

公告

昵称:shines77
园龄:7年7个月
粉丝:101
关注:8
+加关注

2018年8月 日历表

搜索

查找框 谷歌搜索

常用链接

- 我的随笔
我的评论
我的参与
最新评论
我的标签

最新随笔

- 1. How to fix Eclipse Tomcat startup timeout in 45 seconds?
2. [Android]使用 Eclipse 给 APK 签名时遇到的两个问题及解决办法
3. [Android]Eclipse 无法正常更新 Android SDK Manager 的解决办法(使用 GoAgent)
4. [Android]Eclipse 安装 ADT(Android Development Tooling) 失败的两种解决办法
5. 一个无锁消息队列引发的血案(六)——RingQueue(中) 休眠的艺术 [续]
6. 关于 VS 2010 和 VS 2013 的警告 LNK4042
7. 一个无锁消息队列引发的血案(五)——RingQueue(中) 休眠的艺术
8. [八卦]如何通过搜索引擎让一个人无所遁形?
9. 一个无锁消息队列引发的血案(四)——月:RingQueue(上) 自旋锁
10. 一个无锁消息队列引发的血案(三)——地:q3.h 与 RingBuffer

我的标签

- Lock-Free(11)
无锁队列(10)
无锁消息队列(7)
消息队列(7)
互斥锁(7)
自旋锁(7)
多线程编程(7)
RingQueue(7)
spin-lock(7)
mutex(7)
更多

随笔分类(80)

- .Net(1)
Android(3)
C/C++(26)
CPU优化(9)
DirectUI
FFmpeg(4)
Git(3)
Go语言
Java(4)
Libevent(2)
VPS与云主机
多线程编程(6)
服务器管理
软件应用(3)
设计模式(1)
数据结构(7)
网络编程(4)
游戏开发(1)
杂谈(6)

随笔档案(54)

- 2015年6月 (1)
2015年2月 (3)
2015年1月 (8)
2014年12月 (1)
2014年7月 (1)
2014年6月 (4)
2014年5月 (5)
2014年1月 (4)
2013年12月 (2)
2013年11月 (3)
2013年9月 (2)
2013年8月 (11)
2013年7月 (1)
2013年4月 (5)
2013年3月 (3)

最新评论

- 1. Re: 一个无锁消息队列引发的血案(三) ——地:q3.h 与 RingBuffer
关于head.second说成是最新一个已经成功提交数据的位置有些疑问,好像跟楼主要的银行账号排队的例子有些冲突,楼主能不能详细解释一下 --IT\_chen
2. Re: 一个无锁消息队列引发的血案(一)

Git如何永久删除文件(包括历史记录)

有时候不小心上传了一些敏感文件(例如密码),或者不想上传的文件(没及时或忘了加到.gitignore里的),

而且上传的文件又特别大的时候,这将导致别人clone你的代码或下载zip包的时候也必须更新或下载这些无用的文件,

因此,我们需要一个方法,永久的删除这些文件(包括该文件的历史记录).

首先,可以参考 github 的帮助:

https://help.github.com/articles/remove-sensitive-data

步骤一: 从你的资料库中清除文件

以Windows下为例(Linux类似), 打开项目的Git Bash,使用命令:

```
git filter-branch --force --index-filter 'git rm --cached --ignore-unmatch path-to-your-remove-file' --prune-empty --tag-name-filter cat -- --all
```

其中, path-to-your-remove-file 就是你要删除的文件的相对路径(相对于git仓库的跟目录), 替换成你要删除的文件即可. 注意一点. 这里的文件或文件夹. 都不能以 '/' 开头. 否则文件或文件夹会被认为是从 git 的安装目录开始.

如果你要删除的目标不是文件,而是文件夹. 那么请在 'git rm --cached' 命令后面添加 -r 命令. 表示递归的删除(子)文件夹和文件夹下的文件. 类似于 'rm -rf' 命令.

此外,如果你要删除的文件很多,可以写进一个.sh文件批处理执行,如果文件或路径里有中文,由于MinGW或CygWin对中文路径设置比较麻烦,你可以使用通配符\*,例如: sound/music\_\*.mp3, 这样就把sound目录下以music\_开头的mp3文件都删除了.

例如这样,新建一个 bash 脚本文件. del-music-mp3.sh:

```
#!/bin/bash
git filter-branch --force --index-filter 'git rm --cached --ignore-unmatch projects/Moon.mp3' --prune-empty --tag-name-filter cat -- --all
git filter-branch --force --index-filter 'git rm --cached --ignore-unmatch sound/Music_*.mp3' --prune-empty --tag-name-filter cat -- --all
```

如果你看到类似下面这样的,就说明删除成功了:

```
Rewrite 48dc599c80e20527ed902928085e7861e6b3cbe6 (266/266)
# Ref 'refs/heads/master' was rewritten
```

如果显示 xxxxx unchanged, 说明repo里没有找到该文件, 请检查路径和文件名是否正确.

注意: 补充一点, 如果你想以后也不会再上传这个文件或文件夹, 请把这个文件或文件夹添加到.gitignore文件里, 然后再push你的repo.

