**大作业四选一题目及具体要求**

**一、GeekGFS**

目标：

1. 实现一个简化的GFS分布式文件系统，GeekGFS
2. 具有客户端client、元数据服务器master和多个数据服务器chunkserver三个基本组件
3. 客户端client需要实现read、write、append、exist和delete函数功能。函数实现要求：从master获取元数据（包含chunk ID和chunk位置），更新master上的元数据，最后与chunkserver进行实际的数据传输。
4. GFSMaster模拟GFS元数据服务器，在内存中存储所有的元数据，且需要周期备份到硬盘。
5. GFSChunkserver实现分块存储，块固定为64KB。为简化起见，可使用同一个节点的多个路径模拟多个物理节点。
6. 编写测试程序，进行测试验证。
7. 开发语言不限。

备注：可不考虑元数据加锁、chunk租约、复制，master故障转移、chunkserver心跳、垃圾回收等复杂机制。

**二、Bittorrent客户端**

目标：

1. 实现一个简易的Bittorrent客户端
2. 实现对.torrent文件的解析
3. 实现与Tracker交互获得peer的IP和端口
4. 实现从peer获取pieces，可不实现unchoking激励机制，piece选择策略可选用简单的随机或者顺序策略
5. 编写测试程序，进行测试验证

**三、YA-RPC: Yet-Another RPC Framework**

目标：

1. 编写一个简易RPC框架，YA-RPC
2. 支持基本数据类型：int, float, string
3. 支持 At-least-once 语义
4. 使用YA-RPC编写一个demo程序，实现如下API：
   1. 远程调用 float sum(float a, float b)
   2. 远程调用 string uppercase(str)
   3. 不少于2个客户端，1个服务端
5. 开发语言不限

**四、Chandy-Lamport 分布式快照**

目标：

1. 在一个 tokens 交易系统上，使用 Clandy-Lamport算法构建快照
2. 通过 Github 代码仓库获取代码（项目地址：https://github.com/zqyi/Chandy-Lamport-Snapshot），代码中 TODO: IMPLEMENT ME 是需要实现的函数
3. 实现完整的全局快照功能
4. Server 需要在每次快照完成时通知 Simulator
5. 不同轮次的快照间不能相互影响
6. 通过 snapshot\_test.go 中的测试，运行 go run 即可进行测试
7. 开发语言为 Go 语言，通过[Go指南](https://tour.go-zh.org/list)（https://tour.go-zh.org/list）学习编程