

版本管理

深 圳 市 门 道 信 息 咨 询 有 限 公 司

Shenzhen MT Information Consulting Co . , LTD

版权所有.侵权必究

目录

[1. 版本控制 4](#_Toc27839014)

[1.1 版本控制介绍 4](#_Toc27839015)

[1.2 版本控制工具 4](#_Toc27839016)

[2. SVN 4](#_Toc27839017)

[2.1 SVN服务器端安装 4](#_Toc27839019)

[2.2 配置仓库 5](#_Toc27839020)

[2.3 修改配置文件 5](#_Toc27839021)

[2.3.1 修改passwd 5](#_Toc27839027)

[2.3.2 修改svnserve.conf，打开下面3个注释 5](#_Toc27839028)

[2.4 启动和停止 5](#_Toc27839029)

[2.5 客户端访问 6](#_Toc27839030)

[2.6 测试应掌握的技巧 6](#_Toc27839031)

[2.6.1 下载 6](#_Toc27839035)

[2.6.2 上传 6](#_Toc27839036)

[2.6.3 更新 7](#_Toc27839037)

[3. Git简介 7](#_Toc27839038)

[3.1 Git介绍 7](#_Toc27839040)

[3.2 Git结构 7](#_Toc27839041)

[3.3 Git安装 8](#_Toc27839042)

[4. Git命令行操作 9](#_Toc27839043)

[4.1 初始化 9](#_Toc27839045)

[4.2 设置签名 9](#_Toc27839046)

[4.3 查看状态 10](#_Toc27839047)

[4.4 添加并提交（是2个动作） 10](#_Toc27839048)

[5. github 11](#_Toc27839049)

[5.1 注册 11](#_Toc27839051)

[5.2 新建远程库 11](#_Toc27839052)

[…or create a new repository on the command line 11](#_Toc27839053)

[…or push an existing repository from the command line 11](#_Toc27839054)

[5.3 给远程库取别名 11](#_Toc27839055)

[5.4 把本地库推送给远程库 12](#_Toc27839056)

[5.5 克隆远程库 12](#_Toc27839057)

[5.6 邀请其他人进入团队协同开发 12](#_Toc27839058)

[6. Git图形界面 12](#_Toc27839059)

# 版本控制

* 版本控制是指对软件开发过程中各种程序代码、配置文件及说明文档等文件变更的管理，是软件配置管理的核心思想之一。
* 任职角色：配置管理员

## 版本控制介绍

* 协同修改

多人并行不悖的修改服务器端的同一个文件。

* 数据备份

不仅保存目录和文件的当前状态， 还能够保存每一个提交过的历史状态。

* 版本管理

在保存每一个版本的文件信息的时候要做到不保存重复数据， 以节约存储  
间， 提高运行效率。 这方面 SVN 采用的是增量式管理的方式， 而 Git 采取了  
件系统快照的方式。

* 权限控制

对团队中参与开发的人员进行权限控制。

对团队外开发者贡献的代码进行审核——Git 独有。

* 历史记录

查看修改人、 修改时间、 修改内容、 日志信息。

将本地文件恢复到某一个历史状态。

* 分支管理

允许开发团队在工作过程中多条生产线同时推进任务， 进一步提高效率。

## 版本控制工具

集中式版本控制：VSS、CVS、SVN

分布式版本控制：Git、 Mercurial、 Bazaar

# SVN

SVN是subversion的缩写，是一个开放源代码的版本控制系统，通过采用分支管理系统的高效管理，简而言之就是用于多个人共同开发同一个项目，实现共享资源，实现最终集中式的管理。

特点：一个服务器，多个客户端的集群式管理方法



## SVN服务器端安装

|  |
| --- |
| yum install subversion |

## 配置仓库

我们在/home下建立一个名为svn的仓库（repository），以后所有代码都放在这个下面，创建成功后在svn下面多了几个文件夹

|  |
| --- |
| [root@localhost /]# cd /home  [root@localhost home]# mkdir svn  [root@localhost home]# svnadmin create /home/svn  [root@localhost home]# ls svn  conf db format hooks locks README.txt |

