《计算机系统工程导论》课后作业

第 4 章 模块化　　作业题目：简单文件系统

作业要求：选择自己认为正确的选项，并进行简要分析。　　仅打印答题页即可。

Moon 公司在其最新的项目中实现了一个简化版的网络文件系统（Network File System，NFS）：称之为文件系统服务（File System Daemon，FSD）。

一家科技公司的开发团队想为他们的最新产品开发文件系统的网络支持，他们决定采用 Moon 公司的 FSD 技术，以便在分布式环境中方便地管理文件。

服务器端的工程师们开始编写 FSD 服务器，以响应各种 UNIX 文件系统调用，其必须确保 FSD 能够正确地处理读取（read）、写入（write）、打开（open）、关闭（close）、创建（create）等操作。通过简化这些操作，他们能够构建一个高效且可靠的网络文件系统。

与此同时，客户端的开发人员也在着手开发 FSD 的客户端库。这个库负责接收来自应用程序的 UNIX 文件系统调用，并将其转换为与 FSD 服务器通信的请求。他们需要确保客户端库能够正确地处理各种情况，例如网络中断或服务器故障。

整个开发团队努力工作，期待着将这个新的文件系统服务集成到他们的产品中，为用户带来更好的体验，具体伪代码如图所示。

```
// Map FSD handles to host names, remote handles:
string handle_to_host_table[1000]                    // initialized to UNUSED
integer handle_to_rhandle_table[1000]                // handle translation table

procedure FSD_OPEN (string name, integer mode)
    integer handle ← FIND_UNUSED_HANDLE ()
    if name begins with "/fsd/" then
        host ← EXTRACT_HOST_NAME (name)
        filename ← EXTRACT_REMOTE_FILENAME (name)    // returns remote file handle or ERROR
        rhandle ← RPC (host, "OPEN", filename, mode)
    else
        host ← ""
        rhandle ← OPEN (name, mode)
    if rhandle ← ERROR then return ERROR
    handle_to_rhandle_table[handle] ← rhandle
    handle_to_host_table[handle] ← host
    return handle

procedure FSD_READ (integer handle, string buffer, integer nbytes)
    host ← handle_to_host_table[handle]
    rhandle ← handle_to_rhandle_table[handle]
    if host ← "" then return READ (rhandle, buffer, nbytes)
    // The following call sets "result" to the return value from
    // the read(...) on the remote host, and copies data read into buffer:
    result, buffer ← RPC (host, "READ", rhandle, nbytes)
    return result

procedure FSD_CLOSE (integer handle)
    host ← handle_to_host_table[handle]
    rhandle ← handle_to_rhandle_table[handle]
    handle_to_rhandle_table[handle] ← UNUSED
    if host ← "" then return CLOSE (rhandle)
    else return RPC (host, "CLOSE", rhandle)
```

其中：

- 非本地文件，使用“/fsd/hostname/apath”
- fsd_open("/fsd/cse.ouc.edu/foobar", read_only)
- rpc("/fsd/cse.ouc.edu/foobar", "open", "/foobar", read_only).
- open("/foobar", read_only)
- 在服务器端执行时，返回文件描述符
- 采用精确一次（exactly-once）语义，无缓存

问题 1：“”的用处是什么？

A unused entry **B** open file on client machine

C eof **D** error

问题 2：小红同学建议我们去掉 handle_to_rhandle_table，直接返回由 open 函数获得的文件句柄，并对其进行测试。以下哪个测试程序不受影响？

A 读单个、本地文件 **B** 读单个、远程文件

C 读写多个本地文件 **D** 读写单个远程服务器多个文件

E 读多个远程服务器多个文件

F 读多个本地文件+单远程服务器多文件

问题 3：完成下表

Statement	NFS	FSD
Remote handles include inode numbers	Yes/No	Yes/No
Read and write calls are idempotent	Yes/No	Yes/No
Can continue reading an open file after deletion (e.g., by program on remote host)	Yes/No	Yes/No
Requires mounting remote file systems prior to use	Yes/No	Yes/No

问题 4：FSD 的第二个版本将“精确一次”（exactly-once）保证替换为“至少一次”（at-least-once），但偶尔会导致 fsd_read 出现错误。在跟踪分析后，设计者考虑了以下机制：客户端增加响应缓存以应对重复查询，服务器增加响应缓存以应对重复查询，并且在 RPC 中加入单调递增的序列号（nonce），以确保相同类型的请求产生差异。问：你建议用什么方式来解决问题？

A. 每个客户端上的响应缓存，用于应对重复查询；

B. 服务器上的响应缓存，同样用于应对重复查询；

C. 在 RPC 中引入单调递增的序列号（nonce）；

A+C；

B+C；

A+B；

答题页

第 4 章 模块化 作业：简单文件系统

姓名_____ 学号_____