在使用git来进行版本控制时，为了得一个项目的拷贝(copy),我们需要知道这个项目仓库的地址(Git URL). Git能在许多协议下使用，所以Git URL可能以ssh://, http(s)://, git://,或是只是以一个用户名（git 会认为这是一个ssh 地址）为前辍.

有些仓库可以通过不只一种协议来访问，例如，Git本身的源代码你既可以用 git:// 协议来访问：

git clone git://git.kernel.org/pub/scm/git/git.git

也可以通过http 协议来访问:

git clone http://www.kernel.org/pub/scm/git/git.git

git://协议较为快速和有效,但是有时必须使用http协议,比如你公司的防火墙阻止了你的非http访问请求.如果你执行了上面两行命令中的任意一个,你会看到一个新目录: 'git',它包含有所的Git源代码和历史记录.

   在默认情况下，Git会把"Git URL"里最后一级目录名的'.git'的后辍去掉,做为新克隆(clone)项目的目录名: (例如. git clone http://git.kernel.org/linux/kernel/git/torvalds/linux-2.6.git 会建立一个目录叫'linux-2.6')

  另外，如果访问一个Git URL需要用法名和密码，可以在Git URL前加上用户名，并在它们之间加上@符合以表示分割，然后执行git clone命令，git会提示你输入密码。

示例

git clone robin.hu@http://www.kernel.org/pub/scm/git/git.git

这样将以作为robin.hu用户名访问http://www.kernel.org/pub/scm/git/git.git，然后按回车键执行git clone命令，git会提示你输入密码。

另外，我们可以通过-b <name>来指定要克隆的分支名，比如

$ git clone -b master2 ../server .

表示克隆名为master2的这个分支，如果省略-b <name>表示克隆master分支。

GIT URLS

In general, URLs contain information about the transport protocol, the address of the remote server, and the path to the repository. Depending on the transport protocol, some of this information may be absent.

Git natively supports ssh, git, http, https, ftp, ftps, and rsync protocols. The following syntaxes may be used with them:

* **ssh://[user@]host.xz[:port]/path/to/repo.git/**
* **git://host.xz[:port]/path/to/repo.git/**
* **http[s]://host.xz[:port]/path/to/repo.git/**
* **ftp[s]://host.xz[:port]/path/to/repo.git/**
* **rsync://host.xz/path/to/repo.git/**

An alternative scp-like syntax may also be used with the ssh protocol:

* **[user@]host.xz:path/to/repo.git/**

The ssh and git protocols additionally support ~username expansion:

* **ssh://[user@]host.xz[:port]/~[user]/path/to/repo.git/**
* **git://host.xz[:port]/~[user]/path/to/repo.git/**
* **[user@]host.xz:/~[user]/path/to/repo.git/**

For local repositories, also supported by git natively, the following syntaxes may be used:

* **/path/to/repo.git/**
* **file:///path/to/repo.git/**

Examples

* + **Clone from upstream:**

**$ git clone git://git.kernel.org/pub/scm/.../linux-2.6 my2.6 $ cd my2.6 $ make**

* + **Make a local clone that borrows from the current directory, without checking things out:**

**$ git clone -l -s -n . ../copy $ cd ../copy $ git show-branch**

* + **Clone from upstream while borrowing from an existing local directory:**

**$ git clone --reference my2.6 \ git://git.kernel.org/pub/scm/.../linux-2.7 \ my2.7 $ cd my2.7**

* + **Create a bare repository to publish your changes to the public:**

**$ git clone --bare -l /home/proj/.git /pub/scm/proj.git**

* + **Create a repository on the kernel.org machine that borrows from Linus:**

**$ git clone --bare -l -s /pub/scm/.../torvalds/linux-2.6.git \ /pub/scm/.../me/subsys-2.6.git**