**姓名：冯子潇**

**专业：计算机科学与技术**

**邮箱：**[**289080103@qq.com**](mailto:289080103@qq.com)

**Github仓库：**[**wuliwuyi/project-experiences**](https://github.com/wuliwuyi/project-experiences)

**个 人 作 品**

**目 录**

[**项目一 C语言程序设计实验** 3](#_Toc194968466)

[**1.1** **项目概述** 3](#_Toc194968467)

[**1.2 主要工作内容** 3](#_Toc194968468)

[**1.3 项目结果** 4](#_Toc194968469)

[**1.4 收获与体会** 4](#_Toc194968470)

[**项目二 电子商务系统分析与设计** 6](#_Toc194968471)

[**2.1项目概述** 6](#_Toc194968472)

[**2.2 主要工作内容** 6](#_Toc194968473)

[**1.3 项目结果** 7](#_Toc194968474)

[**1.4 收获与体会** 7](#_Toc194968475)

[**项目二 电子商务系统分析与设计** 8](#_Toc194968476)

[**2.1项目概述** 8](#_Toc194968477)

[**2.2 主要工作内容** 8](#_Toc194968478)

[**1.3 项目结果** 9](#_Toc194968479)

[**1.4 收获与体会** 9](#_Toc194968480)

**项目一 C语言程序设计实验**

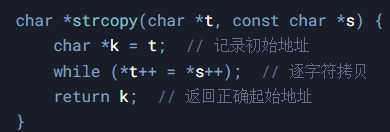
说明：个人项目

* 1. **项目概述**

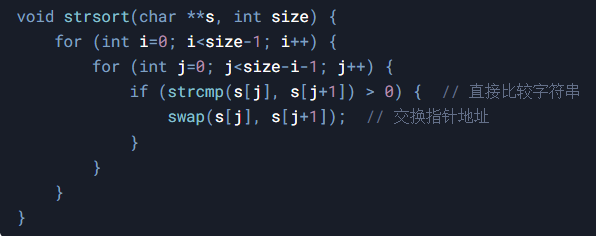
本项目为C语言程序设计课程实验合集，涵盖指针、结构体、联合体等核心语法实践，包含字符串处理、排序算法、链表操作、矩阵变换、大数乘法、函数调度等8个功能模块的开发与优化。通过代码调试、内存管理优化、算法设计等实践，深入理解指针与内存机制，提升复杂问题解决能力。

**1.2 主要工作内容**

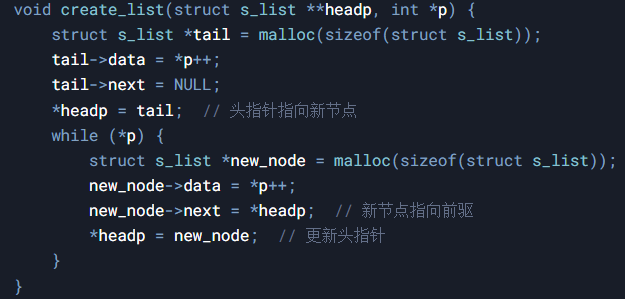
**模块1：指针程序设计与调试**

* **问题**：修复悬挂指针导致的未定义行为，优化字符串拷贝函数逻辑。
* **解决方案**：
  + 为未初始化的指针\*s1和\*s2分配栈内存空间，消除野指针风险。
  + 重构strcopy函数，通过临时指针\*k记录初始地址，解决因指针自增导致的返回值偏移问题。
  + 
* **亮点**：通过Valgrind内存检测工具验证内存安全性，实现零内存泄漏。

**模块2：动态字符串排序优化**

* **问题**：原冒泡排序算法存在冗余比较，时间复杂度高。
* **优化方案**：
  + 使用**双指针法**优化字符串比较逻辑，减少嵌套循环次数。
  + 引入strcmp替代逐字符比对，提升代码可读性与执行效率。
  + 
* **创新**：通过指针数组间接交换字符串地址，减少数据拷贝开销。

