# 实验8 文件管理

# 实验目的

理解内存页面调度的机理。

掌握几种理论页面值换算法的实现方法

通过实验比较各种调度算法的优劣。

# 实验内容

给出一个磁盘块序列：1、2、3、……、500，初始状态所有块为空的，每块的大小为2k。选择使用位表、链式空闲区、索引和空闲块列表四种算法之一来管理空闲块。对于基于块的索引分配执行以下步骤：

随机生成2k-10k的文件50个，文件名为1.txt、2.txt、……、50.txt，按照上述算法存储到模拟磁盘中。

删除奇数.txt（1.txt、3.txt、……、49.txt）文件

新创建5个文件（A.txt、B.txt、C.txt、D.txt、E.txt），大小为：7k、5k、2k、9k、3.5k，按照与（1）相同的算法存储到模拟磁盘中。

给出文件A.txt、B.txt、C.txt、D.txt、E.txt的文件分配表和空闲区块的状态。

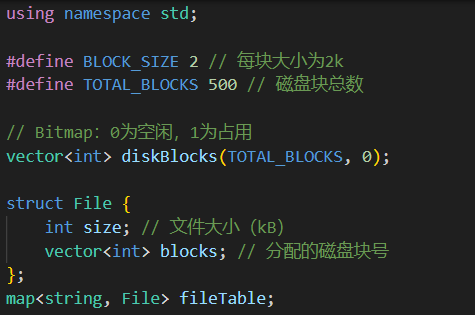
