

毕 业 设 计论 文 任 务 书

　 　 理学院 　 学院 信息与计算科学 　　专业

|  |
| --- |
| **一、题目** |
| （小四宋体；字母数字Times new roman；行距1.25倍）  自制操作系统的设计与实现 |
| **二、课题来源及选题依据（研究背景）** |
| （小四宋体；字母数字Times new roman；行距1.25倍）  操作系统是计算机系统中最重要的系统软件，负责硬件资源的管理、任务调度以及用户与硬件之间的交互。作为计算机科学的核心内容，操作系统的原理与实现对掌握计算机体系结构、底层编程以及系统级软件开发具有重要意义。然而，当前大多以操作系统的课程以理论教学为主，实践教学不足，学生往往缺乏深入的系统设计与实现能力。  主流操作系统（如Windows、Linux和macOS）功能复杂、模块庞大，对于初学者而言直接深入其源码存在较高的学习门槛。设计和实现一个从零开始构建的简单操作系统，可以帮助学生从基础模块入手，逐步理解操作系统的核心工作原理，如引导加载、内核设计、进程调度、内存管理和文件系统等。同时，这种从实践中学习的过程能够弥补纯理论教学的不足，有效提升学生的动手能力和对底层系统的理解。  此外，自制操作系统具有较高的创新性和探索空间。在实现过程中，可以自主设计各功能模块并优化实现方案，激发创新思维并锻炼解决复杂问题的能力。这样的课题不仅为学生深入研究现代操作系统打下扎实基础，也为将来从事嵌入式系统开发、系统级软件设计提供实践经验。  综上所述，开展“自制操作系统的设计与实现”课题，能够帮助学生系统学习操作系统理论，掌握底层开发技能，并增强实践和创新能力，具有重要的理论价值和实践意义。 |
| **三、本设计（论文或其他）应达到的要求** |
| （小四宋体；字母数字Times New Roman；行距1.25倍）   1. **基本功能要求** 2. 引导加载程序： 3. 负责实现基础的操作系统引导加载功能，包括加载操作系统内核以及执行硬件初始化过程。 4. 支持从指定的存储设备（例如硬盘）进行引导启动。 5. 内核设计：   设计并实现系统内核，支持以下核心功能   1. 进程管理：能够创建、调度和终止任务，实现多任务切换 2. 内存管理：完成基本的内存分配和释放，支持进程的独立地址空间 3. 文件系统：设计独立完善的文件系统，支持文件的创建、删除、读取和写入操作。 4. 中断处理：实现基本的中断机制，支持硬件设备的响应。 5. 任务管理：   实现任务调度算法（例如FIFO算法），以支持多任务的运行和切换。此外，还需考虑任务的优先级，确保关键任务能够获得及时处理。同时，系统应具备任务监控功能，实时跟踪任务执行状态，以便于管理者做出快速响应和调整。   1. 输入输出管理：   ·实现基础的键盘输入与屏幕输出操作。  ·支持简易的字符设备驱动程序   1. 用户交互：   提供一个命令行界面，以便用户能够与操作系统进行交互。   1. **拓展功能要求** 2. 图形用户界面（GUI）：   实现基础的图形用户界面，支持窗口管理和鼠标操作，提升用户体验。   1. 基本应用程序：   开发一些预置应用程序，如计算器，文本编辑器或图片查看器，展示操作系统的功能拓展能力。  **三、综合性能要求**   1. 系统稳定性：   系统能够在负载较低和中等情况下稳定运行，避免崩溃或死锁。   1. 资源利用率： 2. CPU调度效率：设计合适的任务调度算法（如时间片轮转调度或优先级调度），保证系统内任务能够高效分配CPU时间，减少空闲等待时间。 3. 内存管理优化：实现合理的内存分配与回收策略（如分区内存管理），确保内存使用率最大化，减少碎片化。 4. 模块化设计：功能模块分离：系统各功能模块独立设计，遵循“单一职责原则”，确保模块功能清晰，互不干扰。   接口规范化：模块之间通过统一的接口标准进行通信，避免直接依赖其他模块的内部实现，从而降低模块之间的耦合度。  易维护性：代码注释充分，模块间依赖关系清晰，确保后期修改、调试和升级时方便快速定位问题或调整设计。 |
| **四、任务落实情况**  **任务起止日期:**  **自20xx年xx月xx日 至20xx年xx月xx日**  **接受任务学生：**  **班级 信计2103 学号** 1131210333 **姓名　胡涛涛**  **指导教师**（签名）：    **专业负责人**（签名）：  **教学院长**（签名）：  **年 月 日** |