

# kangear的专栏

目录视图

摘要视图

RSS 订阅

个人资料



kangear



访问： 244478次  
积分： 6664分  
排名： 第806名

原创： 281篇 转载： 210篇  
译文： 3篇 评论： 248条

文章搜索

文章分类

- LCD (20)
- Flash RAM (6)
- Camera (3)
- No OS (2)
- Linux应用程序编程 (58)
- Bug (21)
- MCU (5)
- Hardware (5)
- X210 (15)
- u-boot (28)
- Qt (12)
- Linux设备驱动 (16)
- Linux系统管理 (48)
- touchscreen (12)
- Android (61)
- 相关学习资源 (1)
- Linux-kernel (11)
- Android\_USB摄像头 (6)
- Maliit (3)
- C/C++ (58)
- Kconfig (3)
- gcc/arm-linux-gcc (7)
- ARM系统 (44)
- Linux内核模块 (9)

有奖征资源，博文分享有内涵 4月推荐博文汇总 第六届中国云计算大会（大数据读书汇）

## [置顶] Git深度使用经验总结

2013-10-27 14:38 1176人阅读 评论(2) 收藏 举报

已经开始使用Git了，有些时候一般的问题只能Google就能找到答案，但是有些时候不能找到答案，因为遇到的问题都不知道该用什么关键字搜索。很多是在Git中文的QQ群中讨论出来的，把一些经验总结一下：

### 1.Git子模块

使用情况：例如Android系统代码和app之间可以使用这种submodule的方法来管理。快速上手：

\$ git submodule add url path/to/name

\$ git submodule init

\$ git submodule foreach git pull

其它情况下submodule不适用，可以用subtree来代替。

### 2.同步一个COMMIT

\$ git cherry-pick xxx

可以同步一个commit到本分支

### 3.Git打TAG

打TAG也就是发布版本

\$ git tag -a v1.2 -m "version 1.4"

\$ git push --tags

如果还不能理解可以到这里看看是linus是怎么给linux内核打的TAG，TAG看起来像什么：<https://github.com/torvalds/linux/releases>

### 4.查看status详情

\$ git add xxx

\$ git diff --cached

这样可以在commit之前先看一下修改详情。

### 5.不产生无用的merge的同步

有这么一种情况，用一个分支专门同步代码提供商的代码的时候，如果一般的pull会不断的产生一个merge看起来会很烦，用下边的使用添加一个--rebase就不会产生无用的merge了

\$ git pull --rebase origin master

### 6.关于stash

适用情况：做了修改后，还没有add commit等等后续工作，现在突然要切换分支做其它事情，默认情况下你在这个分支修改的代码会被带到切换过去的分支中。可以先把你修改的保存起来。这些修改可以再还原过来。

\$ git stash -u

ubifs/file system (0)
usb (5)
V4L2/H264 (8)
GDB/arm-linux-gdb (7)
S5PV210 (26)
数据结构与算法 (5)
JNI/NDK/HAL (10)
1.打印机 (6)

文章存档
2014年05月 (3)
2014年04月 (6)
2014年03月 (8)
2014年02月 (10)
2014年01月 (4)
展开

阅读排行
Android中间件开发----Wi (3640)
Jlink调试S5PV210 (3152)
用eclipse调试uboot工程 (3099)
BMP转成C语言数组文件 (2852)
warning:statement has i (2158)
编译内核模块常见问题 (2105)
Android中设置Ethernet (2105)
基于v4l2的webcam应用, (2071)
error:Target dll has beer (1978)
单片机串口调试丢包验证 (1918)

评论排行
Jlink调试S5PV210 (53)
基于v4l2的webcam应用, (47)
motion的移植和使用 (16)
用eclipse调试uboot工程 (13)
远程监控系统项目总结 (12)
裸机建立C语言环境一设 (11)
基于v4l2 ffmpeg x264的 (10)
mini2440的串口在Qt上 (9)
tiny210v2的u-boot (8)
linux内核: No oob sche (7)

推荐文章
------

最新评论
2014, 我来了 kanglear: @jingxia2008:文中提到我有一个毛病, 急于开始一件事, 而总忘记总结之前的事。正在“吃药”。
Android调试底层服务 kanglear: @jingxia2008:是的, 都是通过adb实现更方便一些。adb 也可以通过wifi调试。不过文...
Android与Wifi jingxia2008: mark
2014, 我来了 jingxia2008: 内容有点少
Android调试底层服务 jingxia2008: adb 是一个方法, 如果 adb 不可用的时候, 还可以使用网络传输文件到 android 文件系统