# 实验环境

Ubuntu 22.04.4 LTS

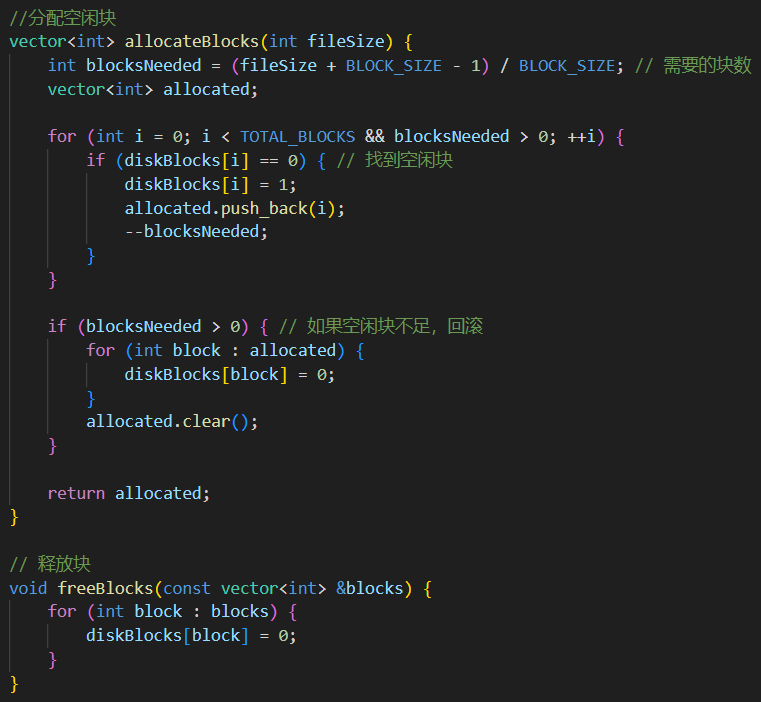
gcc version 11.4.0

# 源代码

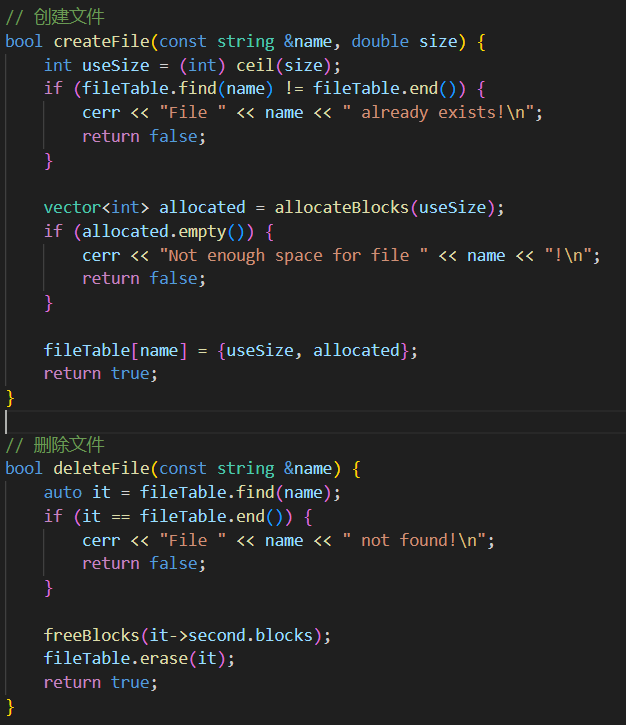
使用位表实现



定义部分,包括宏定义,位表,简单文件存储结构体,以及文件表



分配块与释放块,找到可用空闲块就分配并修改bitmap,直到满足需求或者空间不足



增加文件与删除文件,除了错误输出,基本是使用前面的分配块和删除块并向fileTable写入(删除)记录



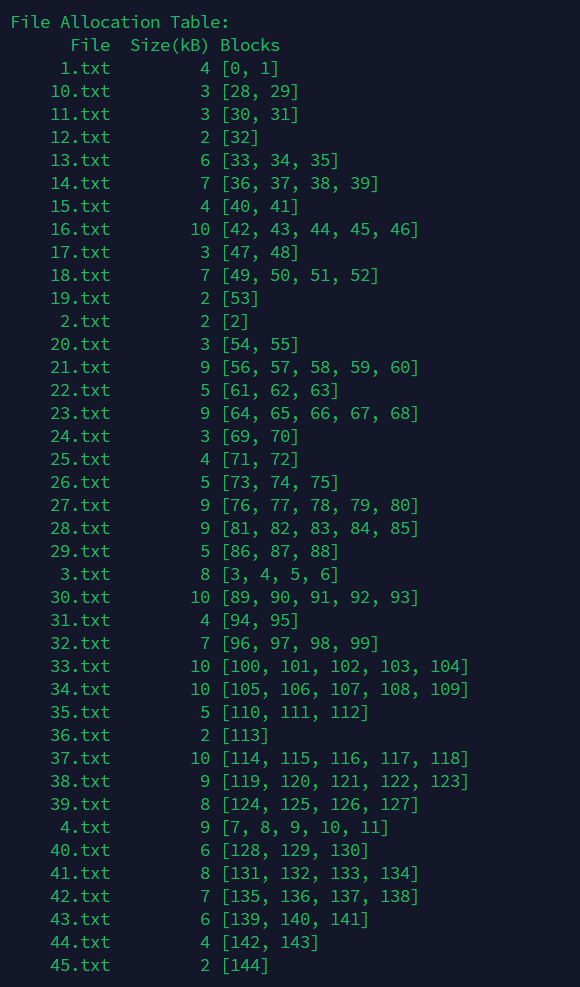
两个格式化打印函数

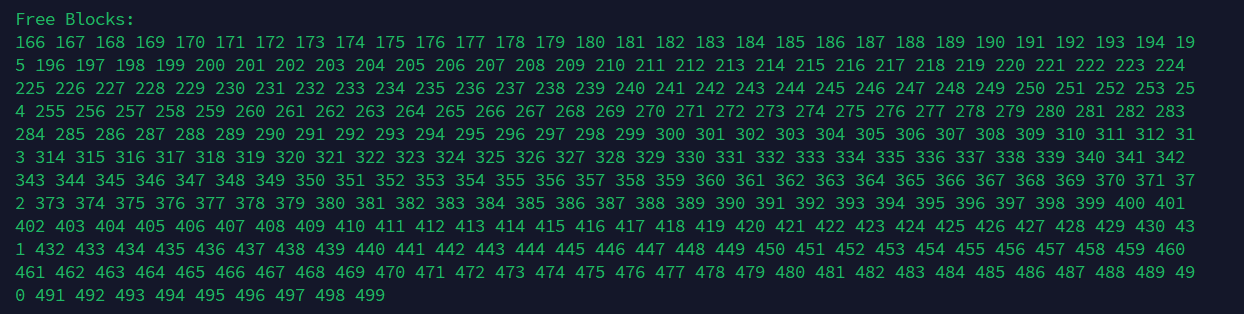


按要求完成代码

# 实验结果

刚开始生成50个文件后的输出:

