回测系统说明v1.0

```
from datetime import datetime
start = datetime(2008, 4, 30)    # 回测起始时间
end  = datetime(2010, 6, 1)    # 回测结束时间
universe = ['000002.sz','000001.sz','000007.sz'] # 股票池
capital_base = 100000     # 起始资金
freq = 'd' #时间单位
#refresh_rate = 5 #调用频率
#benchmark = 'SH50'     # 策略参考标准
```

上面是一些必要的信息，作为运行系统用

```
def initialize(account):     # 初始化虚拟账户状态
    account.SD = 0.1#SD是一个例子，这里是让系统在account这个实例类里帮你存储你想要的属性，这个函数只在系统初始化时调用一次
```

```
def handle_data(account):#这个函数会以初始化设定的频率调用，如上面的freq='d'（现在只实现天为单位的）则说明是每天调用一次
    ...#一些相关属性，数据在account里面
```

上面两个函数一定一定要按照给的格式写

接下来是最重要的：account的重要属性

1. account.days_counts 调用handle_data的次数，也就是第几个交易日
2. account.current_time 当前历史交易日的时间，类型为datetime,可通过account.current_time.month等查看月份啊等等
3. account.last_time类似current_time，为上次交易日的时间
4. account.tomorrow
5. account.all_universe初始化时指定的交易股票池，类型为list
6. account.universe当前交易日将all_universe中停牌，跌停等不可交易的股票剔除后剩余的股票
7. account.cash当前的现金
8. account.valid_secpos为一个字典,key为股票代码,value是对应手头上持有的该股票的数量,如果要得到所有持仓数目大于零的股票代码,可以这样写account.valid_secpos.index[account.valid_secpos>0] 这是pandas.DataFrame的写法
9. account.avgBuyprice为一个字典，key为股票代码，value是对应买该股票时的平均价格
10. account.buyPosition为一个字典，key为股票代码，value是对应最近购买该股票时的时间单位，与days_counts对应，=days_counts即为当天购买，>,<....
11. account.is_upORdownLimit(code)判断该股票是否涨，跌停
12. account.isOK2order(code)判断该股票当天是否能买卖
13. account.getTodayOpen(code)该股票的当天开盘价，float类型
14. account.getTodayHigh(code)
15. account.getTodayLow(code)
16. account.getDailyHistory(code,length)得到历史数据，length为时间长度，例如getDailyHistory('000007.sz',5)['close'].values[-1]得到前一交易日的close价格，values[-1]为最后，也就是前一交易日，如果是-2则是前两个交易日，等等。。。不过不能超过length。除了close，还有date_time,open,high,low等,就是数据库对应的表的几个属性
17. account.order(code,num,price) 下单函数,买多少手。code股票代码，num手数，>0为买入，<0为卖出，price买卖价格
18. account.order_to(code,num,price)下单函数，买或者卖到num手。code股票代码，num想让账户上持有该股票的数目
19. account.isBeginOfMonth(t)判断是否月的第一天,t为时间，类型是datetime.date
20. account.getBeginOfMonth(y,m)得到某年某月的第一个交易日,注意两个参数必须为int类型
21. account.static_profit 字典，对应股票的静态收益，如要清空，请这样:

    ```
    for i in self.all_universe:
        self.static_profit[i]=0
    ```

22. account.getIfData(type,t) 得到IF000的数据，type可为 'Open' ,'Close','High' ， 'Low' ，t为时间
23. account.holdIF 一个变量，记录当前持有的股指期货的手数

24. account.EntryIF(money)做空股指期货，money为用来做空的钱
25. account.Empty(lot)做多股指期货，lot为做多的手数

运行说明：首先将import my_strategy as st的my_strategy改为你的策略文件名

简单回测--运行my_test.py

推进回测--运行my_recursive.py

分布式推进回测：首先运行iniPP.py,再运行my_recursive_pp.py