操作系统课程实验报告

学生姓名：/

班 学 号：/

指导教师：/

地理与信息工程学院

2019 年 6 月 8 日

实习题目：内存管理模型的设计与实现

【需求规格说明】

对内存的可变分区申请采用链表法管理进行模拟实现。要求：

（1） 对于给定的一个存储空间自己设计数据结构进行管理，可以使用单个链表，也可以使用多个链表，自己负责存储空间的所有管理组织，要求采用分页方式（指定单元大小为页，如 4K，2K，进程申请以页为单位）来组织基本内容；

（2） 当进程对内存进行空间申请操作时，模型采用一定的策略（如：首先利用可用的内存进行分配，如果空间不够时，进行内存紧缩或其他方案进行处理）对进程给予分配；

（3） 从系统开始启动到多个进程参与申请和运行时，进程最少要有 3 个以上，每个进程执行申请的时候都应能对系统当前的内存情况进行查看；

（4） 对内存的申请进行内存分配，对使用过的空间进行回收，对给定的某种页面调度进行合理的页面分配。

（5） 利用不同的颜色代表不同的进程对内存的占用情况，动态更新这些信息。

【算法设计】

（1） **设计思想：**

用链表存储每个被进程申请的内存块与空闲的内存块，初始链表只有两个节点，存放系统占用内存与空的内存块，每页大小为 1k。

申请内存时提供四种分配方法：最先匹配法、下次匹配法、最佳匹配法和最坏匹配法。最先匹配法：遍历链表，找到的第一个符合要求的空块分配；下次匹配法：每次分配完内存纪录当前位置，下次分配内存时从当前位置开始，若无则从头再遍历链表；最佳匹配法：遍历链表，纪录符合要求的最小的块的节点用于分配；最坏匹配法：遍历链表，纪录符合要求的最大的块的地址用于分配。

函数具体实现见代码。

（2） **详细设计：**

链表节点构成：

int begin;	起始地址
int size;	块大小
Node next;*	指向下一节点
string process;	状态(os->操作系统,hole->空闲区,process->进程)

我主要设计了一个类 *MemoryManage*，来模拟内存管理模型：

成员变量说明：

*Node *head;*	链表头节点
*Node *current;*	纪录当前位置，服务于下次匹配法
Int size;	纪录内存大小

函数说明：

MemoryManage(int size=256, int osSize=8);	构造函数，默认内存大小 256，操作系统占用 8
~MemoryManage();	析构函数，释放申请的内存
void init(int size, int osSize);	类初始化，相关变量初始化
bool request(string process, int size, int mode);	进程按照相关方法申请内存
void assignMemory(Node node, string process, int size);*	将 node 节点分配给进程
bool firstFit(string process, int size);	最先匹配法
bool bestFit(string process, int size);	最佳匹配法
bool nextFit(string process, int size);	下次匹配法
bool worstFit(string process, int size);	最坏匹配法
void print();	打印相关信息
void scanf();	用户输入交互
bool memoryRecycling(string process);	内存回收
bool tighteningMemory();	内存紧缩，即将不连续的空块连续