步骤二: 推送我们修改后的repo

以强制覆盖的方式推送你的repo, 命令如下:

```
git push origin master --force --all
```

这个过程其实是重新上传我们的repo, 比较耗时, 虽然跟删掉重建一个repo有些类似, 但是好处是保留了原有的更新记录, 所以还是有些不同的. 如果你实在不在意这些更新记录, 也可以删掉重建, 两者也差不多, 也许后者还更直观些.

执行结果类似下面:

```
Counting objects: 4669, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4352/4352), done.
Writing objects: 100% (4666/4666), 35.16 MiB | 51 RiB/s, done.
Total 4666 (delta 1361), reused 0 (delta 0)
To https://github.com/defunkt/github-gem.git
+ beh839d...81f21f3 master -> master (forced update)
```

为了能从打了 tag 的版本中也删除你所指定的文件或文件夹. 您可以使用这样的命令来强制推送您的 Git tags:

```
git push origin master --force --tags
```

步骤三: 清理和回收空间

虽然上面我们已经删除了文件, 但是我们的repo里面仍然保留了这些objects, 等待垃圾回收(GC), 所以我们要用命令彻底清除它, 并收回空间.

命令如下:

```
rm -rf .git/refs/original/
git reflog expire --expire=now --all
git gc --prune=now
Counting objects: 2437, done.
# Delta compression using up to 4 threads.
# Compressing objects: 100% (1378/1378), done.
# Writing objects: 100% (2437/2437), done.
# Total 2437 (delta 1461), reused 1802 (delta 1048)
git gc --aggressive --prune=now
Counting objects: 2437, done.
# Delta compression using up to 4 threads.
# Compressing objects: 100% (2426/2426), done.
# Writing objects: 100% (2437/2437), done.
# Total 2437 (delta 1483), reused 0 (delta 0)
```

注: 绿色字部分是命令执行后的结果.

现在你再看看你的.git目录文件大小是不是变小了.

参考自:

- http://whoop.sinaapp.com/blog/article/21
http://blog.csdn.net/meteor1113/article/details/4407209

分类: 软件应用, Git

标签: Git, 永久删除文件, rm, filter-branch, removing sensitive data, 删除敏感文件

好文置顶 关注我 收藏该文

shines77 关注 - 8 粉丝 - 101 +加关注

« 上一篇: git中如何查看一个文件的修改(更新)历史
» 下一篇: [转载]C++多线程技术

评论列表

#1楼 2014-02-24 11:19 NextLife @

2 0 推荐 反对

@whynot163volatile 的确不是原子变量,但是

\_\_sync\_bool\_compare\_and\_swap() 函数一个原子指令啊(gcc的原子指令包装函数),原子指令所使用的变量必须声明.....

--shines77

3. Re:<mvcc:annotation-driven/> 都做了那些事情  
mvc比较方便

--两点普

4. Re:一个无锁消息队列引发的血案(一) 一地:起因

volatile 关键字用得不对,volatile 可以理解为避免编译器优化,但是volatile不是原子变量。kernel.org 文档里曾经有一份 Documentation/volatile-c.....

--whynot163

5. Re:一个无锁消息队列引发的血案(四) 一月:RingQueue(上) 自旋锁

谢谢,我去试试,另外好巧,我也刚用 C++11写了dpdk的ring queue,只是还没测试速度,等弄完了也传到GITHUB和我的空间里去。我看了下你的C11实现,感觉全是full fence或者se.....

--xueyouchao

阅读排行榜

- 1. Git如何永久删除文件(包括历史记录)(19652)
2. 一个无锁消息队列引发的血案(一) 一地:起因(11948)
3. 一个无锁消息队列引发的血案(三) 一地:q3.h 与 RingBuffer(6726)
4. 认识我们的太阳系(Solar System)(6458)
5. 一个无锁消息队列引发的血案(五) 一地:RingQueue(中) 休眠的艺术(5867)

评论排行榜

- 1. 认识我们的太阳系(Solar System)(34)
2. 编译器是如何实现32位整型的常量整数除法优化的?[C/C++](30)
3. 一个无锁消息队列引发的血案(五) 一地:RingQueue(中) 休眠的艺术(27)
4. 一个无锁消息队列引发的血案(一) 一地:起因(26)
5. 一个无锁消息队列引发的血案(三) 一地:q3.h 与 RingBuffer(22)

推荐排行榜

- 1. 认识我们的太阳系(Solar System)(21)
2. 一个无锁消息队列引发的血案(五) 一地:RingQueue(中) 休眠的艺术(13)
3. 一个无锁消息队列引发的血案(一) 一地:起因(10)
4. 编译器是如何实现32位整型的常量整数除法优化的?[C/C++](7)
5. 一个无锁消息队列引发的血案(二) 一月:混合自旋锁(7)

🔒 注册用户登录后才能发表评论,请 登录 或 注册,访问网站首页。

最新订新闻:

- NASA捕获银河系超新星的“消遣幽灵”尘埃
· 特斯拉财报电话会议实录:马斯克称要降低人工成本提高效率
· 豆瓣“务实”、“知平”下沉”,它们赚钱的姿势为何如此不同?
· 万亿美元苹果 以及它走过的42年
· Google Maps分享地理位置会显示电量情况
» 更多新闻...

最新知识库文章:

- 成为一个有目标的学习者
· 历史转折中的“杭派工程师”
· 如何提高代码质量?
· 在腾讯的八年,我的职业思考
· 为什么我离开了管理岗位
» 更多知识库文章...