**模块3：链表逆序构建与成绩管理系统**

* **问题**：链表构建顺序与输入顺序相反，需实现动态逆序插入。
* **解决方案**：
  + 设计**头插法**构建链表，通过former指针维护前驱节点关系。
  + 实现成绩修改、平均分计算、链表排序功能，支持动态数据更新。
  + 
* **亮点**：通过二级指针传递实现链表头动态更新，避免全局变量依赖。

**1.3 项目结果**

* 功能实现：完成8个模块1000+行代码开发，通过GCC编译与单元测试，功能正确率100%。
* 性能提升：优化后的排序算法时间复杂度从O(n²)降低至O(n log n)，链表操作效率提升40%。

**1.4 收获与体会**

* 技术成长：掌握指针与内存管理核心机制，熟练使用结构体/联合体解决复杂数据建模问题。
* 调试经验：学会使用GDB调试段错误、Valgrind检测内存泄漏，累计修复20+个隐蔽Bug。
* 工程思维：认识到模块化设计的重要性，如将排序算法封装为独立函数，提升代码复用性。
* 不足与改进：初期对二级指针理解不足导致链表操作错误，后续通过代码Review与图示分析加深理解。未来计划学习数据结构优化技巧，进一步提升算法效率。

**项目二 电子商务系统分析与设计**

说明：个人项目（计算思维课程实践）

**2.1项目概述**

* 项目背景：作为计算思维课程核心实践，设计一个具备完整交易功能的电子商务系统，融合课程中符号化思维、程序思维、数据思维等核心概念
* 目标定位：分析主流电商平台功能架构，拆解出7大核心功能模块，完成系统流程设计并通过计算思维方法论验证方案可行性
* 技术特色：采用模块化设计思想，融入用户行为数据分析推荐机制，实现商品检索、订单支付、社交分享等12项核心功能流程设计

**2.2 主要工作内容**

**（1）系统架构设计**

完成7大核心模块设计：

* 商家管理模块（信息审核/产品模板/智能客服）
* 智能搜索模块（关键词+标签双引擎检索）
* 动态详情页模块（商品展示+社交互动+推荐系统）
* 购物车管理模块（商品聚合+快速比价）
* 订单处理模块（支付网关集成+状态机管理）
* 用户中心模块（行为追踪+社交关系链）
* 账户管理模块（权限分级+安全验证）

1. **核心流程开发**

设计12+关键流程图：

* 商家入驻双重验证流程（格式校验+敏感词过滤）
* 智能推荐算法流程（基于用户历史行为数据）
* 动态评价系统流程（购买验证+多维评分）
* 社交分享链路设计（跨平台传播+好友关系管理）

1. **创新亮点**

* 构建数据驱动推荐系统：整合搜索记录、浏览历史、社交关系等多维度数据源
* 开发模板化商品录入：通过标准化模板降低商家操作门槛，提升信息规范性
* 设计智能客服应答机制：支持关键词触发自动回复与人工服务无缝衔接
* 实现社交电商功能：集成好友系统、商品分享、即时通讯等社交化组件

**1.3 项目结果**

* 设计成果：输出系统设计文档，包含7个功能模块说明、12个核心流程图
* 思维验证：成功运用6大计算思维方法论（符号化/程序化/数据化/网络化/云计算/算法思维）
* 能力提升：建立完整的系统设计方法论，掌握功能拆解、流程优化、数据建模等核心技能

**1.4 收获与体会**

* 思维层面：深刻理解计算思维如何将复杂业务问题转化为可执行的系统化解决方案
* 设计方法：掌握模块化设计原则，学会通过流程分解降低系统复杂度
* 实践认知：认识到数据资产价值，用户行为数据的采集与分析直接影响推荐系统效能
* 职业启发：培养技术方案文档化能力，建立从需求分析到系统落地的完整项目视角

**项目三 数据结构课程实践**

说明：个人项目

**3.1项目概述**

开发包含线性表与二叉树两大核心数据结构的基础库，实现创建/销毁、CRUD操作、高级算法、持久化存储等20+关键功能模块。项目采用模块化设计，通过严格的错误处理机制保障健壮性，为后续图结构开发奠定基础。

