date: 2021-06-01 18:22 description: 'Colinsoft Studios Linux course' status: draft title: GitHub Description: 科林明伦Linux课堂笔记

GitHub介绍与使用

GitHub是一个项目托管平台， 大量的私有工程或Free开源工程在该平台托管发布，聚集了超过900万的开发者 ， 平台中托管累计超过1000万+ ， 以学习为目的使用GitHub可以帮助学生了解一个开源项目的全部细节，通过Github中的Commit功能，可以让同学们阅览一个项目从0到发布上线的全部流程，也可以通过Commit观察制作者对代码的每一次修改和添加，学习编码经验、见识更好的设计编码思路，开发者使用GitHub可以直接重用或二次发开上面的Free工程，大大节省开发时间，避免重复制造开发成本。
总而言之无论是计算机相关专业的学生还是软件工程师GitHub都是不可或缺的重要工具！

GitHib官方网站地址 : GitHub

Github网站介绍：

首页标签：

- Explore #探索
- Topics #专题标签，使用者可以根据自己的语言或技术需求阅览专题相关内容，例如C++专题、Python专题等等..
- Trending #趋势，社区会推送展示一些最近比较流行热门的项目、实例或文章
  - 可以在Trending中修改用户语言，编程语言及时间轴
- Collections #收藏
- Events #事件

项目中值得注意的文档与选项：

LICENSE （许可证）

- MIT
- Apache 2.0
- GPL
- LGPL
- BSD

如果一个工程中LICENSE中出现以上备注，那么这份代码是完全开源免费的，这些声明给开发者最大的权力以及最少的限制，您可以随意使用制定这份代码， 如果未出现以上声明那么使用该代码时可能需要额外的授权或许可，及时与开发者或版权方沟通，如果擅自商用可能会出现版权或法务问题！

README （说明书）

在GitHub上托管的每一个项目中都有一份README文档，并且这份文档会在项目目录下方直接显示出来，该文档一般是项目说明书，虽然不会描述所有项目细节，但是会将项目概述、使用方式、注意事项等等内容呈现给使用者，让使用者快速了解该项目及使用方式， 如果你在Github中找到一份不错的开源项目并想快速了解它，那么打开README，仔细阅读一遍吧。
README项目说明的内容一般采用Markdown轻量级标记语言编写，标准文件后缀为md， 例如: README.md

Code

点击该选项可以展开项目目录，内包含项目资源、项目说明、项目源码等等....

Issues

点击该选项可以查看项目使用者对项目的意见，Bug反馈以及制作者的解决情况，经典的问答互动窗口.

Pull Requests

这里可以查看工程被使用情况，有哪些用户拉取使用了该项目（但通常设置为Private私有外部不可见）

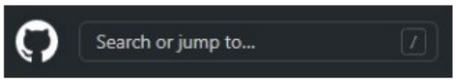
Watch

如果你对某一个开发者感兴趣，可以使用Watch关注开发者动态信息或软件更新信息。

Star

在GitHub社区中Star(点赞)是评定一个开发者或一个项目影响力的主要因素，一个作者的项目获取的Star数量越多，表示该项目在社区中很有价值且影响力非凡。 如果社区中哪一篇文章或实例帮助到了你，不妨点一下Star支持以下!

