

Project 2 文件权限查询设计

题目

描述

要求

输入输出

过程示例

案例

Tips

提交

知识背景

Project 2 文件权限查询设计

题目

描述

给定一个文件树、一个用户树、一个用户_文件权限列表；

如果用户的父节点有对某文件的操作权限，那么用户有对该文件的操作权限有效；

如果用户的父节点没有对某文件的操作权限，那么用户有对该文件的操作权限无效；

如果用户有对某文件的父文件的操作权限，那么用户有对该文件的操作权限有效；

如果用户没有对某文件的父文件的操作权限，那么用户有对该文件的操作权限无效；

如果无法根据用户_文件权限列表查询到结果，那么默认用户对该文件操作权限无效；

要求

- 权限有三种，分别是 0、1、2
 - 0 表示没有任何权限
 - 1 表示只读权限
 - 2 表示有写权限
- 每个文件或文件夹生成一个唯一的 id；
- 每个用户生成一个唯一的 id；
- id (unsigned int) 均按照创建顺序赋值；
- 文件树 `FILE_root` 及其结构 `file_tree.txt`、用文件模拟的用户树 `USER_root` 及其结构 `user_tree.txt`、权限列表 `user_permission_file.xlsx` 已经提供，需要使用树结构还原文件和用户。

输入输出

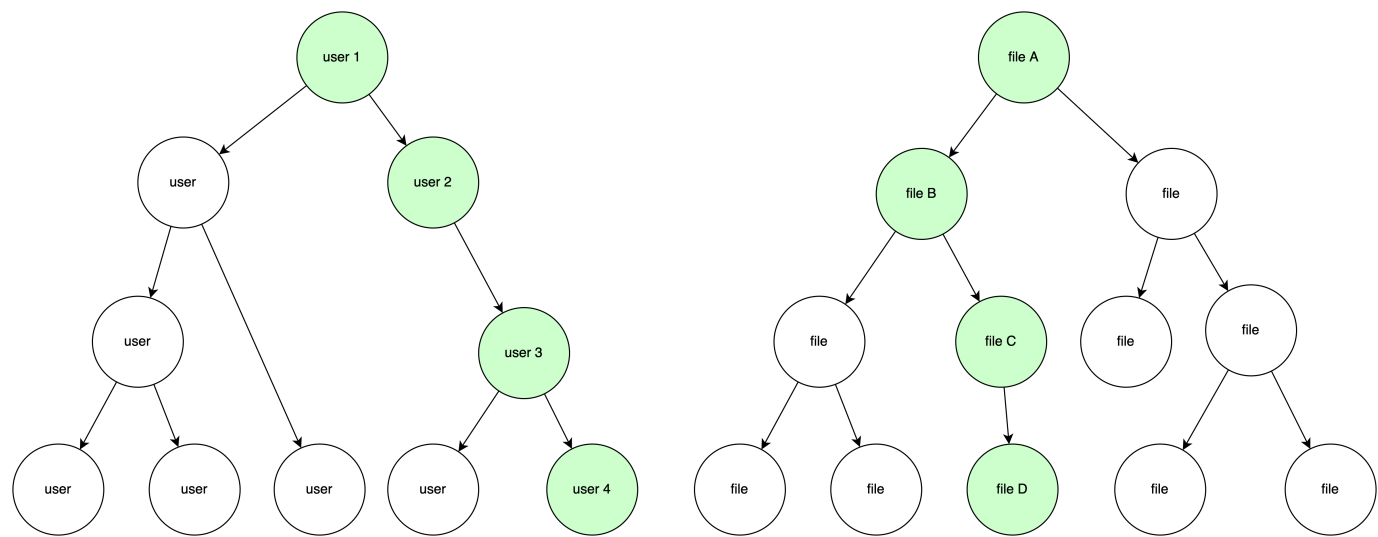
输入

- 一个用户 `id`，一个文件 `id`

输出

- 用户对该文件的权限值（0 或者 1 或者 2）

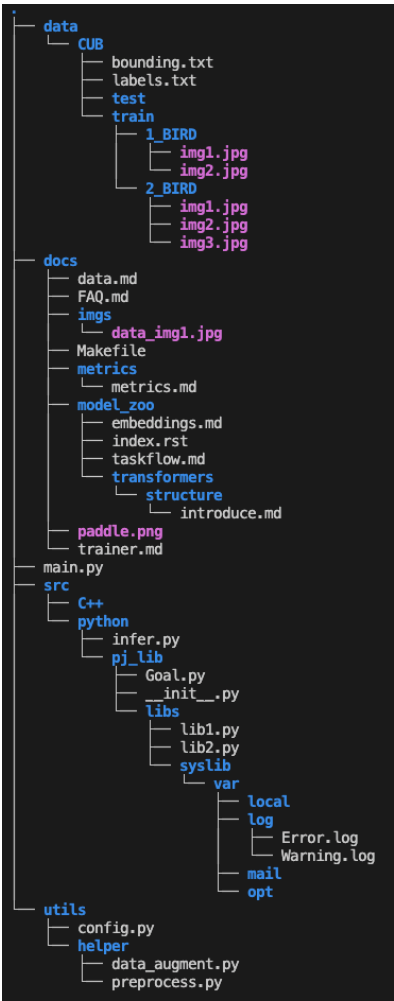
过程示例



查询 `user 4` 对 `file D` 是否有读权限