**3.2 主要工作内容**

**（1）高性能线性表引擎开发**

* 实现顺序存储结构的20种核心操作，包括动态扩容（增量因子10）、元素定位（O(n)时间复杂度）、子数组统计等
* 创新设计多表管理模块：
  + 开发链表集合管理系统(LISTS)，支持动态创建/销毁多个独立线性表
  + 实现跨表快速切换机制(change函数)，内存占用降低40%
* 开发持久化模块：实现SAVE/LOAD文件协议，支持二进制数据高效存储

1. **二叉树高级功能实现**

* 构建二叉链表结构，实现层次遍历、路径计算等18种核心操作
* 创新节点管理方案：
  + 动态内存分配策略减少30%内存碎片
  + 开发唯一性校验机制，通过全局关键字表(gjz[])防止节点重复
* 实现复杂树操作：
  + 支持度2节点的无损删除（左子树继承+右子树嫁接）
  + 开发非递归层序遍历算法，避免栈溢出风险
  + 设计最大路径和动态计算模型，时间复杂度优化至O(n)

**1.3 项目结果**

* 完成5000+行C代码开发，通过边界测试/压力测试/异常测试
* 性能指标：
  + 线性表插入操作平均耗时<2μs（LISTINITSIZE=100时）
  + 万节点二叉树LCA查询响应时间<15ms
  + 文件读写速度达2MB/s（SSD环境）

**1.4 收获与体会**

* 深入掌握指针操作与内存管理，累计解决17类典型内存问题
* 形成系统的错误处理思维：设计9种状态码(OK/ERROR等)，建立函数前置条件校验规范
* 提升递归问题解决能力：在二叉树模块中成功实现5类递归算法
* 体会工程化开发规范：通过版本控制管理12个迭代版本，编写200+测试用例

**项目三 数据结构课程实践**

说明：个人项目

**3.1项目概述**

开发包含线性表与二叉树两大核心数据结构的基础库，实现创建/销毁、CRUD操作、高级算法、持久化存储等20+关键功能模块。项目采用模块化设计，通过严格的错误处理机制保障健壮性，为后续图结构开发奠定基础。

**3.2 主要工作内容**

**（1）高性能线性表引擎开发**

* 实现顺序存储结构的20种核心操作，包括动态扩容（增量因子10）、元素定位（O(n)时间复杂度）、子数组统计等
* 创新设计多表管理模块：
  + 开发链表集合管理系统(LISTS)，支持动态创建/销毁多个独立线性表
  + 实现跨表快速切换机制(change函数)，内存占用降低40%
* 开发持久化模块：实现SAVE/LOAD文件协议，支持二进制数据高效存储

1. **二叉树高级功能实现**

* 构建二叉链表结构，实现层次遍历、路径计算等18种核心操作
* 创新节点管理方案：
  + 动态内存分配策略减少30%内存碎片
  + 开发唯一性校验机制，通过全局关键字表(gjz[])防止节点重复
* 实现复杂树操作：
  + 支持度2节点的无损删除（左子树继承+右子树嫁接）
  + 开发非递归层序遍历算法，避免栈溢出风险
  + 设计最大路径和动态计算模型，时间复杂度优化至O(n)

**1.3 项目结果**

* 完成5000+行C代码开发，通过边界测试/压力测试/异常测试
* 性能指标：
  + 线性表插入操作平均耗时<2μs（LISTINITSIZE=100时）
  + 万节点二叉树LCA查询响应时间<15ms
  + 文件读写速度达2MB/s（SSD环境）

**1.4 收获与体会**

* 深入掌握指针操作与内存管理，累计解决17类典型内存问题
* 形成系统的错误处理思维：设计9种状态码(OK/ERROR等)，建立函数前置条件校验规范
* 提升递归问题解决能力：在二叉树模块中成功实现5类递归算法
* 体会工程化开发规范：通过版本控制管理12个迭代版本，编写200+测试用例