

1、 Your design of RPC

首先，我使用了自己定义的消息类，在 message 包中，里面有抽象类 AbstractMessage 和它的 4 个子类。

这些消息用来传递需要的数据和用来表明自己是哪一种消息的 type，不同的 type 会有不同的处理。

所以，整个 RPC 的过程就是，在 client 里会调用 sdfs，sdfs 会调用 NameNodeStub，这里面会制作不同的消息，通过 socket 传递给 NameNodeServer，然后根据 type 处理。NameNodeServer，会调用 NameNode 去真正处理文件树，这一步和上一个 lab 一样，还要制作相应的返回消息和返回 channel。

接收到需要的 channel 的 sdfs 会接着调用 channel，channel 会调用 DataNodeStub 发送读写的消息和数据，也是同样通过我自己定义的消息，DataNodeServer 接收到之后处理，并调用 DataNode 进行数据读写。

我目前只完成了 put、get 和 mkdir 的命令的实现，另外的三个命令还没有测试。

另外，目前 IP 地址是写死在程序中的，为了便于本机测试，一律都是 127.0.0.1。

NameNodeServer 监听的端口是 4341，DataNodeServer 监听的端口是 4342。

2、 Describe the problem you met during developing and how you solve it

最大的困难是复杂的流程让我 debug 很困难，我只能一步一步调试或者 print 来查看问题出在哪里。

然后就是由于不再存储 inode，没有办法直接查看 dirnode 的内容，debug 很麻烦。

3、 Describe the change of the architecture of yours(if possible)

与之前相比，整个项目变化最大的地方是少了 inode 的读写和更多的层

次。

由于可以直接存入对象，我在存储和恢复文件树的时候方便了很多。

为了实现 RPC，必须加入 stub，并用 socket 进行通信，所有要定义发送方，接收方和中间的消息。

4、 Extra work you have done

自定义了很多消息，便于通信。

在要求的 6 个命令之上还会添加 ls 命令，用来列出目录结构。