\$ xxxx 随便你的操作

\$ git stash pop

注意:-u是代表是也把添加的新文件（术语是未跟踪）也藏起来，一般是要有这个u的。

### 7.恢复一个COMMIT

如果一个COMMIT你不想要了，想要去除，可以考虑使用以下的方法：

\$ git revert xxxx

这个就可以去掉这个COMMIT的改动，这个是明式的去掉，如果你又后悔了，还可以再次恢复。

### 8.安装最新Git版本

能轻易安装的一般不会是最新的，用apt-get install便是如此。下面用离线的方式安装Git。

a先到<https://github.com/git/git/releases>看最新版本，然后复制链接。红色随具体版本变化。

\$ apt-get remove git # 卸载现有

\$ wget <https://github.com/git/git/releases/tag/v1.8.4.3> (链接地址)

\$ tar xvfz v1.8.4.3.tar.gz

\$ cd git-v1.8.4.3

\$ make prefix=/opt/git all

\$ make prefix=/opt/git install

\$ 添加/opt/git/bin到环境变量中 /etc/environment中，或者其它。

### 9.安装subtree

\$ git clone <https://github.com/apenwarr/git-subtree>

\$ cp git-subtree/git-subtree.sh /opt/git/bin/git-subtree

关于使用的subtree的几个帖子：

《使用git subtree集成项目到子目录》

《一个成功的Git分支模型》

### 10.Git之本地忽略

这个分同种情况：a是本地永久忽略，效果的gitignore一样，只不过不适于写到gitignore中而已，可以自己建立一个本地独享的gitignore，然后git config --global core.excludesfile 文件的绝对路径，也可以直接将本地要忽略的文件添加到.git/info/exclude中。不过上述都是针对没有跟踪的文件来说的，如果文件已经被跟踪了你如果在本地想要忽略它的改动，就不能使用以上的方法了。这正情况b。通俗地讲比如一个编译Android的脚本在其它电脑上都是使用的-j32来编译的，但是你的电脑配置没有别人的好，不能开到-j32，但是这个脚本是已经跟踪过的，你修改了就会在每次的git status中看到。对于这种情况Git有一个忽略改动的方法：

\$ git update-index --assume-unchanged /path/to/file #忽略跟踪

\$ git update-index --no-assume-unchanged /path/to/file #恢复跟踪

motion的移植和使用  
kangear: @h\_2010007:关于MySQL的问题, 已经解决补充到文章结束。

motion的移植和使用  
kangear: @chenhanzhun:那个是开发板的ip, 需要连接网络, 我用的是局域网, 你可以用有线或者无线。

motion的移植和使用  
逆风微积分: 博主你好, 我想问下, 那个浏览器地址是怎么什么相关的? 还有开发板和PC机需要连接网线吗?

U-boot中实现Yaffs2+HwEcc  
kangear: @lhqsine:我有心但精力不够, 我荒废了很多事了。:

U-boot中实现Yaffs2+HwEcc  
lhqsine: 楼主还在更新吗

```

root@kangear: /work/module/vold
root@kangear: /work/module/vold 89x24
root@kangear: /work/module/vold# git status
# 位于分支 master
# 尚未暂存以备提交的变更:
#   (使用 "git add <file>..." 更新要提交的内容)
#   (使用 "git checkout -- <file>..." 丢弃工作区的改动)
#
#       修改:      Android.mk
#
修改尚未加入提交 (使用 "git add" 和/或 "git commit -a")
root@kangear: /work/module/vold# git update-index --assume-unchanged Android.mk
root@kangear: /work/module/vold# git status
# 位于分支 master      http://blog.csdn.net/kangear
# 无文件要提交, 干净的工作区
root@kangear: /work/module/vold# git update-index --no-assume-unchanged Android.mk
root@kangear: /work/module/vold# git status
# 位于分支 master
# 尚未暂存以备提交的变更:
#   (使用 "git add <file>..." 更新要提交的内容)
#   (使用 "git checkout -- <file>..." 丢弃工作区的改动)
#
#       修改:      Android.mk
#
修改尚未加入提交 (使用 "git add" 和/或 "git commit -a")
root@kangear: /work/module/vold#

