|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | 实际  思路+写代码+调试及修正:(1 +1.5+ 1.5)  其实:调试用半小时就行了；但为了更好，花了比较多的时间  对比学习neo +修正自己写的(2+2) | 解决 | 备注  (不关大局:-2处 |
| 12/4  可变分区：  最坏适应算法 | 思路: | 最坏适应算法：  把空闲区按容量从大到小排列了，查看时只要看第一个分区能否满足作业的要求与否。  见稿纸 | 基于顺序表 | C++ ,题原理比较熟，写起来容易点；但也不敢大意  关键点： |
| 对比点 | 见下表2 |  | **练熟：文件的操作** |
| 调试 | 调整的合理点 |  |  |
| 1h | 1小的语法错误:  using namespace std; |  |  |
| 2 do{}while();  Do  {  //执行的语句  cout<<"继续进行内存的申请吗？请输入(yes/no)\n";  cin>>ch;  new\_line();  }while( (ch!='N') && (ch!='n') ); |  |  |
| 0.5h | 3 关于接收字符串  0: Cin>>name; //当输入空格就被截断了，所以不太合适  A: Getline()函数的使用  b.: cin.getline（）后第二次接收到位空字符？  C:改用strcpy函数 | A:  Getline()的敲两下回车才能完成;  B  char c\_sNo[50];  cout<<"请输入学号,不能重复"<<endl;  new\_line(); //不知为何有时可以去 cin.getline(c\_sNo,50,'\n');  //这样c\_sNo就接收了字符串了 | void new\_line()  {  char symbol;  do  {  cin.get(symbol);  }while(symbol!='\n');  } |
|  | 4: ++kxqNum | 需把文件的读写函数吃透，  读文件返回的记录数错，是不明白什么情况下文件结束 |  |
| 待完善的点 | 输入数字 要为正数的验证  程序名的输入 |  |  |
|  | 总结 | 三种算法是基于顺序表的。（链表不好操作，且麻烦） | (20130427)提示当要求输入两个数，而结束符以0结束，**必须输入两个数才行**；  Int a,b;  printf("请输入内存的起始地址 长度(用空格隔开)");  scanf("%d %d",&a,&b);  while(a!=0)  {  printf("您输入的是:%d,%d",a,b);  scanf("%d %d",&a,&b);  }  printf("\n已结束了\n"); | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 表2 | 对比点（网上Neo的答案） | fy自己 | |  |  |
|  |  | 明显错误 | 微调点 |  |  |
| 三存储管理 | :1：代码分格不错,挺好的。  不足：最坏适应算法的思路不精练 | **空闲区数据文件**  **而自己是从键盘输入 (已改)** | (完)最后输出新的空闲区表和已分配区表的内容  不要用户每申请一次就输出 |  |  |
| **(顺序表**  **当空闲区不可用时，应把其覆盖掉** |  |
|  |  |
|  |  |
|  | 区别点:  A: 最坏适配算法核心思想  b最坏适配算法的空闲区容量为0时处理:  方案2:  当空闲区的容量为0是，元素前移，把此空闲区覆盖，相当于释放 | A:  B: 学习了 |  |  |  |
|  |  |  |  |  |  |