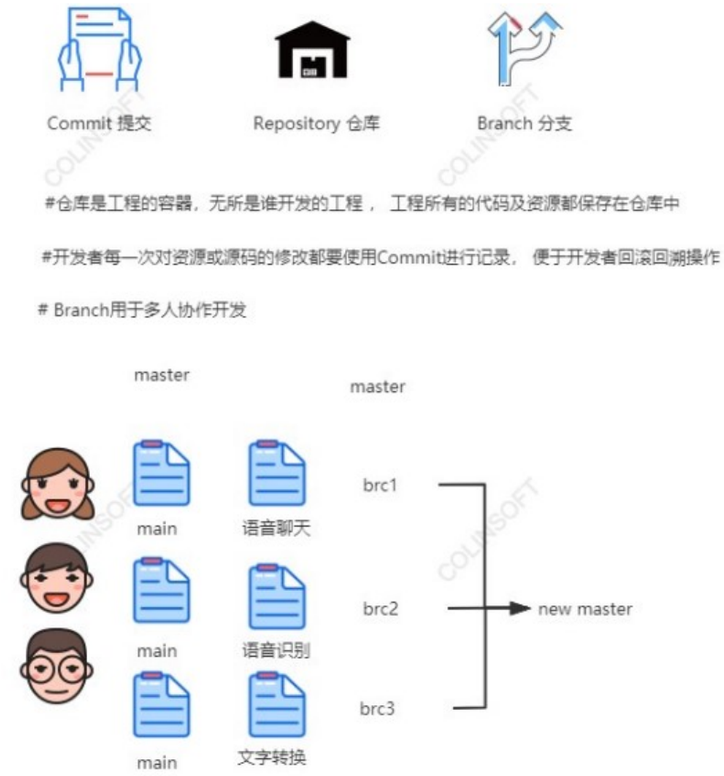
在GitHub中通过关键字查询资料或项目



可以在本窗口直接输入你想查询的特定项目或实例（中/英）,如果你想查询某一类内容，可以通过下面三种关键词搜索方式查找：

- awesome c++ #在awesome系列中查找比较酷的c++项目
- c++ sample #使用sample关键字查找关于c++的例子或demo
- c++ tutorial #使用tutorial关键字查找关于c++的资料

GitHub三要素



Git介绍

Git是一款与GitHub协同的版本控制软件即(分布式版本控制系统)是一个开源的分布式版本控制系统，可以有效、高速地处理从很小到非常大的项目版本管理。
我们可以使用Git软件将我们本地的工程代码Push推到GitHub云端仓库中，也可以非常方便地将云端仓库中的很酷的项目克隆到本地， 总而言之Git是一款非常高效的GitHub仓库管理系统，可以随心随遇的通过Git访问修改云端仓库及项目。

Git软件官网下载入口 : GitDownloads

Git基本使用与常用命令

Git与GitHub账户关联及测试关联

- 测试Git是否关联了账户

```
ssh -T git@github.com
```

- 显示所有Git配置列表

```
git config --list
```

- 修改配置中的用户名及账户名(邮箱),如果配置中没有这两项，也可以通过下面的两个命令添加。

```
git config --global user.email "your email"
git config --global user.name "your username"
```

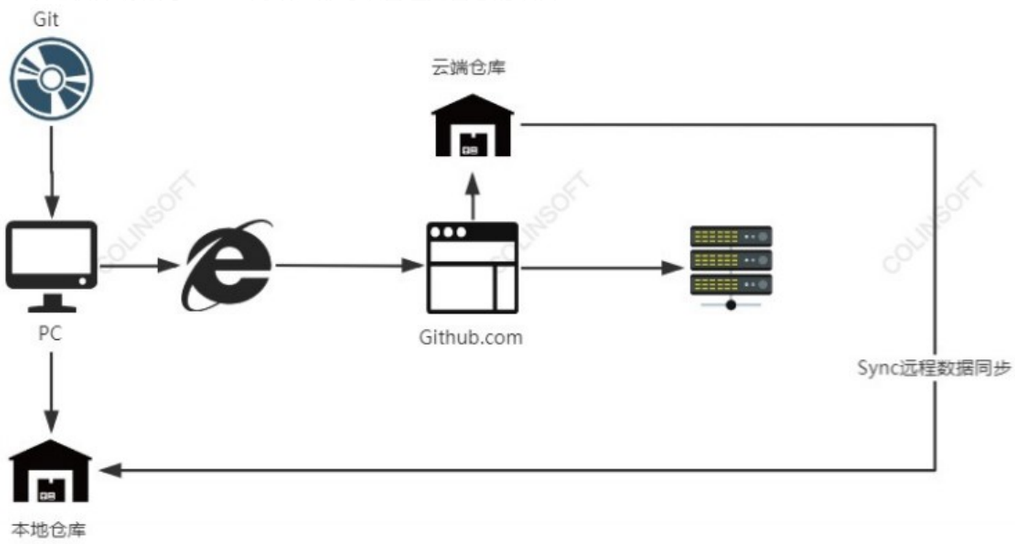
- 创建访问密钥ras_keys

```
ssh-keygen -t rsa -C "your email"
```

