

课程设计要求

- 功能实现：最好采用教材中的方法进行实现
 - 如果使用其他方法，必须通过实验验证、理论分析等方式，证明该方法在时间和空间性能上是最优的
 - 若选择其他方法，如果无法充分达到验收要求，可能会导致分数较低
- 评判标准：
 - 代码量：有效代码量很重要，800-1000行算较少
 - 专业性：代码需符合操作系统内核的规范（例如不能使用 `printf`，因为它是用户进程的输入输出方式）
- 内核代码：
 - 必须使用C语言或汇编语言进行实现
 - 代码应写成一系列 `.c` 文件，每个文件定义一系列函数，最终将所有 `.c` 文件合并为一个 `.lib` 库文件
 - 若使用Visual Studio，操作系统代码应作为一个独立项目，生成的结果是一个 `.lib` 库文件
- 演示代码：
 - 不限制编程语言
 - 通过演示代码来调用操作系统函数，证明代码的正确性
 - 评分时，演示代码仅作为参考

(详细规范请参考课程设计指导书)

- 优秀基本标准：完整实现所有功能，并且实现方式专业
- 验收时间：在第18周进行

选题要求

- 支持平台：Windows、Unix/Linux、OpenEuler
- 总共有50到60个题目。参加OS竞赛的同学，无需选择题目
- 每个班级内每个人的题目不能重复
- 可以选择单人或1-2人小组合作
 - 如果是两人合作，则必须两人共同完成一个完整的项目（而不是两人独立完成）
- 难度要求：
 - 不要选择难度为1的题目，原则上成绩只能达到“中”等级；难度为2的题目，原则上成绩不能超过“良”等级；只有难度为3及以上的题目，才有可能达到“优秀”
 - 但“原则上”并不是绝对的。如果难度为1的题目实现得非常专业，代码量较大，也可能获得“优秀”；反之，难度为3以上但代码量较少，可能只能获得“中”或“及格”
- 在第15周由班长提交选题结果

竞赛方式参与课程设计

- 验收时需要针对每个小组成员单独进行验收。每个成员需说明自己所做的具体工作
- 报告要围绕自己所做的工作撰写，不得重复
- 即使其他成员表现优秀，但如果小组内有成员的验收质量较差，也可能会影响最终的分数