```

之后你在本地修改/path/to/file这个文件, Git也不管它了。就实现了本地忽略。

## 11.Gitlab用户头像的问题

如果装的Git服务器是GitLab的话, 是可以使用用户头像的, GitLab管理头像的一般方法是用gravatar, 这一点和github一样, 是关联email地址的, 先在上边注册一个账户, 上传一个头像并关联某一个邮箱, Gitlab或者Github会自动去gravatar上拉取你的个人头像。

## 12.Git打包源码

对Git管理的源码进行压缩打包, 如果使用tar xvzf xxxx.tag.gz xxxxx的话并不是一个很好的选择, 因为会将.git/目录下的中间文件全部压缩, 如果只想要某一个版本的源码。Git提供了archive.它会给你打包一份纯净的代码。当然这个只适用于发一个版本的源码, 而不是备份Git管理的整套源码。

\$ git archive xxxx

xxx代表SHA-1 Hash值。

## 13.Git反向打补丁

一套没有管理有源码, 中间有功能A, 且也有功能A的补丁文件, 可以使用补丁文件进行反打补丁来实现去掉功能A。

\$ git apply -R

注: 只是收集, 还没有完整测试[这里](#)。

## 13.关于Git服务器选择和安装

选择建议GitLab, 安装我参考加总结这几个帖子:

1. 《[gitlabhq / doc / install / installation.md](#)》根据国情为官方的安装方法添加了说明的安装文档。
2. 《[在 Ubuntu 12.04 上安装 GitLab6.x](#)》一个对安装过程打印信息都记录的安装文档, 安装过程中可以对比安装过程。
3. 《[重设MySQL root密码](#)》中间遇到的问题MySQL密码找不到了, 这篇文章给了答案。

## 14.合并commit

1. 《[使用git合并多个提交](#)》
2. 《[git如何合并只有两个commit到一个?](#)》

## 15.Git只clone最新版本代码不要.git目录

这个也是在群中出现了问题, 我曾经也有过这样的需求, 想法是好的, 但是理解是错误

的。如果只要最后一个版本的代码，而不要.git目录中的东西，有办法：

```
git clone --depth=1 git://someserver/somerepo dirformynewrepo
```

```
rm -rf !$.git
```

但是这又说明不是要参与开发的，是使用的。使用一般是要使用代码开发最新发布版本，如果只要最新的提交版本是没有用处的。因为它不是稳定的。这样就引入了发布版本这一说明，Git中是打Tag，一个tag是一个版本，可以以刚才的需求去找最后一个tag版本。github上是可以直接下载某一个tag版本的。如果要用git clone来完成，会复杂一些。现在的github是结合了web的，例如github或者是自己搭建的gitlab都是可以直接从web上下载指定的tag版本。而没有用于版本控制的.git目录。

## 16.关于pull和merge

都在提倡添加--no-ff选项。

## 17.删除子模块-解决子模块自动找上门的问题

也许你并不想要用子模块，但是有时它会自动找上门，那么你要了解一下怎么去除了。怎么个自动上门，且听我慢慢道来。

在项目A中，如果用git clone了另外一个项目B，或许你只是想把项目B的代码添加到项目A中，但是这时执行git status/add的时候会发现项目B中的代码，并不会被添加，也就是说不能被跟踪。这是为什么呢，这里因为会自动将项目B做为了子模块管理，就是因为它是一个包含.git目录的完整的工程。解决这个问题有同种方法：

1.在clone项目b的后立即将项目B中的.git目录删除

2.git rm --cached path/to/B

然后再执行git status/add 之类的命令就会可以正常的跟踪了。

## 18.恢复一个文件到之前的某个提交

这个种情况一般是出在某个bug解决后整理规范代码的时候，解决这个bug的时候可能添加了不少的调试代码，最后解决后，有些调试信息只为了了解代码流程，对以后并没有保留的意义，所以就要将其恢复开始解决这个bug之前的提交。xxxxx代表开始解决这个bug之前的一个提交的哈希码。