## 修改配置文件

我们这里特别关注一下conf文件夹，这个是存放配置文件的

|  |
| --- |
| [root@localhost home]# cd svn/conf  [root@localhost conf]# ls  authz passwd svnserve.conf |

* authz 是权限控制文件
* passwd 是帐号密码文件
* svnserve.conf 是SVN服务配置文件

在这里我们修改2个文件，passwd和svnserver.conf



### 修改passwd

|  |
| --- |
| [root@localhost conf]# vi passwd  [users]  admin = 123456  caichang = 123456 |

admin：用户名 123456：密码

如上建了2个用户

### 修改svnserve.conf，打开下面3个注释

|  |
| --- |
| [root@localhost conf]# vi svnserve.conf  anon-access = read #匿名用户可读  auth-access = write #授权用户可写  password-db = passwd #使用哪个文件作为账号文件 |

## 启动和停止

启动：-d表示守护进程， -r 表示在后台执行

|  |
| --- |
| [root@localhost conf]#svnserve -d -r /home/svn |

停止：

|  |
| --- |
| [root@localhost conf]#killall svnserve |

## 客户端访问

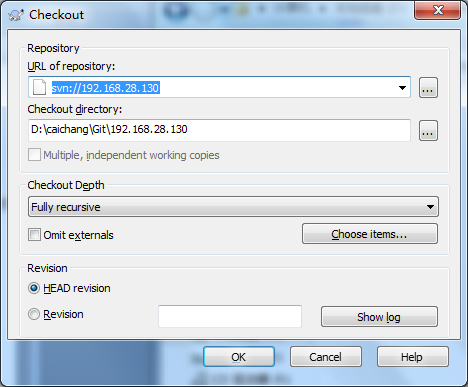
打开TortoiseSVN，输入svn://192.168.28.130，即可访问svn服务器，其中192.168.28.130为linux服务器ip

## 测试应掌握的技巧



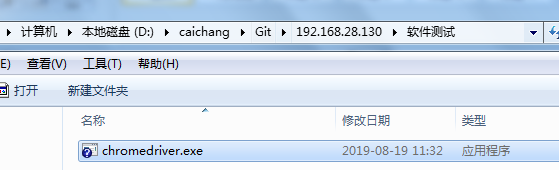
### 下载

右键》svn checkout，输入仓库地址和导出地址



### 上传

1、把要上传的文件拷贝到要上传的目录下



2、选中文件》右键》TortoiseSVN》add



3、选中文件》右键》TortoiseSVN》SVN Commit

### 更新

1、进入要更新的目录

2、右键》SVN Update

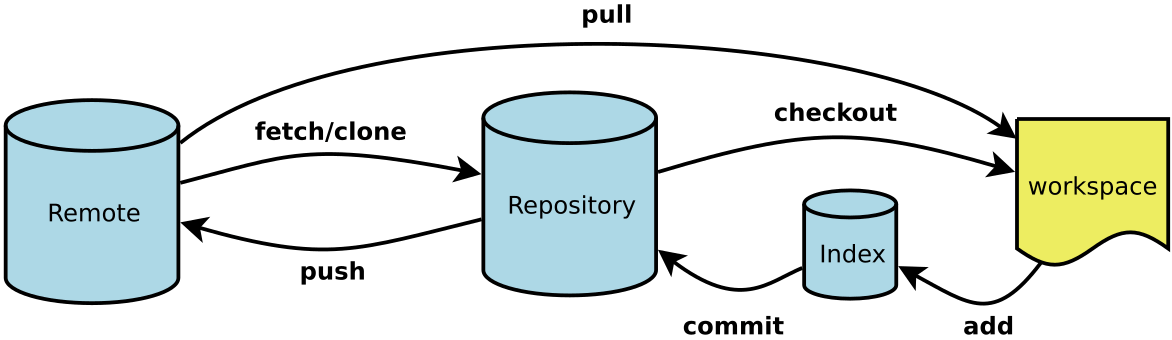
# Git简介



## Git介绍

Git是一个开源的分布式版本控制系统，分布式相比集中式的最大区别是Git没有“中央服务器”，每位开发者都可以通过克隆远程库，在本地机器上存储一个完整的Git仓库，还可以把代码的修改提交到本地库

