

## 关于网络的 project

1 DDL 为 18 年 1 月 7 号晚上 12 点，强 DDL，过期不候。

2 3 到 5 人组队，共同完成。

2 提交一个压缩文件夹给我，文件夹里边包含源码，可执行码，第三方组件和文档（具体要求与 Socket 提交要求编程一致，见附 3）文件夹和文档命名为班级-小组名字(英文)

3 文档要求英文书写，不统一给出模板，但至少应该包括（1）小组内部成员名字(英文)，所负责的工作和邮箱（推荐学校邮箱）（2）编译代码的编译选项和编译说明（3）所实现协议的简单说明分析（4）协议测试结果，格式为 pdf

附：

1 可选的工作(具体可见 ppt)

- Implement crash-tolerant consensus

- Make it terminate

- Deal with message loss

- Implement state machine replication

—bank with accounts?

- Implement gossip

- Implement byzantine-tolerant consensus

- Implement byzantine state machine replication

2Robbert 上课的代码（有 bug）

(1)Python 源代码：

```
import sys, socket, pickle
```

```
def toAddr(s):
```

```
    parts = s.split(":")
```

```
    return parts[0], int(parts[1])
```

```
estimate = sys.argv[1]
```

```
me = int(sys.argv[2])
```

```
nodes = map(toAddr, sys.argv[3:])
```

```
print "the nodes are %s" % nodes
```

```
skt = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

```
skt.bind(nodes[me])
```

```
round = 0
```

```
while True:
```

```
    for node in nodes:
```

```
        msg = (me, round, estimate)
```

```
        skt.sendto(pickle.dumps(msg), node)
```

```
    recvd = [None] * len(nodes)
```

```
    while True:
```

```
        data, addr = skt.recvfrom(4096)
```

```
(src, r, e) = pickle.loads(data)
print "got a message! (%s, %s %s)" % (src, r, e)
```

```
if r < round:
    skt.sendto(pickle.dumps(msg), addr)
    continue
```

```
if r > round:
    round, estimate = r, e
    break
```

```
# assert(r == round)
recvd[src] = e
if not filter(lambda x: x == None, recvd):
    break
```

```
round = round + 1
```

(2)

Shell script:

```
estimate=$1
```

```
me=$2
```

```
python consensus.py $estimate $me 127.0.0.1:5000 127.0.0.1:5001
127.0.0.1:5002
```

(3) socket 编程的提交要求:

1. **【源码】** 存放源码
2. **【可执行码】** 存放编译后的 release 版本
3. **【第三方组件】** 存放编译源码和运行可执行码需要的所有第三方组件（开发平台自带的需要）
4. **【文档】** 说明开发平台（包括具体的版本）、编译步骤（确保能够成功编译并运行）、运行说明（确保使用可执行码顺利运行）