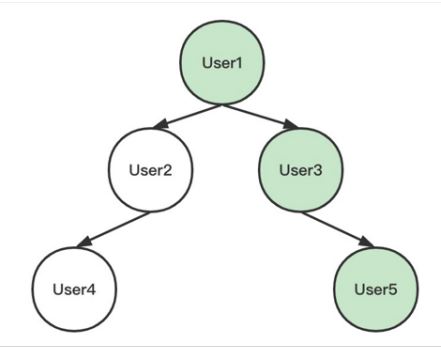
- 首先判断 `user 1` 对 `file A`
- 如果没有，则直接判定 `user 4` 对 `file D` 无读权限
- 如果有，则再判断 `user 2` 对 `file A` 的权限
 - 如果没有，则直接判定 `user 4` 对 `file D` 无读权限
 - 如果有，则再判断 `user 3` 对 `file A` 的权限
- 依次查询，直到得到 `user 4` 对 `file A` 有读权限
- 则再判断 `user 4` 对 `file B` 的权限
 - 如果没有，则直接判定 `user 4` 对 `file D` 无读权限
 - 如果有，则重复前面 `user 4` 对 `file A` 的过程判断 `user 4` 对 `file B` 的权限
- 直到最终获取到 `user 4` 对 `file D` 的权限

案例



这是以 DS_Test 文件夹为根目录的文件树结构，蓝色表示目录，白色和粉色表示具体文件。

现在要查询 User5 用户对文件 DS_Test/src/python/pj_lib/Goal.py 的权限，由用户权限表和树状用户结构可知，User5 对该文件的权限受到 User3 和 User1 的限制，User3 对目标文件的访问权限为 0，因此 User5 对该文件的权限为 0（不可读）。



树状用户结构

USER	FILE	PERMISSION
User1	DS_Test/src/python	2
...
User3	DS_Test/src/python/	0
User3	DS_Test/src/python/pj_lib	2
...
User5	DS_Test/src/python/pj_lib/Goal.py	1
...

