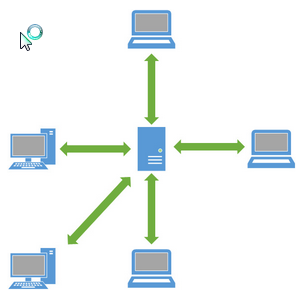
Git 简介

Git是Linus用C实现的一个分布式版本控制工具，注意这里对分布式的强调。不同于Git，像Perforce、SVN和CVS这类版本控制工具都是集中式的。

下面大概介绍下集中式版本控制工具和分布式版本控制工具的区别。

# 集中式

所谓集中式的版本控制，就是在一个系统中只有一个机器是服务端，其他机器全是客户端。以Perforce版本控制为例，在一个系统中会有一个Perforce服务器，所有的代码以及版本信息都保存在这个服务器上。每个客户端可以从服务器get下来一份代码，然后在本地修改，最后submit修改的代码。

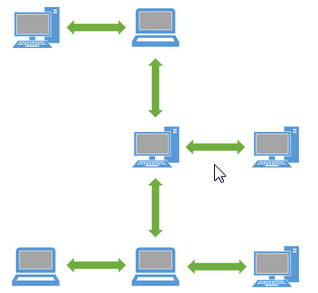


可以看到集中式的版本控制还是有一些问题的：

* 网络依赖性强，工作环境保持网络连接，如果网络断掉了，所有的客户端就无法工作了。
* 安全性较弱，所有的代码以及版本信息保存在服务器中，一旦服务器挂掉了，代码和版本控制信息就丢失了。

# 分布式

而在分布式版本控制系统中，没有服务端/客户端的概念，每台机器都是一个服务器。也就是说，在分布式本版控制系统中，每台机器都有一份代码，并且有代码的版本信息。



所以可以看到Git的优势：

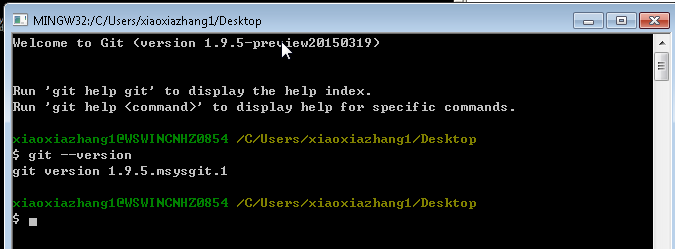
* 每台机器都是一台服务器，无需依赖网络就可以帮自己的更新提交到本地服务器，支持离线工作。当有网络环境的时候，就可以把更新推送给其他服务器。
* 安全性高，每台机器都有代码以及版本信息的维护，所有即使某些机器挂掉了，代码依然是安全的。
* 在Git中，同步更新的方式有很多种，可以把自己的更新推送给别人；也可以生成一个diff的patch，通过邮件方式把这个patch发送给别人。

建议：虽然分布式版本控制没有服务端的概念，但一般在一个Git系统中，为了方便大家交换更新，会找一台机器作为中心服务器，这台机器的目地只是为了方便大家交换更新。即使这台中心服务器挂了，大家依然可以继续工作，只是相互之间交换更新比较麻烦。

# Git安装

Windows版本下载地址：https://git-for-windows.github.io/

由于我主要是在Windows环境下工作，所以下面就主要介绍Windows上面安装Git。一般在Windows下使用Linux/Unix的工具时，需要Cygwin这样的模拟环境。但是已经有人把模拟环境和Git打包好了，msysgit是Windows版的Git，从上面的地址下载，然后按默认选项安装即可。安装完成后，在开始菜单里找到”Git”->”Git Bash”，弹出一个命令行窗口，就说明Git安装成功！



安装完成后，一般都会对本机的Git进行一些基本的配置。下面的命令就是给Git环境配置全局的用户名和邮箱地址，这样每一个从这台机器上提交的更新都会标上这些用户信息。

|  |
| --- |
| git config --global user.name "your user name"  git config --global user.email "your email address"  #举例  git config --global user.name "xiaoxiazhang1"  git config --global user.email "xiaoxiazhang1@hengtiansoft.com" |

# Git命令流

在Git中支持上百个命令，每个命令又有很多的选项，所以初学者看到这些就会有一些恐惧。其实，真正接触过Git一段时间后，会慢慢的发现我们会经常使用的命令也就十几二十个，掌握了这些命令之后就可以满足我们大部分的日常工作了。下面是我根据日常使用整理的一个Git命令流图，包括了一些常用的命令，可以方便自己查阅。不要被密密麻麻的箭头吓到，其实都是比较初级、常用的命令，后面的文章会详细介绍图中命令的用法以及Git中的一些基本概念。注意，图中没有涉及branch、patch等信息，但是后面的文章会进行介绍。

