操作系统实验报告

学院：计算机科学与技术学院 专业：计算机科学与技术 班级：计科2103

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **姓 名** | |  | **学 号** |  | **成绩** |  |
| **实验项目名称** | |  | **实验时间** |  | | |
| **教 师 评 语** | 评价指标：   * 题目内容和要求完成情况 优 □ 良 □ 中 □ 差 □ * 对算法原理的理解程度 优 □ 良 □ 中 □ 差 □ * 程序设计水平 优 □ 良 □ 中 □ 差 □ * 程序运行效果及正确性 优 □ 良 □ 中 □ 差 □ * 实验报告结构清晰 优 □ 良 □ 中 □ 差 □ * 报告中总结和分析详尽 优 □ 良 □ 中 □ 差 □ | | | | | |
| **实验目的** |  通过设计简单的文件系统，掌握文件系统的基本操作和实现过程。   理解文件系统的目录管理、文件操作（创建、删除、打开、关闭）和用户权限控制。   模拟二级文件系统，熟悉文件的物理结构与逻辑结构的映射关系。   掌握C++文件操作和系统功能调用的基本实现方法。 | | | | | |
| **实验**  **内容及要求** | 1、实现功能：  用户登录：用户通过用户名和密码登录系统。  文件操作：支持文件创建、删除、打开和关闭。  目录管理：支持在用户目录下创建、列出文件。  2、实验要求：  支持用户登录后操作，非登录状态无法执行文件系统操作。  文件创建时，存储在当前用户的活动目录中，并在操作系统桌面中可见。  打开文件后支持查看文件内容，关闭文件后需要将其状态重置 | | | | | |
| **实验环境** | 硬件环境：   CPU：Intel Core i5   内存：8GB 及以上   存储：500GB 可用磁盘空间  软件环境：windows操作系统   操作系统：Windows 10 / Linux   开发工具： DEVC，g++ 编译器   支持 C++17 标准及以上版本 | | | | | |
| **总体设计** |  **设计思想**：   * 文件系统以用户目录为基础，每个用户拥有独立的文件存储空间。 * 使用二级目录结构：主目录（用户）下包含子目录（活动目录），每个子目录管理多个文件。 * 文件状态（如打开、关闭）由内存中的数据结构管理，文件内容通过磁盘文件存储。 * 支持权限控制，确保未登录用户无法访问文件系统。    **数据结构**：   * 文件结构： 包含文件名、物理路径、文件保护码、文件长度、打开状态。 * 子目录结构： 子目录中包含文件列表。 * 主目录结构： 主目录管理多个用户及其子目录。    **模块划分**：   * 登录模块：用户身份验证及初始化。 * 文件操作模块：实现文件的创建、删除、打开、关闭功能。 * 目录管理模块：支持列出当前目录文件信息。 * 交互界面模块：通过菜单实现用户命令选择。 | | | | | |
| **程序流程图** |  | | | | | |
| **主要源代码及注释** | **1. 数据结构**  // 文件属性结构  struct File {  string name; // 文件名  string physicalPath; // 文件的实际路径  string protection; // 保护码（读写权限）  int length; // 文件长度  bool isOpen; // 文件是否打开  };  // 子目录结构  struct Directory {  string dirName; // 目录名称  map<string, File> files; // 文件列表  };  // 主目录结构  struct MainDirectory {  map<string, Directory> subDirs; // 用户的所有子目录  string currentDir; // 当前活动目录  };  **2. 登录系统**  void login() {  string username, password;  cout << "请输入用户名：";  cin >> username;  cout << "请输入密码：";  cin >> password;  if (username == "admin" && password == "1234") { // 简单验证  currentUser = username;  isLoggedIn = true;  initUserFolder(currentUser); // 初始化用户目录  cout << "登录成功！" << endl;  } else {  cout << "用户名或密码错误！" << endl;  }  }  **3. 创建文件**  void create() {  if (!isLoggedIn) {  cout << "请先登录系统！" << endl;  return;  }  string fileName;  cout << "请输入文件名：";  cin >> fileName;  Directory &directory = fileSystem.subDirs[fileSystem.currentDir];  if (directory.files.find(fileName) != directory.files.end()) {  cout << "文件已存在！" << endl;  return;  }  File newFile;  newFile.name = fileName;  newFile.physicalPath = "./" + currentUser + "/" + fileName; // 文件路径  newFile.protection = "RW";  newFile.isOpen = false;  ofstream outFile(newFile.physicalPath); // 创建物理文件  if (outFile.is\_open()) {  outFile << ""; // 写入空内容  outFile.close();  } else {  cout << "文件创建失败！" << endl;  return;  }  directory.files[fileName] = newFile;  cout << "文件 " << fileName << " 创建成功，路径：" << newFile.physicalPath << endl;  }  **4. 打开文件**  void open() {  if (!isLoggedIn) {  cout << "请先登录系统！" << endl;  return;  }  string fileName;  cout << "请输入要打开的文件名：";  cin >> fileName;  Directory &directory = fileSystem.subDirs[fileSystem.currentDir];  if (directory.files.find(fileName) == directory.files.end()) {  cout << "文件不存在！" << endl;  return;  }  File &file = directory.files[fileName];  if (file.isOpen) {  cout << "文件已处于打开状态！" << endl;  return;  }  file.isOpen = true;  cout << "文件 " << fileName << " 已打开，路径：" << file.physicalPath << endl;  thread([](const string &path) {  system(("notepad " + path).c\_str());  }, file.physicalPath).detach(); // 非阻塞打开文件  }  **5. 关闭文件**  void close() {  if (!isLoggedIn) {  cout << "请先登录系统！" << endl;  return;  }  string fileName;  cout << "请输入要关闭的文件名：";  cin >> fileName;  Directory &directory = fileSystem.subDirs[fileSystem.currentDir];  if (directory.files.find(fileName) == directory.files.end()) {  cout << "文件不存在！" << endl;  return;  }  File &file = directory.files[fileName];  if (!file.isOpen) {  cout << "文件未处于打开状态！" << endl;  return;  }  file.isOpen = false;  cout << "文件 " << fileName << " 已关闭！" << endl;  } | | | | | |
| **程序测试及结果分析** | 1、登录测试：  输入正确用户名和密码，系统提示登录成功。    输入错误用户名或密码，系统提示登录失败。    2、文件操作测试：  需要用户登入后才能操作：    创建文件后，文件成功出现在用户目录中。    打开文件，文件内容在记事本中显示。    关闭文件后，文件状态重置为关闭。    **测试结果**  所有测试用例均成功运行，系统按预期完成各项功能。 | | | | | |
| **实验总结** | 本实验通过 C++ 实现了一个简单的二级文件系统，掌握了文件管理的核心功能。  理解了文件系统的基本组成结构，包括目录、文件和用户权限。  实现了非阻塞的文件打开操作，提升了用户体验。  实验可以进一步扩展，例如添加多用户管理、文件内容读写和更复杂的权限控制。 | | | | | |