用户权限表

Tips

★ 对pj有任何疑问可以直接在[腾讯文档](#)中编辑，助教会尽快回复。

提交

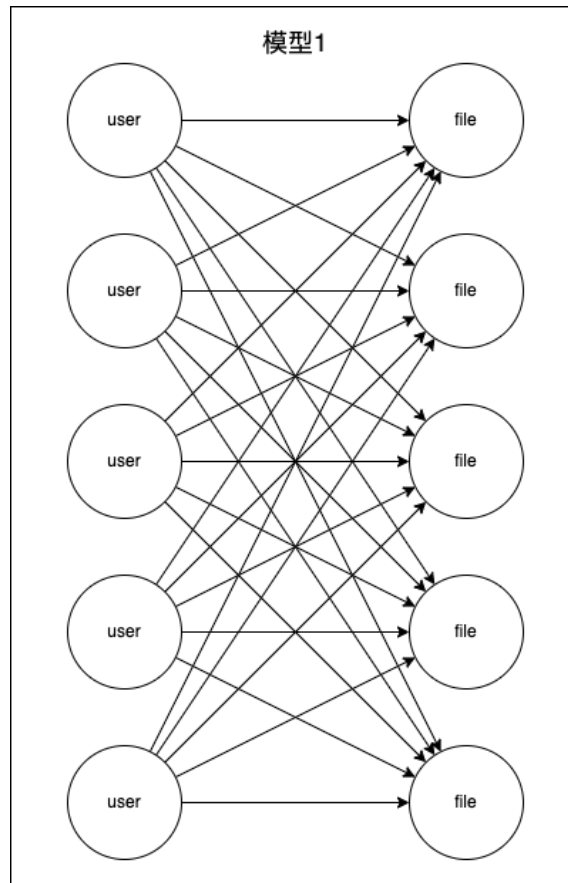
提交到 elearning 上

提交一个压缩包，命名为 `<学号>-pj-1.zip`，如 `21012345678-pj-1.zip`

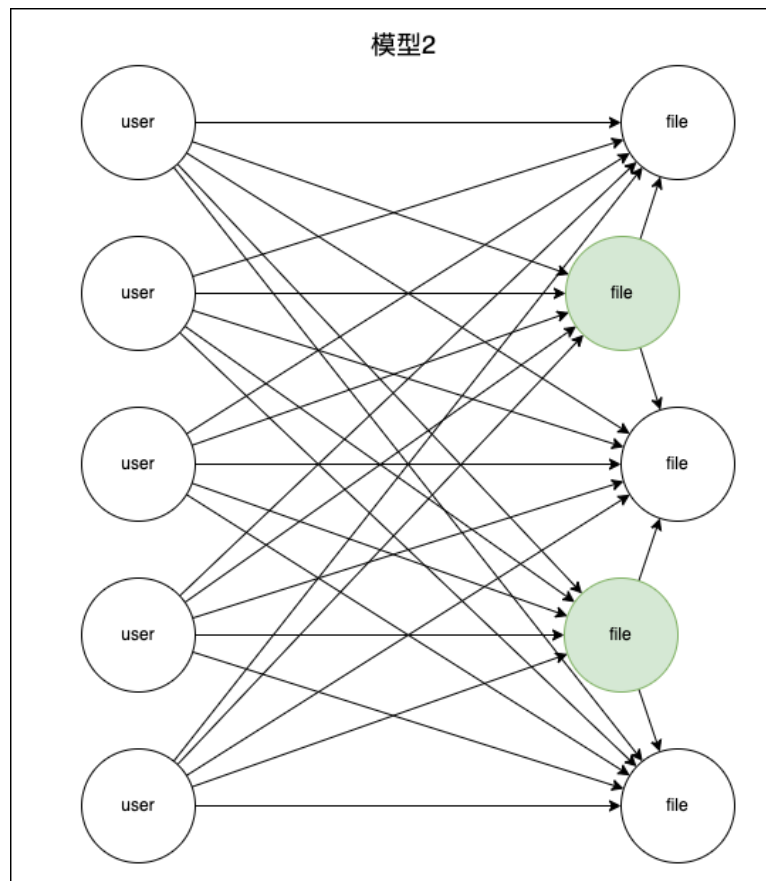
- 压缩包包括：
 - 所有代码
 - 请确保代码可以正常编译运行
 - 请写下详细注释
 - 报告文档，pdf 格式
 - 实验环境，如编译器什么版本等
 - 使用的数据结构和简单的复杂度分析

知识背景

- 有一个文件系统，和一些用户，我们想要设置指定用户对指定文件的权限
- 程序员设计了模型 1(ACL 模型)，保存所有用户对所有文件的权限。
 - 模型 1 的缺点是当用户或文件数量多时，需要保存的信息会有很多条。如有 100 个用户和 1000 个文件，那么我们需要保存 $100 * 1000 = 100000$ 条权限信息。



- 为了提高文件安全性，程序员设计了模型 2。文件依靠树结构传递权限信息，子文件访问受到父文件权限影响。
 - 模型 2 仍有一些缺点，系统需要记录所有用户对文件的权限。



- 为了方便管理用户，程序员设计了模型 3（RBAC模型）。相同权限或者同级的用户构成用户组，系统记录用户组对文件夹的访问权限，这样很大程度上减少了权限存储的开销。

