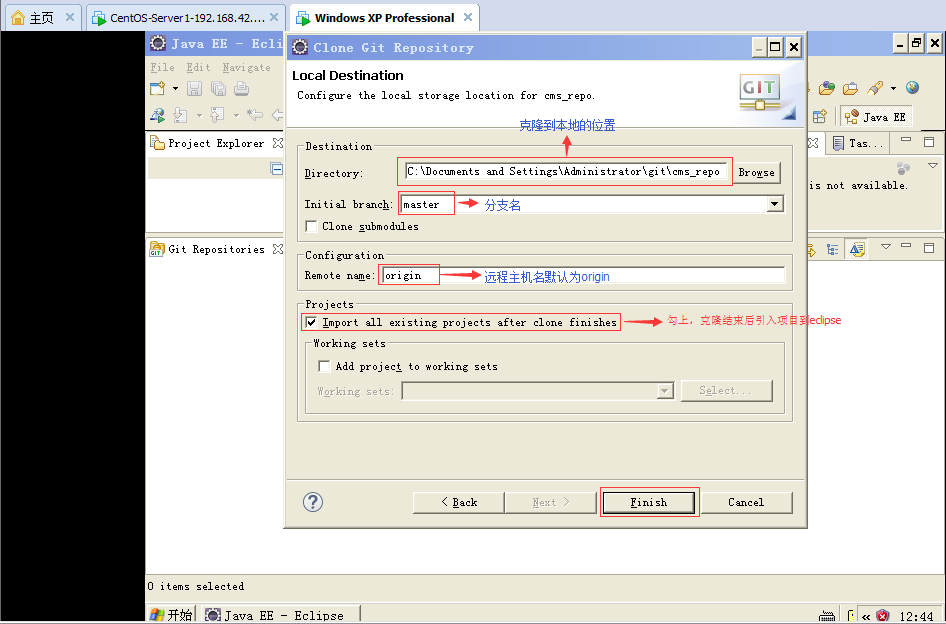
点击Next填写链接信息protocol选择ssh协议端口默认22



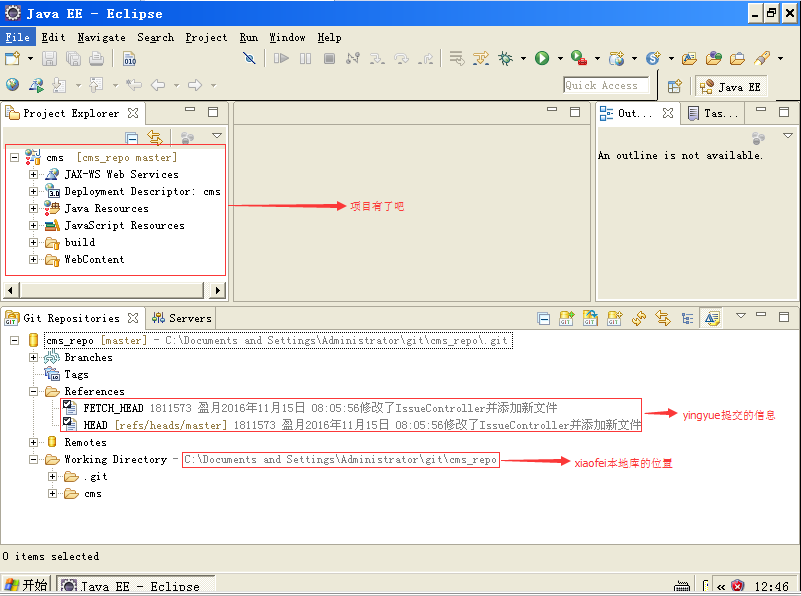
 点击Next

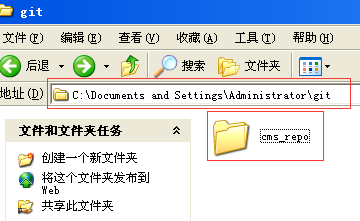


选择master分支后点击Next

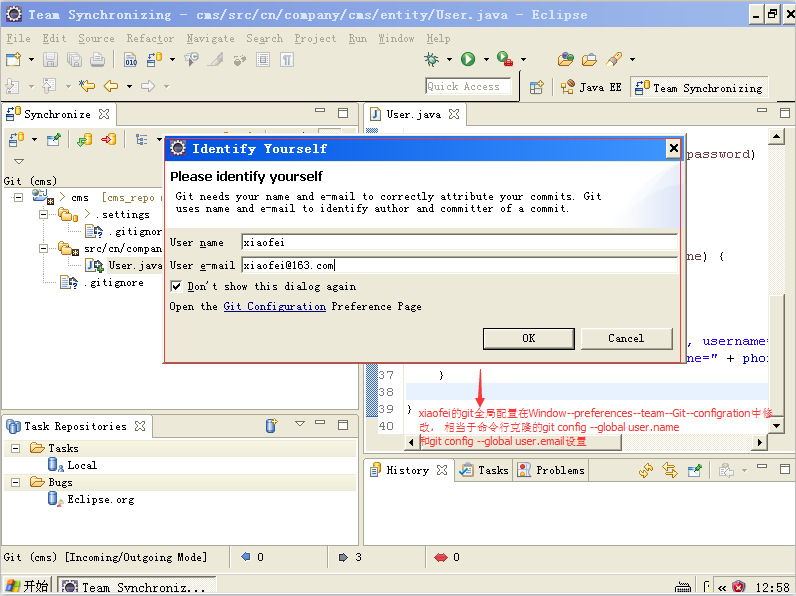


点击Finish引入项目到eclipse

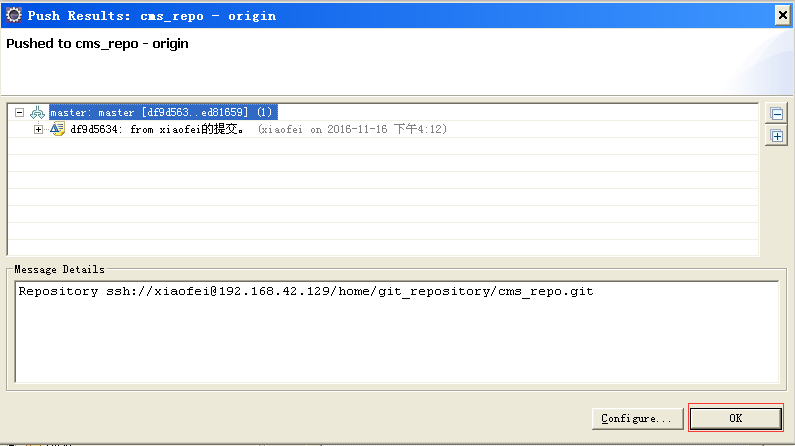