1.完成如上操作后打开.ssh/id_rsa.pub文件,拷贝其中内容.
2.打开GitHub->User_menu(用户头像下的下拉菜单)->settings->SSH or GPG keys->new SSH keys->将复制的内容粘贴，而后点击add SSH keys,完成关联.
3.通过ssh -T git@github.com 测试是否关联成功 , 如果显示 "Hi 账户名! You've successfully",表示已关联成功.

Git操作逻辑及流程概述：

- 使用Git本地软件或访问GitHub网站，都可以管理托管您的项目:



- Git缓存到仓库使用及上传：

- Git创建本地仓库

```
git init
```

- 将资源添加到Git缓冲区或者删除：

```
git add files #添加资源到Git缓冲区
git rm files #从缓冲区删除资源
```

- 将缓冲区的内容提交到本地Git仓库中:

```
git commit -m "description" #对本次提交添加描述信息
```

- 查看本地仓库状态命令:

```
git status
```

- 对云端仓库创建并起别名为origin或删除

```
git remote add origin "git@github.com:Repository/Repository.git"
git remote remove origin
```

- 本地仓库与云端仓库同步合并

```
git push origin master
```

- 如果Push同步仓库失败，那么先拉取合并云端仓库再次推即可:

```
git pull --rebase origin
```

- 从GitHub当中下载克隆项目到本地

```
git clone 工程地址(SSH)
```

MarkDown语法介绍

- Markdown标准文件后缀为.md， 注意md语法中修饰符后面一般都有空格。

```
# Content (一级标题,字体较大有下划线效果)
## Content（二级标题，字体较大有下划线效果）
### Content（三级标题）
#### Content（四级标题）
##### Content（五级标题）
###### Content（六级标题）
```

- 如果要添加一段普通的文本直接编辑即可，无需任何修饰符修饰，但是注意：如果需要换行md语言中需要用

```
  这是一段普通的测试文本<br>
  #在md中<br>为换行符。
```

- 可以使用星号对文本进行修饰，显示效果为数据项:

```
#如果你要指定一个数据项，用星号即可，如果该项里还有子项可以在它下方加几个空格，而后再使用星号。子项就创建成功了，而且它们
* Content
* Content
* Content #A项
    * Content #A(1)子项
      * Content #A(1)(1)子项
#在这里没有任何效果，但是如果你以markdown格式预览，就可以看到它的作用
```

- 可以使用>指定层级结构效果:

```
#不同的>数量代表不同的层级
> Content #一级
>> Content #二级
>>> Content #三级，依此类推
```

- 关键字:

```
#如果你要在一段话中凸显某个关键词，可以使用反引号实现
这是一段普通的测试文本，用于凸显`关键词`。
```

- 你可以再md中插入一段代码，无论他是任何编程语言或是终端语言，只要修饰符准确，就可以显示对应的效果:

```
C语言:
```c<br>#这里编写c代码<br> ```
```

```
C++语言:
```cpp<br>#这里编写c++代码<br> ```
```

```
Python语言:
```python<br>#这里编写python代码<br> ```
```
```

```
Java语言:
```java<br>#这里编写Java代码<br> ```
```

```
Shell语言:
```bash<br>#这里编写命令或脚本语言代码<br> ```
```

注意：以上所有代码显示效果，编辑时`<br>`都替换为`Enter`回车，编辑时不要将修饰符与代码放在一行!

- 在Markdown中插入一个可访问的网页超链接：

```
[百度搜索](https://www.baidu.com "悬停文字") #超链接已生成,点击即可访问
```

- 在Markdown中插入一张图片，可以是本地图片也可以使用网络图片：

```
![图片别名](图片地址) #与插入超链接大致相同，只不过起始位置有叹号修饰。
```

当然Markdown语法远不止如此，还可以在md中将数据可视化，例如加入图表，表格等等，如果感兴趣大家可以自行在网上查阅关于md的更多内容。

END