

模拟实现动态分区内存管理

1. 实验目的

通过模拟实现动态分区存储分配算法，加深对动态分区存储管理的理解。

2. 预备知识

动态分区分配又称可变分区分配，是一种动态划分内存的分区方法，这种分区方法是在进程装入内存时，根据进程的大小动态地建立分区，并使分区的大小正好适合进程的需要。基于顺序搜索的动态分区分配算法有如下四种：

- 首次适应算法：在空闲区链表中从头开始查找符合申请大小的块，找到大小能满足申请要求的第一个空闲块。
- 最佳适应算法：先将空闲区按照容量大小从小到大排序，再从排序后的空闲区链表找到大小能满足申请要求的第一个空闲块。
- 最坏适应算法：先将空闲区按照容量大小从大到小排序，再从排序后的空闲区链表找到大小能满足申请要求的第一个空闲块。
- 循环首次适应算法：由首次适应算法改进而成，不同于首次适应算法每次都从头开始查找，循环首次适应算法是从上次查找结束的位置开始继续查找。

3. 实验内容

利用首次适应、最佳适应、最坏适应及循环首次适应算法中的一种分配算法模拟实现对内存空间的管理。程序一开始接收一个输入 `n` 表示待分配的内存空间大小；之后可以输入操作命令，命令格式如下：

分配空间命令： `alloc [PID] [size]`

释放空间命令： `free [PID]`

输出内存分配信息命令： `memstat`

退出程序： `exit`

其中PID为进程号，size为需要分配的空间大小；内存分配信息应显示各进程在内存中的起始地址和占用大小，以及空闲空间的起始地址和大小。

对于剩余内存空间不够分配或是所释放进程不存在的情况，应输出 `ERROR`。