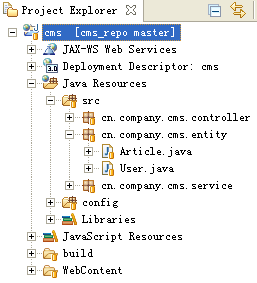
xiaofei添加了新代码并提交



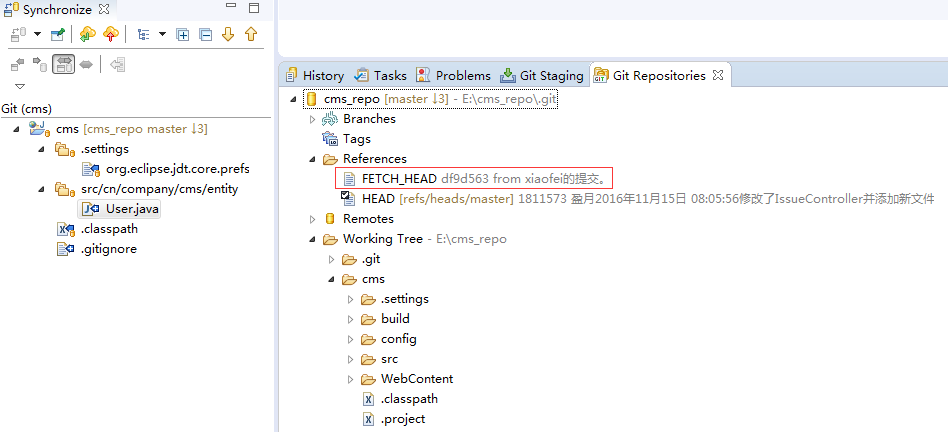
点击OK



点击OK后xiaofei完成了提交



这时yingyue就可以看到xiaofei提交的代码了



完成

————————————————————————————————————————————————————————————

### ****提示：如果报这个错****

error: insufficient permission for adding an object to repository database ./objects

