



调度系统复现手册

- 部署好多节点树状区块链

(包括部署合约, 得到合约地址, 修改maoContract.js中变量mapContractAddress的值, 以及注意变量mapContractServer的端口, 不知道有没有用, 一致更好。同样修改trafficContract.js中内容)

- 运行 auto.js 文件可自动生成新账户

并可通过accountConfig函数的参数分配给车辆和乘客(多少个车辆账户, 多少个乘客账户), 代码逻辑简单, 使用之前最好看一看。

会将生成的账户信息写入passengerAccounts.py和vehicleAccounts.py, 以及genesis.json, 关闭geth终端。

注意auto.js里的这两个参数,

```
node1 = callfile.exec("gnome-terminal -e 'bash -c \"cd ~; cd 桌面; cd treenode1; geth --datadir ./gethdata --networkid 91036 --port
```

```
node1 = callfile.exec("gnome-terminal -e 'bash -c \"cd ~; cd 桌面; cd treenode1; geth --datadir ./gethdata --networkid 91036 --port 30303 --rpc --rpcaddr 127.0.0.1 --rpcport 8545 --rpcapi 'personal,net,eth,web3,admin' --rpcorsdomain='*' --ws --wsaddr='localhost' --wsport 8546 --wsorigins='*' --wsapi 'personal,net,eth,web3,admin' --nodiscover --allow-insecure-unlock --dev.period 1 --syncmode='full' console; exec bash\"", async function(error, stdout, stderr){
```

- 重复初始化赋予余额的操作

不用新建账户了, 也不用复制粘贴, 已经写进去了, 然后解锁所有账户, 开始挖矿

```
personal.unlockAccount(eth.accounts[?], "123456", 0)
```

- 上传地图

uploadmap_cjz.js 中, 修改myContractInstance和account(改为现有的一个账户就行)的值, 注意文件路径。终端中运行下列命令:

```
node uploadmap_cjz.js
```

注意: 开始上传前, 该区块链保持挖矿状态, 上传过程中也保持挖矿, 才可以写入区块链。

- 运行vehicle_test.py初始化车辆并等待车辆上传好初始位置和状态

vehicle_test.py 是自动化控制车辆的文件, 可以同时初始化和维护多个车辆, 可以通过workConfig函数控制车辆的工作时间, 代码逻辑简单, 使用之前最好看一看。关注vehicleAccounts.py文件, 是自动化脚本会读取的乘客信息。

- 运行passenger_test.py让乘客发出需求, 系统正式开始工作, 等待系统工作结束输出数据

passenger_test.py 是自动化控制乘客需求生成的文件, 可以通过workConfig函数控制乘客提出需求的次数和提出需求的时间间隔, 代码逻辑简单, 使用之前最好看一看。

运行自动化系统的顺序:

首先部署好多节点树状区块链→上传地图→运行auto.js自动化进行账户生成和分配→运行vehicle_test.py初始化车辆并等待车辆上传好初始位置和状态→运行passenger_test.py让乘客发出需求，系统正式开始工作，等待系统工作结束输出数据。