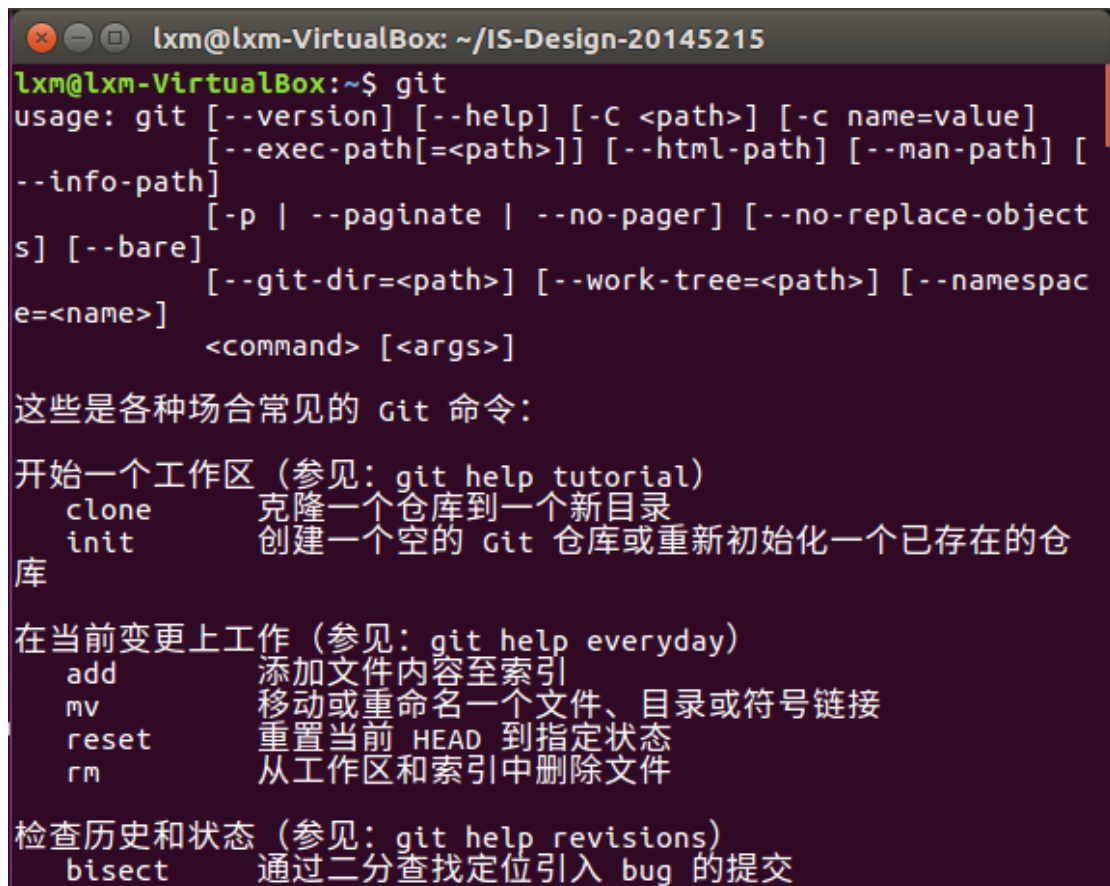


---Git 的安装与配置

首先确定自己的系统有没有安装 git



```
lxm@lxm-VirtualBox: ~/IS-Design-20145215
lxm@lxm-VirtualBox:~$ git
usage: git [--version] [--help] [-C <path>] [-c name=value]
         [--exec-path[=<path>]] [--html-path] [--man-path] [
         --info-path]
         [-p | --paginate | --no-pager] [--no-replace-objects] [--bare]
         [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
         <command> [<args>]

这些是各种场合常见的 git 命令：

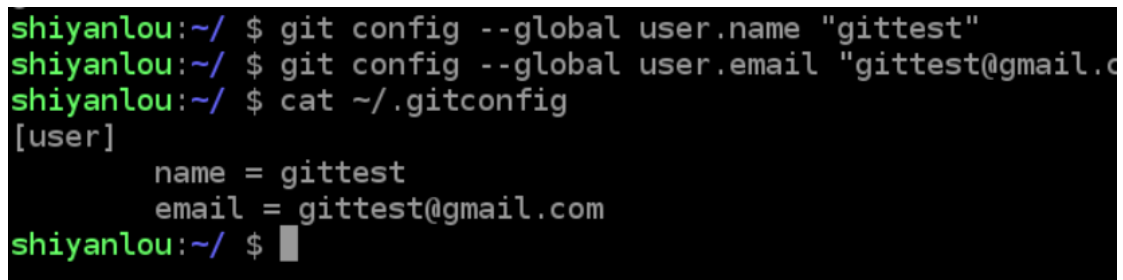
开始一个工作区（参见：git help tutorial）
  clone      克隆一个仓库到一个新目录
  init       创建一个空的 git 仓库或重新初始化一个已存在的仓库

在当前变更上工作（参见：git help everyday）
  add        添加文件内容至索引
  mv         移动或重命名一个文件、目录或符号链接
  reset      重置当前 HEAD 到指定状态
  rm         从工作区和索引中删除文件

检查历史和状态（参见：git help revisions）
  bisect     通过二分查找定位引入 bug 的提交
```