对于已经跟踪的文件：

1.git reset xxxxxx path/to/file

2.git checkout path/to/file

对于没有跟踪的文件：

1.git rm path/to/file

## 19.合并时只产生合并提交

还有其它说法：合并时不合并历史；合并时将所有commit合为一个。

git pull --squash another

话说一般用于master分支，像linux的linux内核一样。

更多信息见《Git merge no history commit》《git merge --squash介绍》

## 20.关于GUI Clients

这里《GUI Clients》。

## 21.git pull 前简单审查

情况是这样的：本人管理的是develop分支，其他人管理的他们自己的分支。所以应该每次改动的都应该是package/apps下的文件，但是有一种情况是我担心他们会不小心改到系统代码并提交。在合并之前查看都改了哪些文件。这样就能有效地防止误修改。

和老友沟通后，得出以下结论：

- 1.让他们添加本地忽略，忽略他们一定不会改的目录。
- 2.在每次合并代码的时候，看一下每个分支都修改了哪些文件，有没有系统文件被修改。
- 3.使用grrit。
- 4.先git fetch 再merge。

最后选择了第二种。因为git pull后，提交的时候会显示出这次合并代码都有哪些修改。

22.Gitk 不错的查看Log的工具

23.Git遇上代码规范

当Git遇上代码规范会有些事情要处理好，这里写先写个经典的：

1.检测自己要提交的代码中是否有table

一般多人开发的程序中不准使用table代替空格的，以下是Google的Java代码规范(中文)中的截图：

2.3.1 Whitespace characters

Aside from the line terminator sequence, the ASCII horizontal space character (0x20) is the only whitespace character that appears anywhere in a source file. This implies that:

http://blog.csdn.net/kangear

1. All other whitespace characters in string and character literals are escaped.

2. Tab characters are not used for indentation

所以在提交代码前自检一下自己代码中是否有table是很有必要的。

\$ git diff | grep -P "\t" | grep "+"

如果tab已经存在，这里有很好的解决方案，所以项目代码不要急于commit前期工作很重要。

更多 0

上一篇 Android shell中特有命令

下一篇 LDD3-最简字符设备驱动freg (alloc\_chrdev\_region)

顶

10

踩

0

主题推荐

经验gitlinux内核版本控制解决方案

猜你在找

- 如何取消office 2010启动配置过程

好东西分享给大
- ssh 密钥访问

自己对杨一夫创业心得的感悟。
- JavaScript中Web应用程序事件处理

你不懂我 我不怪你
- 如何在eclipse中运行launcher4.0代码

SVG多分辨率、自适应缩放解决方案
- 警察与小偷的实现之一客户端与服务端通信

js 判断变量是否 字符串变量

免费学习IT4个月,月薪12000

中国[官方授权]IT培训与就业示范基地,学成后名企直接招聘,月薪12000起!

查看评论

1楼 sunny-dream 2014-04-14 23:35发表



好东西 先感谢 后mark

Re: kangear 2014-04-15 10:24发表



回复sunshine\_love: 幸会幸会。

您还没有登录,请[登录](#)或[注册](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

- 全部主题
- JavaVPNAndroidiOSERP
- IE10EclipseCRMJavaScriptUbuntuNFC
- WAPjQuery数据库BIHTML5Spring
- ApacheHadoop.NETAPIHTMLSDK
- IISFedoraXMLLBSUnitySplashtop
- UMLcomponentsWindows MobileRails

<a href="#">QEMU</a>	<a href="#">KDE</a>	<a href="#">Cassandra</a>	<a href="#">CloudStack</a>	<a href="#">FTC</a>	<a href="#">coremail</a>	<a href="#">OPhone</a>	<a href="#">CouchBase</a>	<a href="#">云计算</a>	<a href="#">iOS6</a>
<a href="#">Rackspace</a>	<a href="#">Web App</a>	<a href="#">SpringSide</a>	<a href="#">Maemo</a>	<a href="#">Compuware</a>	<a href="#">大数据</a>	<a href="#">aptech</a>	<a href="#">Perl</a>	<a href="#">Tornado</a>	<a href="#">Ruby</a>
<a href="#">Hibernate</a>	<a href="#">ThinkPHP</a>	<a href="#">Spark</a>	<a href="#">HBase</a>	<a href="#">Pure</a>	<a href="#">Solr</a>	<a href="#">Angular</a>	<a href="#">Cloud Foundry</a>	<a href="#">Redis</a>	<a href="#">Scala</a>
<a href="#">Django</a>	<a href="#">Bootstrap</a>								