Sjjy.py 包含spider 类，这个类继承了threading.Thread类可以实现多线程

这个类利用改写的run()方法实现了传入一个网址的ID(完整的网址是根域名+ID)完成：

1. 对这个网址下网页内容的抓取（利用urllib2库）
2. 从抓取到的网页内容中匹配到需要的信息（利用正则式）

这个功能在infomatch.py文件中的infoMatcher类实现

1. 将匹配到的信息存入mongodb，在DB.py文件中用DB\_mongo类实现

start.py文件中的main()方法负责开启程序，程序开启需要确定5个参数：

1. Global\_urlpool 这个参数是指起始的网址ID（这个参数在globalpara.py文件里，顾名思义这个文件放全局变量，由于这是demo版，所以有一个配置变量也放里面，后期要搞一个配置文件的类，将所有配置以及临时参数，故障状态保存等加进去，提高程序容错能力）
2. slavenum 这个参数表示一次开启的线程数，默认100
3. totalnum 这个参数表示你想爬取的网页数（100的整数倍）
4. 两个线程间等待的时间：目前默认0.0005秒，这个参数的设置要靠尝试，影响这个参数的因素至少有slavenum和目前网速，等待时间太长影响性能，等待时间太短影响稳定性，所以这个参数的设置可以做一个自适应

mac.py中的类负责获取本机用户名和mac这个是为后期多机部署做准备，目前没用

至此一个单机版完成了，存在的问题就是性能与稳定性不行，容错几乎没有。

下一步要做的就是集群了，可行的方案有：

方案1：部署一个服务器（比如部署到阿里云或者租用一个），这个服务器包括两个功能：调度和存储：实时了解子机在线情况，运算能力，然后给子机安排爬取任务，并且获得子机返回的结果后进行存储（当然要有备份与滚回机制）。这个方案涉及到类似webserver架构的设计，最好自己设计一个。

方案2：弄一个局域网，在局域网中做hadoop集群开发，这就需要一个局域网，好处是可以练练hadoop开发。

方案3：离线实现。。。这个方案就是多台电脑分别负责一块网址段，比如1-10000000，所有电脑都将信息存入mongodb，最后拷贝到一起。这个方案没有难度，但也没有什么价值。

以上是一些可行的分布式计算方案

拿到所有数据后，下一步就是数据分析了，如果世纪佳缘成功了，就可以改写去爬其它类似网站，最后可以实现横向比较，为广大青年指出一条明路

这里需要从网站上挖取更多的有用信息，请大家尽情去体验一下这些网站，看看有没有有价值的东西，有了需求再去考虑把这些信息抓下来的方法，毕竟到这里架构上已经可行了。

跑Demo版的配置：

1. python2.7环境，具体配置见百度不难，注意不要安装最新的python3.3这个版本不支持很多这个程序要用的库。
2. 下载并安装第三方python库: pymongo,这个库提供了python操作mongodb的接口，这个安装也可百度
3. 安装mongodb，配置可百度，注意版本，不同版本的配置方式有点差异，用cmd命令的方式来配置，需要熟悉一些windows的cmd命令，主要就是创建一个数据库存放的文件夹，以及开启mongodb连接，并且将mongodb做成一个windows的服务，这样就不用每次要用前都去手动连接了（注意360有时候会提示关掉这个服务，请无视），这个开源的数据库在配置上不太友好也没有原生的可视化界面，但是并发还是很不错的（这点很好用，但是32位机器只支持最大4G的单个文件，所以存的时候不要超过），查询也过得去。
4. 安装mongoVUE，mongodb没有可视化界面，但是有mongoVUE这个第三方界面，这是付费软件，但是网上有一个唯一的破解版，百度可搜到，安装和配置不难。
5. 最后就是IDE的选择了，这个可以按照自己口味，我用pycharm 非常不错，毕竟是大公司开发的，同样有破解方法，你懂的。

1.学python可以用python tutorial 这个电子书，有C语言的基础，浏览一遍就行

2.要学习标准库的用法可以搜： *python标准库中文版PDF(带章节书签)*这个电子书，比较清晰，但用法Demo不够全面。

3.需要搞算法请看其他各种算法相关书，python的运行原理，python的库文件源代码（这个安装完python后在Lib这个文件夹里找，看懂这些应该就算深入理解python了。。。）还有其他等等等等等

4.分布式架构，这个不懂

5.程序成型后最好部署到liunx环境上去，这个是后话

**我们的终极目标是做一个：利用廉价硬件来实现高性能的，多并发的，强鲁棒性的，高容错性的，全自动无人值守的，日志记录完整的，易于部署的分布式网站内容爬取、存储架构！**