## Git结构



Git中独有的术语，理解这些术语非常重要：

* 远程仓库（Remote）：也叫作资源库，是远程机器上的代码库。
* 本地库（Repository）：是用户在本地创建的目录，拥有远程库的一个快照，由工作区和版本库构成。
  + 工作区（Workspace）：本地库的根目录中除.git目录以外的内容，存储内容的实际文件；
  + 版本库（.git目录）：是本地库的根目录中的一个隐藏目录.git，用于记录版本信息，Git进行版本控制所需要的文件，则都放在.git文件夹中；
    - 暂存区（Index）：也叫做缓存区，暂存信息存放在.git目录"下的index文件（.git/index）中，用于临时保存内容的修改；
    - HEAD文件：是.git目录下的HEAD文件(.git/HEAD)，指向最后依次提交的结果。
* 分支（Branch）：本地库中默认创建一个主（master）分支，分支意味着你可以从开发主线上分离开来，然后在不影响主线的同时继续工作。

## Git安装

1、进入/local，新建目录git

|  |
| --- |
| [root@192 /]# cd /usr/local/  [root@192 local]# mkdir git |

2、进入git目录，下载git的源码包，最新

|  |
| --- |
| [root@192 local]#cd git;wget https://github.com/git/git/archive/v2.17.0.tar.gz |

3、安装编译源码所需依赖，出现提示输入y即可

|  |
| --- |
| [root@192 local]#yum install curl-devel expat-devel gettext-devel openssl-devel zlib-devel gcc perl-ExtUtils-MakeMaker |

4、解压

|  |
| --- |
| [root@192 local]#tar zxvf v2.17.0.tar.gz |

5、进入目录，执行编译

|  |
| --- |
| [root@192 git-2.17.0]# make prefix=/usr/local/git all |

6、make install

|  |
| --- |
| [root@192 git-2.17.0]#make prefix=/usr/local/git install |

7、配置环境变量，在底部加入

PATH=$PATH:/usr/local/git/bin  
export PATH

|  |
| --- |
| [root@192 local]#vi /etc/profile |

8、编辑环境变量，让配置即时生效

|  |
| --- |
| [root@192 local]#source /etc/profile |

9、测试

|  |
| --- |
| [root@192 git]# git --version  git version 2.17.0 |

# Git命令行操作



## 初始化

|  |
| --- |
| [root@localhost tmp]#mkdir /home/gitrepo  [root@localhost tmp]#cd gitrepo  [root@localhost tmp]#git init |

初始化完成后，就多一个隐藏了.git文件，ls -la能看到具体细节

|  |
| --- |
| [root@localhost gitrepo]# ll .git  total 32  drwxr-xr-x. 2 root root 4096 Nov 26 14:09 branches  -rw-r--r--. 1 root root 92 Nov 26 14:09 config  -rw-r--r--. 1 root root 73 Nov 26 14:09 description  -rw-r--r--. 1 root root 23 Nov 26 14:09 HEAD  drwxr-xr-x. 2 root root 4096 Nov 26 14:09 hooks  drwxr-xr-x. 2 root root 4096 Nov 26 14:09 info  drwxr-xr-x. 4 root root 4096 Nov 26 14:09 objects  drwxr-xr-x. 4 root root 4096 Nov 26 14:09 refs |

## 设置签名

类似于设置了用户信息，需要设置用户名和邮箱

注意：这里设置的签名和远程库的账号、密码没有任何关系

设置本地仓库用户名和邮箱

|  |
| --- |
| [root@localhost gitrepo]# git config user.name caichang  [root@localhost gitrepo]# git config user.email caichangfast@qq.com |

