本文中所有的`账号地址列表`为以下形式的文件：

```
nch17djntj9jj3kqvyl8hfdnae8ayuutwkqaag47z7
nch1h5tae6kfms3xvwtqsfu3y57hd7jau3j4eq6uk5
nch12f5knqfa5v8a8elajmpkm972tvu5532fdlqxuu
nch1jys3k2g02z9r4hh84k456q7wxyjd3eglrvlawu
```

批量添加账号

```
# bash batch_add_account.sh 账号名字前缀 账号数量 账号密码
# 创建[acc1,acc2,....,acc10]10个账号，密码是11111111
bash batch_add_account.sh acc 10 11111111
```

批量转账

```
# bash batch_send.sh 账号地址列表 转出账号 转出账号密码 金额 转账间隔单位秒
# 向`账号地址列表`文件中的每个地址转账10000000000000000pnch，从sky账号转出，密码是11111111，每6秒转账1次
bash batch_send.sh mac_accs sky 11111111 10000000000000000pnch 6
```

快速批量转账

- 必须使用netcloth的speedup_nchcli_sign_txs分支的代码编译出的nchcli，该分支的nchcli支持并发的读取账号用于签名
- 建立在`批量转账`基础上，需要先有一批有token的n个账号(这里称为`银行账号`)，转账速度提升n倍，受限于cpu的性能，建议n=cpu线程数量*4左右

```
# bash quick_batch_send.sh 银行账号列表 账号地址列表 密码 金额 每批转账间隔单位秒
bash quick_batch_send.sh bankers n1_accs 11111111 500000000000000pnch 8
```

模拟大量并发的转账操作

1.生成转账表

假设有n1,n2,n3,n4节点，每个节点上都已经创建了1000个账号，他们之间互相转账

```
# bash batch_gen_send_list.sh [地址表1,地址表2,...,地址表n]
# n1-->n2,n2-->n3,n3-->n4,n4-->n1形成4张转账表n1_accs_to_n2_accs,n2_accs_to_n3_accs,n3_accs_to_n4_accs,n4_accs_to_n1_accs,
bash batch_gen_send_list.sh n1_accs n2_accs n3_accs n4_accs
```

2.转账

```
# bash send_by_send_book.sh 转账表 金额 密码 第几轮 总共几轮
# 通过转账表n1_accs_to_n2_accs转账5000pnch，总共1轮，当前只是第1轮
bash send_by_send_book.sh n1_accs_to_n2_accs 5000pnch 11111111 1 3

# 如果要根据转账表转账多轮
# bash n_batch.sh 轮数 转账表
# 对转账表执行10轮转账
bash n_batch.sh 10 n1_accs_to_n2_accs
```

批量查询账号

查询账号是否存在，如果不存在输出到account_not_exist

```
# bash batch_q_account.sh 账号地址列表
# 查询mac_accs账号列表中的账号是否存在，如果不存在相应的地址会存储在文件account_not_exist
bash batch_q_account.sh mac_accs
```

批量调用合约

有了快速转账之后就可以有大量有效的账号用于调用合约

创建合约

```
echo 11111111 | nchcli vm create --from $(nchcli keys show -a sky) --amount=0pnch --code_file ./payment/pay.bc --abi_file ./payment/pay.abi  --args="100" -y --gas 5000000
```

批量调用

```
# bash n_batch_call_contract.sh 合约地址 账号地址列表 密码 轮数 调用合约间隔单位秒
# mac_accs为每个账号调用合约nch1telsavxyay67y6d64tu5tce67t3exu8ng52x38 10次，调用间隔0.03，间隔根据机器的cpu性能来配置，签名非常消耗cpu
bash n_batch_call_contract.sh nch1telsavxyay67y6d64tu5tce67t3exu8ng52x38 mac_accs 11111111 10 0.03
```

批量对账

用于调用合约，转账前后余额变化的对账

调用合约的所有交易hash记录下来

```
bash batch_call_contract.sh nch1f60rlguj00ymdvdt2wsxxl7y27d486g995pugr |grep txhash |awk -F '"' '{print $4}' > txs
```

为了方便对账，假设所有账号创建后的余额都是5000nch，然后全部调用一次合约在进行对账

```
for txh in `cat txs`
do

gas=`nchcli q tx $txh | grep gas_used | awk -F '"' '{print $4}'`
fee=$((gas*1000))
acc=`nchcli q tx $txh | grep from | awk -F '"' '{print $4}'`
amount=`nchcli q account $acc | grep amount | awk -F '"' '{print $4}'`
total=$((amount+fee))
echo $txh:$acc:$total:$amount:$fee

done
```

批量拷贝脚本

用于升级程序等
为了方便使用需要在拷贝对象机器上配置ssh key

```
# bash batch_scp.sh 机器列表 待拷贝文件 目标机器绝对路径
bash batch_scp.sh m_list ~/go/bin/nchd /root/go/bin
```

- m_list如下:

```
n2
n3
n4
```

- m_list对应的ssh配置~/.ssh/config

```
Host n2
User root
Hostname 121.196.14.103

Host n3
User root
Hostname 121.196.121.217

Host n4
User root
Hostname 47.110.9.3
```

批量重置节点(nchd unsafe-reset-all)

```
bash reset.sh
```

批量检查nch版本

```
bash check_nch_version.sh
```