**实验四 存储管理的模拟程序设计**

**一、实验目标**

通过本实验使学生深入理解计算机存储器管理的连续分配方式或离散分配方式的分页技术的思想与实现方法，使学生更好分析和掌握连续或分页实现内存管理的特点，达到对计算机学科基础实践能力、问题分析/设计/算法实现的专业核心能力、综合创新能力的培养。

**二、实验要求**

从基于连续分配思想的**动态分区分配方案**和基于离散分配思想的**分页**中选择一种方案进行内存空间的模拟管理，具体管理工作包括：

1. 模拟实现1MB内存空间的管理；
2. 利用内存空间的数据结构（如分页系统中空闲块链表或位示图法），记录空间的使用情况；
3. 设计内存分配算法（如动态分区中可采用首次适应法、最佳适应法、最坏适应法或循环首次适应法）和过程，完成要运行的进程的空间分配，可能要考虑进程分配空间的记录，比如页表的设计；
4. 当进程运行完成时，设计内存回收算法（如动态分区中考虑相邻空间的合并）实现释放空间的回收；
5. 可动态显示内存使用状况；
6. 针对用户给出的逻辑地址能计算出对应的物理地址。

**三、主要仪器设备**

实验环境：云桌面计算机，Windows 2000及以上操作系统 / Linux

实验开发工具：Visual Studio / GCC

**四、考核方式及要求**

考核方式包括：考勤、实验操作、实验报告三方面，所占比为10%、20%、70%。

要求学生上机实验前进行预习，上机时进行考勤；实验过程中对学生操作过程进行考察，优先高质量完成的同学实验操作分应予以体现；实验报告具体考核要求如下：

1. 代码中数据结构、分配算法与整体流程设计的描述与总结是否正确；
2. 验证代码的数据设计是否合理；
3. 实验报告是否完整及总结有否收获。

**动态分区分配**

**一、实验目标**

开发一个C语言程序实现内存空间管理的动态分区分配方案。

**二、实验原理**

动态分区分配：根据进程的实际需要，动态地创建分区为之分配内存空间，在实现动态分区分配时，将涉及分区分配中所使用的数据结构，分区分配算法和分区的分配与回收操作等问题。

1. 分区分配中的数据结构
   * 空闲分区表：一个数据表，用于记录每个空闲块的情况，如起始地址、大小、使用情况等；
   * 空闲分区链表：把所有的空闲分区链接成一个链表，便于内存空间查看与分配回收。
2. 内存分配过程

利用分配算法找到满足要求的内存块，设请求的内存大小为size：

* + 若找到的空闲分区的大小等于size，完全分配；
  + 若找到的空闲分区大小大于size，且一分为二后，剩余大小小于1K，则不再分割，作为整体进行分配；否则一分为二，剩余部分仍然作为空闲分区存在；
  + 若无满足要求空闲分区，则分配失败

1. 分配算法
   * 首次适应法：空闲分区按首地址递增次序组织，每次查找时从链首出发，寻找满足要求的内存块。
   * 最佳适应法：空闲分区按空闲分区大小址递增次序组织，每次查找时从链首出发，寻找满足要求的最小内存块进行分配。
   * 最坏适应法：空闲分区按空闲分区大小递减次序组织，每次查找时直接判断最大空闲分区是否满足要求。
2. 内存回收

根据释放区首址和大小，查找空闲分区表/链表，判断是否有相邻的空闲分区存在：

* + 释放区与前空闲区相邻：将释放区与前空闲区合并为一个空闲区。其首址仍为前空闲区首址，大小为释放区大小与空闲区大小之和。
  + 释放区与前后两个空闲区相邻：将这三个区合为一个空闲区，其首址为前空闲区首址，大小为这三个区大小之和，并取消原后空闲区表目。
  + 释放区与后空闲区相邻：则把释放区合并到后空闲，首地址为释放区首地址，大小为二者大小之和。
  + 释放区不与任何空闲区相邻：将释放区作为一个空闲区，将其大小和首址插入到空闲区表的适当位置。

**分页技术实现**

**一、实验目标**

开发一个C语言程序实现内存管理的分页技术。

**二、实验内容**

1、获取物理内存的总大小和逻辑页面大小；

2、对物理内存进行分块；

3、初始化物理内存的数据结构，注意内存中的块有些应当为已分配，有些为空闲，可通过随机生成；

4、构建空闲块链表或者位示图；

5、获取进程的内存需求，将进程的程序地址空间进行划分，n个页面；

6、如果有n个物理块可用，则给进程分配并更新进程页表；

7、能完成一些逻辑地址到相应物理地址的转换；

8、完成进程的相应空间回收；

9、能重复完成多个进程的空间管理。