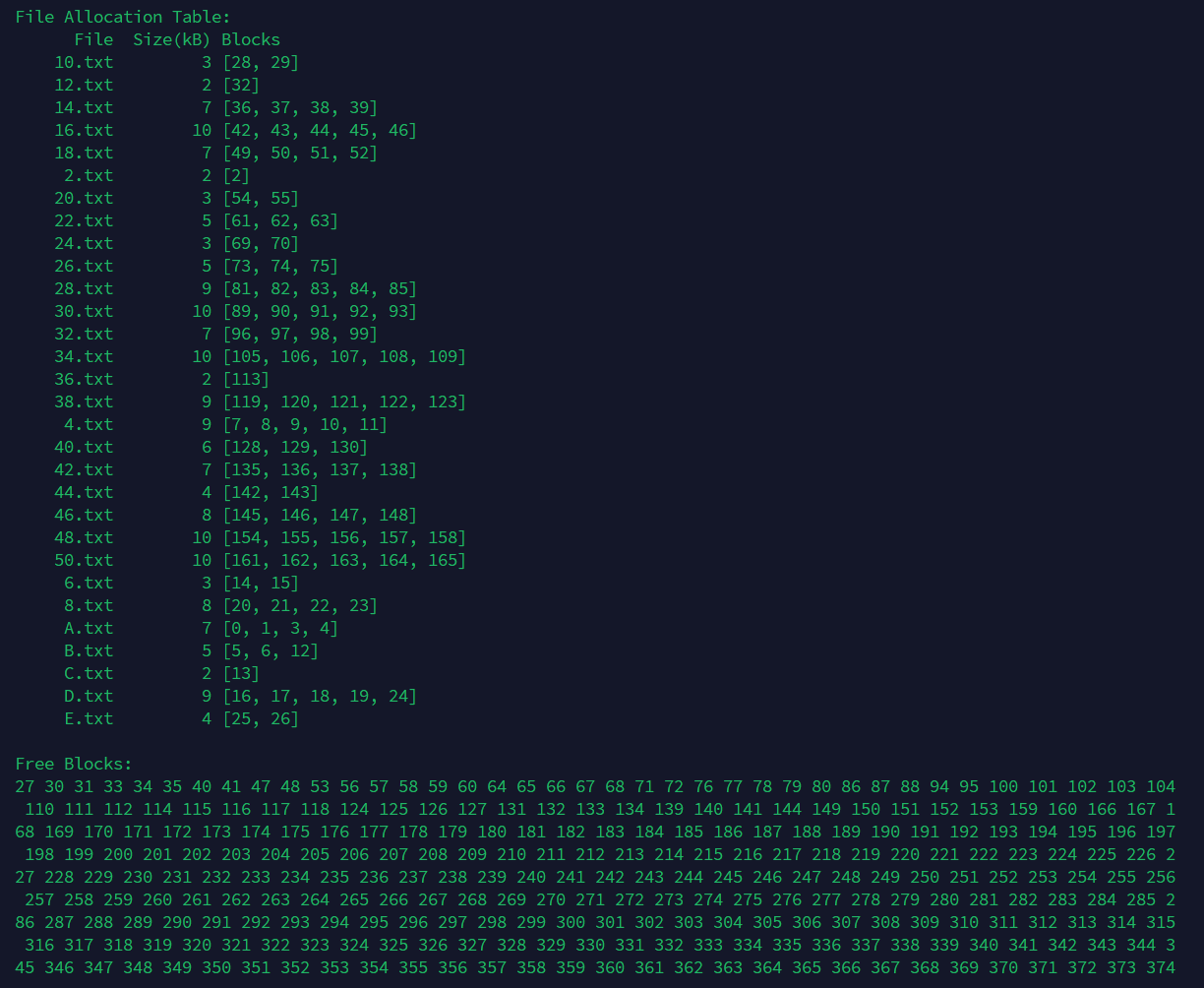




顺序占用了前166个块



删除了奇数号文件,释放了它们所占用的块



增加A-E txt文件,利用了原来部分奇数占用的块

# 实验小结

在本次实验中，我深入探讨了文件系统的工作机理，并理解了多种文件管理算法的实现方法。并模拟了基于位图算法的文件管理。通过本次实验，我们不仅掌握了文件系统的基本原理和操作，还深入理解了不同文件管理算法的特点和适用场景。这为我们在实际应用中选择合适的文件管理策略提供了宝贵的参考。实验过程中，我们也体会到算法设计和实现的复杂性，以及在不同场景下权衡性能和资源利用的重要性。