

Lab2: NFS

实验目标与背景

实验目标

本实验旨在帮助学生理解分布式文件系统的核心概念，通过实现一个类似 NFS（网络文件系统）的服务机制，掌握客户端-服务器模型、文件缓存和分布式锁等技术。学生需要完成以下目标：

- 实现一个 NFS 服务器，提供文件操作接口。
- 实现多个客户端，通过接口访问服务器文件。
- 使用本地文件缓存优化客户端操作。
- 利用 Zookeeper 实现分布式锁机制。

NFS 背景

NFS（Network File System，网络文件系统）是一种分布式文件系统协议，允许客户端通过网络访问远程服务器上的文件，如同操作本地文件系统。NFS 基于客户端-服务器模型，服务器提供文件存储和访问服务，客户端通过挂载远程文件系统进行操作。NFS 的核心特点包括：

- 透明的文件访问：客户端无需关心文件位置。
- 缓存机制：客户端缓存文件数据以提高性能。
- 锁机制：协调多客户端并发访问。

本实验要求学生实现一个简化版的 NFS 服务，重点关注文件操作接口、本地缓存和分布式锁机制。

实验任务

任务 1：实现 NFS 服务机制

实现一个类似 NFS 的服务机制，包含一个 NFS 服务器（单个进程），提供文件服务。服务器需支持以下接口：

- open：打开指定文件并返回其内容。
- read：读取文件中指定数据。
- write：向文件中写入数据。
- close：关闭文件并保存更改。

客户端通过这些接口访问 NFS 服务器上的文件。

要求：

- 服务器使用单个进程，通过 TCP 套接字处理客户端请求。
- 文件存储在服务器指定目录（例如 `./server_files`）。

任务 2：实现多个客户端

实现至少两个客户端，通过上述接口读写 NFS 服务器管理的文件。

要求：

- 每个客户端能够调用 `open`、`read`、`write`、`close` 接口。
- 运行至少两个客户端实例，验证并发访问功能。

任务 3：客户端本地文件缓冲

客户端在调用 `open` 时，将文件从服务器下载到本地目录，保存为临时文件名，同时记录该文件在 NFS 服务器上的原始名称。客户端的 `read` 和 `write` 操作仅对本地缓冲的临时文件进行操作。

要求：

- 确保临时文件名唯一，记录服务器文件名与临时文件的映射。
- 所有读写操作基于本地临时文件。

任务 4：Zookeeper 锁机制

安装 Zookeeper 服务器并创建一个节点 `/nfslk`。客户端在调用 `open` 操作之前，利用 `/nfslk` 节点实现锁机制（需自行设计）。

要求：

- 设计锁机制。
- 确保只有获取锁的客户端可以打开文件。
- 在 `close` 时释放锁。

任务 5：写操作与关闭

对于 `write` 操作，所有更改仅保存在本地临时文件，不上传到 NFS 服务器。当客户端调用 `close` 操作时，本地缓冲文件被上传到服务器，覆盖服务器上的原文件。

要求：

- 确保 `write` 操作仅修改本地文件。
- `close` 时上传内容完整覆盖服务器文件。

任务 6：读操作

当另一个客户端打开文件执行 read 操作时，文件数据被下载到本地作为临时文件。客户端的 read 操作（例如读取 10 个字节）直接从本地临时文件读取内容，无需访问服务器。

要求：

- 确保 read 操作高效，基于本地临时文件。
 - 支持任意读取大小（如 10 字节）。
-

提交要求

- **代码：**
 - 提交服务器代码文件（`nfs_server.py`）。
 - 提交客户端代码文件（`nfs_client.py`）。
 - **实验报告：**
 - 提交 PDF 格式的实验报告，包含以下内容：
 - **设计思路：**描述服务器、客户端、锁机制的设计。
 - **实现细节：**说明关键实现步骤和代码结构。
 - **测试结果：**提供多客户端并发访问的日志或截图，证明功能正确。
 - **提交格式：**
 - 将代码和报告压缩为 `学号_姓名_Lab2.zip`。
 - **截止时间：**2025年5月27日 23:59。
-

评分标准

总分 100 分，评分细则如下：

- **NFS 服务器（30 分）：**
 - 正确实现 `open`、`read`、`write`、`close` 接口（15 分）。
 - 稳定处理多客户端并发请求（15 分）。
- **客户端（30 分）：**
 - 正确实现客户端接口和本地缓存（15 分）。
 - 实现至少两个客户端并验证并发访问（15 分）。
- **Zookeeper 锁机制（20 分）：**
 - 正确配置 Zookeeper 和 `/nfslk` 节点（10 分）。
 - 实现分布式锁并正确加锁/解锁（10 分）。

- **实验报告（20 分）：**
 - 清晰的设计说明和实现细节（10 分）。
 - 完整的测试结果和问题分析（10 分）。
-

关于迟交

对于逾期提交的实验，成绩将按照每天扣除总分的 10% 进行递减，直至分数为零。例如，迟交 1 天，成绩为原分数的 90%；迟交 2 天，成绩为原分数的 80%，依此类推。