

实验10 连续空间存储管理模拟 实验

```
struct AllocatedSpace{  
    int start_address;  
    int length;  
    char job;  
    struct AllocatedSpace *next;  
};
```

```
struct FreeSpace{  
    int start_address;  
    int length;  
    struct FreeSpace *next;  
};
```

```
struct AllocatedSpace *allocated_header;  
struct FreeSpace *free_header;
```

实验要求

- 采用链表进行空闲区和已分配空间的管理
- 初始的内存大小为100000字节
- 采用最先适应分配算法分配内存
- 功能键
 - 0 退出
 - 1 为作业分配内存
 - 输入：作业名 要求的内存大小
 - 执行逻辑：若无满足要求大小的空闲区，则显示无法分配的消息并退出，否则依据最佳适应分配算法找到空闲区，分配给作业（有可能需要对空闲区进行切割），并在已分配链表中进行登记
 - 2 回收内存
 - 输入：作业名
 - 执行逻辑：在已分配链表中找到给定的作业名对应的项，将对应的内存回收，修改已分配链表和空闲区链表（有可能需要在空闲区链表中进行合并）
 - 3 显示内存分配情况
 - 已分配内存空间情况（作业名，起始地址, 长度）和空闲区情况（起始地址，长度）

连续内存分配算法测试案例

- 分配 A 3000
- 分配 B 3000
- 分配 C 5000
- 分配 D 15000
- 分配 E 2000
- 回收 B
- 分配 F 4000
- 回收 E
- 分配 H 90000
- 分配 G 1500
- 回收 D
- 回收 G
- 回收 A