**安徽师范大学**

**数计学院 学院实验报告**

**专业名称 2015级物联网工程**

**实 验 室 2060302**

**实验课程 计算机操作系统**

**实验名称 动态分区分配算法**

**组 长 孙浩**

**同组人员 孙洋，台昌杨，杨娜娜，凌莉**

**实验日期 2017/11/16**

|  |  |  |
| --- | --- | --- |
| 1. **实验目的**   掌握动态分区分配方式中的数据结构、分配算法，针对不同的分配算法如何实现内存空间的分配与回收，必要时如何实现“紧凑”。   1. **实验内容**   编写一个内存动态分区分配模拟程序，分别实现：首次适应、循环首次适应、最佳适应算法，对内存的分配和回收过程，必要时进行“紧凑”。  每次分配和回收后把空闲分区的变化情况以及各进程的申请、释放情况最好以图形方式显示，尽可能设计一个友好的用户界面，直观显示内存区域经分配、回收、紧凑后的动态变化情况。  （1）采用链表表示内存使用情况，链表中的结点可以给出对应的某块内存区域的信息，如：区号、起始地址、大小、使用情况（是否空闲）、所装入的进程名等。可以设置两个链表，一个是空闲分区表，一个是已分配分区表。  通过菜单的选项来完成进程对内存的申请或释放操作。  （2）选择分配算法，根据进程申请的内存空间实施分配，若分配成功，返回所得的内存首地址，并显示调整后的空闲分区表。若没有单个空闲分区满足进程需求，而紧凑后可以满足，则实施紧凑并分配。若紧凑后仍不能满足，则分配失败。  （4）进程结束后，回收其占有的内存，按内存回收的四种情况进行回收。  （5）无论是分配还是回收，都要按相应的分配算法对空闲分区的组织要求重新排序。  **三、各动态分区分配算法的设计思想**  **（1）首次适应算法（First Fit）：**  从空闲 分区表的第一个表目起查找该表，把最先能够满足要求的空闲区分配给作业，这种方法目的在于减少查找时间。为适应这种算法，空闲分区表（空闲区链）中的空闲分区要按地址由低到高进行排序。该算法优先使用低址部分空闲区，在低址空间造成许多小的空闲区，在高 地址空间保留大的空闲区。  **（2）循环首次适应算法（Next Fit）：**  在为进程分配内存空间时，不再每次从链首开始查找，直至找到一个能满足要求的空闲分区，并从中划出一块来分给作业，使内存中的空闲分区分布的更为均匀，减少了查找时的系统开销。  **（3）最佳适应算法（Best Fit）：**  将所有的空闲区按其大小排序后，以递增顺序形成一个空白链。这样每次找到的第一个满足要求的空闲区，必然是最优的。孤立地看，该算法似乎是最优的，但事实上并不一定。因为每次分配后剩余的空间一定是最小的，在存储器中将留下许多难以利用的小空闲区。同时每次分配后必须重新排序，这也带来了一定的开销，每次分配给文件的都是最合适该文件大小的分区。 | | |
| **四、实验设备及其环境**  （1）电脑配置：    （2）编译软件    **五、算法比较**  **首次适应算法（First Fit）：**    优点： 该算法倾向于使用内存中低地址部分的空闲区，在高地址部分的空闲区很少被利用，从而保留了高地址部分的大空闲区。显然为以后到达的大作业分配大的内存空间创造了条件。    缺点：低地址部分不断被划分，留下许多难以利用、很小的空闲区，而每次查找又都从低地址部分开始，会增加查找的开销。  **循环首次适应算法（Next Fit）：**    优点：使内存中的空闲分区分布的更为均匀，减少了查找时的系统开销。    缺点：缺乏大的空闲分区，从而导致不能装入大型作业。  **最佳适应算法（Best Fit）：**  优点：每次分配给文件的都是最合适该文件大小的分区。    缺点：内存中留下许多难以利用的小的空闲区。  **六、流程图**  First Fit释放空间算法流程    First Fit 空间申请算法    **七、实验结果及其分析**  详情见附件“[测试数据](测试数据.docx)” 。 | | |
|  | | |
| 姓 名 | 承 担 主 要 工 作 | 小组评价 |
| **孙浩** | **代码编写** |  |
| **孙洋** | **流程图和文档整理** |  |
| **台昌杨** | **代码编写** |  |
| **杨娜娜** | **测试数据** |  |
| **凌莉** | **文档编写** |  |