信息放在.git/config配置文件中

|  |
| --- |
| [root@localhost .git]# cat config  [core]  repositoryformatversion = 0  filemode = true  bare = false  logallrefupdates = true  [user]  name = caichang  email = caichangfast@qq.com |

设置全局用户名和邮箱

|  |
| --- |
| [root@localhost .git]# git config --global user.name caichang\_global  [root@localhost .git]# git config --global user.email caichangfast@sohu.com |

存放路径放在/root/.gitconfig，因为是用root用户登录

|  |
| --- |
| [root@localhost .git]# cat /root/.gitconfig  [user]  name = caichang\_global  email = caichangfast@sohu.com |

## 查看状态

注意路径要在新建的版本库位置上，不能在具体的隐藏文件下

|  |
| --- |
| [root@localhost gitrepo]# git status  On branch master  No commits yet  nothing to commit (create/copy files and use "git add" to track) |

## 添加并提交（是2个动作）

|  |
| --- |
| [root@localhost gitrepo]# git add httpd.conf  [root@localhost gitrepo]# git commit httpd.conf |

注意：1、要撤销add，可以采用

|  |
| --- |
| [root@localhost gitrepo]# git rm --cached httpd.conf |

2、这样提交，要输入为什么要做更改的理由，和svn一样

|  |
| --- |
| this is my first commit.  # Please enter the commit message for your changes. Lines starting  # with '#' will be ignored, and an empty message aborts the commit.  #  # On branch master  #  # Initial commit  #  # Changes to be committed:  # new file: httpd.conf  # |

3、修改文件后想再次提交，直接commit就可以

4、提交时采用带 -m "具体message"就可以，这样就不用进vi编辑器了

|  |
| --- |
| [root@localhost gitrepo]# git commit -m "second commit" httpd.conf |

# github



## 注册

网站：https://github.com/

## 新建远程库

新建一个远程库，右上角，new repository，填入对应信息即可。新创建的库有提示，这个提示很重要，可以用命令去实现提交等，但实际往往用一些工具去处理

|  |
| --- |
| …or create a new repository on the command line echo "# mtesting" >> README.md  git init  git add README.md  git commit -m "first commit"  git remote add origin https://github.com/caichang5201314/mtesting.git  git push -u origin master …or push an existing repository from the command line git remote add origin https://github.com/caichang5201314/mtesting.git  git push -u origin master |

|  |
| --- |
| 会得到远程库地址：https://github.com/caichang5201314/mtesting.git |

## 给远程库取别名

不然特别麻烦，每次都要输入一长串远程地址，mtesting为远程库的别名

|  |
| --- |
| [root@localhost gitrepo]# git remote add mtesting https://github.com/caichang5201314/mtesting.git |

4、查看，fetch表示取回地址，push表示推送

|  |
| --- |
| [root@localhost gitrepo]# git remote -v  mtesting https://github.com/caichang5201314/mtesting.git (fetch)  mtesting https://github.com/caichang5201314/mtesting.git (push) |

## 把本地库推送给远程库

需要输入你远程库的用户名和密码

|  |
| --- |
| [root@localhost gitrepo]# git push mtesting master |

5、刷新远程库，就可以看到已推送过去了

## 克隆远程库

1、进入任何你想要clone的目录，如：/tmp/caichang

2、输入命名clone下来

|  |
| --- |
| [root@192 caichang]# git clone https://github.com/caichang5201314/mtesting.git |

## 邀请其他人进入团队协同开发

略

# Git图形界面

1. 安装Git-2.17.0-64-bit.exe

<https://www.cnblogs.com/wj-1314/p/7993819.html>

1. 安装TortoiseGit-2.8.0.0-64bit.msi
2. 安装TortoiseGit-LanguagePack-2.8.0.0-64bit-zh\_CN.msi