如果没有，输入命令：`sudo apt-get install git` 进行安装。

安装完成，进行配置。



```
shiyanolou:~/ $ git config --global user.name "gittest"
shiyanolou:~/ $ git config --global user.email "gittest@gmail.com"
shiyanolou:~/ $ cat ~/.gitconfig
[user]
    name = gittest
    email = gittest@gmail.com
shiyanolou:~/ $
```

这里的 name 和 Email，需要填你自己的。

--克隆一个仓库

配置好之后，我们现在就需要一个 Git 仓库，有两种方法，一是从已有的仓库里克隆，还有一种就是新建一个仓库。

克隆一个仓库，我们需要知道这个仓库的地址（Git URL），平时在 github 上抄代码也是这么克隆下来的。仓库需要有一个文件夹来放，在根目录里创建一个专门放 git 上克隆来的代码的文件夹。

```

shiyanolou:~/ $ cd ~
shiyanolou:~/ $ mkdir git
shiyanolou:~/ $ cd git
shiyanolou:git/ $ git clone https://github.com/wangzheng0822/algo.git
Cloning into 'algo'...
remote: Enumerating objects: 28, done.
remote: Counting objects: 100% (28/28), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 4737 (delta 10), reused 11 (delta 0), pack-reused 4709
Receiving objects: 100% (4737/4737), 1.11 MiB | 12.00 KiB/s, done.
Resolving deltas: 100% (2202/2202), done.
Checking connectivity... done.
shiyanolou:git/ $ ls
algo
shiyanolou:git/ $

```

这是在实验楼的虚拟环境里,在 GitHub 上 clone 一个列举了很多数据结构代码的代码仓库。这样子就成功克隆了一个仓库了。

```

shiyanolou:git/ $ cd algo
shiyanolou:algo/ (master) $
shiyanolou:algo/ (master) $ ls
c-cpp      go          LICENSE    python     StackBaseArray.java
csharp     java        notes      README.md  swift
DynamicStackBaseArray.java javascript  object-c   rust       typescript
f21        kotlin      php        scala
shiyanolou:algo/ (master) $

```

--向仓库传代码

1. 创建或修改文件
2. 使用 **git add** 命令添加新创建或修改的文件到本地的缓存区 (Index)
3. 使用 **git commit** 命令提交到本地代码库

git add <file>(这里的 file 可以通过 git status 查看有哪些是 untracked 的)

```

root@ubuntu:~/datastructure/week04# git status
On branch master
Your branch is up-to-date with 'origin/master'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    ../clonetest/
    ../project/

nothing added to commit but untracked files present (use "git add" to track)

```

接下来 git commit

```
root@ubuntu:~/datastructure# git add project/
root@ubuntu:~/datastructure# git commit -m "create js file"
[master 0b0503b] create js file
6 files changed, 11 insertions(+)
create mode 100644 project/README.md
create mode 100644 project/file1
create mode 100644 project/file2
create mode 100644 project/file3
create mode 100644 project/newfile
create mode 100644 project/week04/yy/test.c
```

再次 git status

```
root@ubuntu:~/datastructure# git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        clonetest/

nothing added to commit but untracked files present (use "git add" to track)
```

这样子，我们就把想要更改的，提交到了本地仓库（刚才克隆下来的），这时候我们要来传代码了。

首先，利用 `git pull origin master` 命令，同步远方仓库和本地仓库（更新本地仓库）

```
shiyancelou:algo/ (master) $ git pull origin master
From https://github.com/wangzheng0822/algo
 * branch                master      -> FETCH_HEAD
Already up-to-date.
```

再利用 `git push` 命令，向远方仓库传代码（也就是你改动的地方）
需要你输入用户名和密码，输入完成之后，就成功了。

```
shiyancelou:algo/ (master) $ git push origin master
Username for 'https://github.com': sonhg
Password for 'https://sonhg@github.com':
```

一定要注意：每次 push 的时候先要 add 和 commit，不然不会更新的