说明远程服务器的目录权限还不够

终极必杀技是使用

chmod 777 -R 库地址

更改，例如：

chmod 777 -R ../cms\_repo.git/

另外还要注意每个人的Git版本要一致， eclipse的egit插件也要一致，否则会报奇奇怪怪的错。

 ————————————————————————————————————————————————————————————

**最最后一步禁用shell登录：**

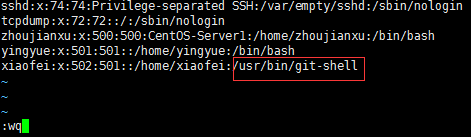
出于安全考虑，创建的git用户不允许登录shell，这可以通过编辑/etc/passwd文件完成。找到类似下面的一行：

git:x:1001:1001:,,,:/home/禁止shell登录的用户名:/bin/bash  
  
改为

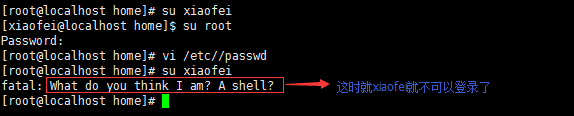
git:x:1001:1001:,,,:/home/禁止shell登录的用户名:/usr/bin/git-shell

如下, 设置xiaofei不可登录远程主机

 vi /etc/passwd



:wq保存并退出，当xiaofei在登录时 What do you think I am? A shell?



这样，git用户可以正常通过ssh使用git，但无法登录shell，因为我们为git用户指定的git-shell每次一登录就自动退出。不是更安全吗？

————————————————————————————————————————————————————————————

**2016年11月18日 12:32:25 补充**

当xiaofei修改了代码push到服务器上后，yingyue更新代码后也修改了代码这时yingyue提交，发现push失败，提示objects目录下有文件没有写入权限。

情况与下面的这段场景相似：

最近给公司部署了 git 服务器（CentOS 6.3），由于之前没有配置 git server 的经验，搞了半天才把配置搞定。这里就不讲配置的问题了。创建了账户、项目并设置 git 权限后，使用 a 账号开始测试。一切正常，没有问题。

第二天上班，给同事建了个账号 b（a,b 账号同在 git 用户组），出现问题了，pull 正常， push 失败，提示 objects 目录下有文件没有写入权限。

果断 ssh 上服务器，把账号 a 和 b 都放到 git 用户组。继续测试，还是报错。没办法，只好一路 cd 到 objects 下，ll，发现有些文件没有对同组用户开启 w 权限。想了想，用“linux 创建文件默认权限”作为关键词 Google 了一下，看了几篇文章，了解了 umask 命令。

解决办法：

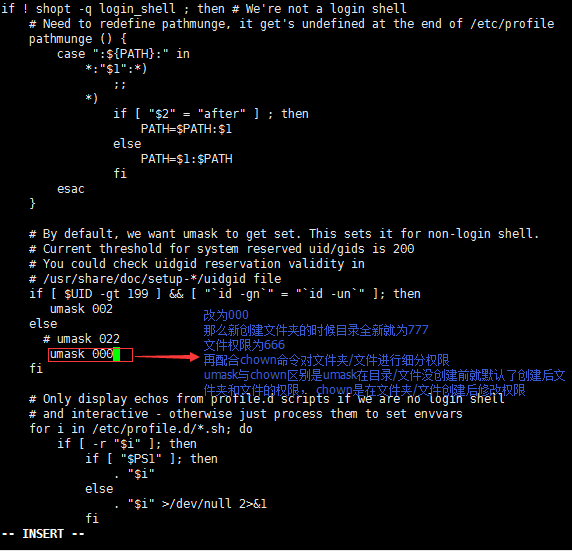
设置umask

/etc/profile和/etc/bashrc都可以用于设置用户登录系统时自动执行某些操作，他们的区别是/etc/profile只在用户第一次登录时被执行，而/etc/bashrc则在用户每次登录加载Bash Shell时都会被执行。

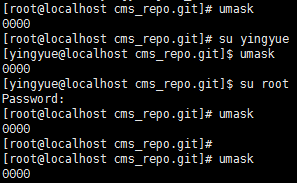
因而，如果是修改/etc/profile文件，将只对新创建的用户生效；而如果是修改/etc/bashrc文件，则对所有用户都生效。

如下：

[root@localhost cms\_repo.git]# vi /etc/bashrc



改后



 OK;