程序执行过程：初始化->用户选择申请内存/释放内存/退出->系统进行相关操作。

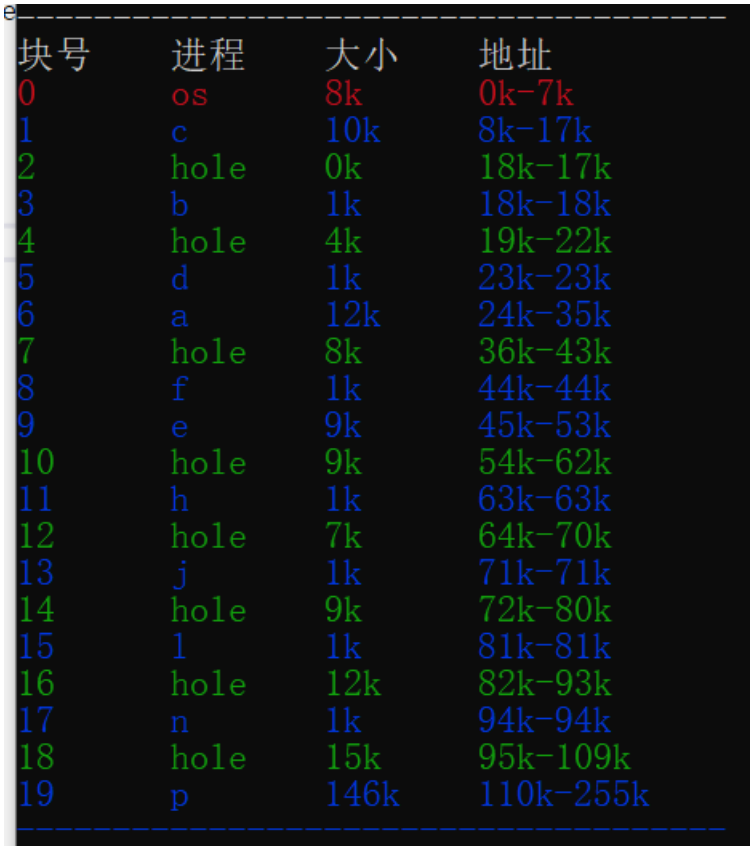
【调试报告】

内存分配测试，用书上的习题做测试，先模拟出初始情况：

块号	进程	大小	地址
0	os	8k	0k-7k
1	hole	10k	8k-17k
2	b	1k	18k-18k
3	hole	4k	19k-22k
4	d	1k	23k-23k
5	hole	20k	24k-43k
6	f	1k	44k-44k
7	hole	18k	45k-62k
8	h	1k	63k-63k
9	hole	7k	64k-70k
10	j	1k	71k-71k
11	hole	9k	72k-80k
12	l	1k	81k-81k
13	hole	12k	82k-93k
14	n	1k	94k-94k
15	hole	15k	95k-109k
16	p	146k	110k-255k

依次用各种分配方法申请：a->12k，c->10k，e->9k：

最先匹配法：

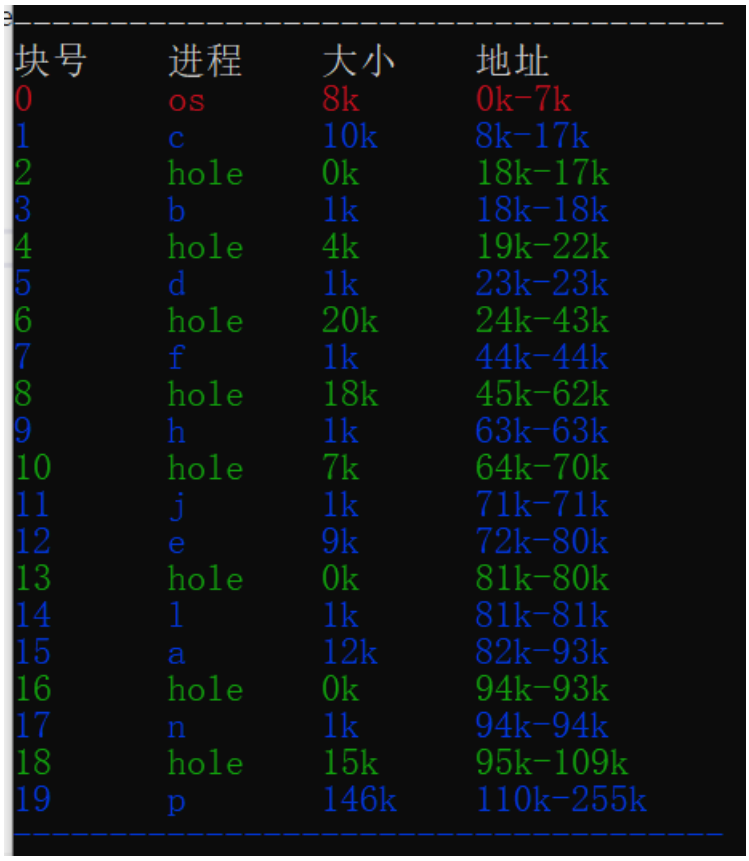


块号	进程	大小	地址
0	os	8k	0k-7k
1	c	10k	8k-17k
2	hole	0k	18k-17k
3	b	1k	18k-18k
4	hole	4k	19k-22k
5	d	1k	23k-23k
6	a	12k	24k-35k
7	hole	8k	36k-43k
8	f	1k	44k-44k
9	e	9k	45k-53k
10	hole	9k	54k-62k
11	h	1k	63k-63k
12	hole	7k	64k-70k
13	j	1k	71k-71k
14	hole	9k	72k-80k
15	l	1k	81k-81k
16	hole	12k	82k-93k
17	n	1k	94k-94k
18	hole	15k	95k-109k
19	p	146k	110k-255k

下次匹配法：

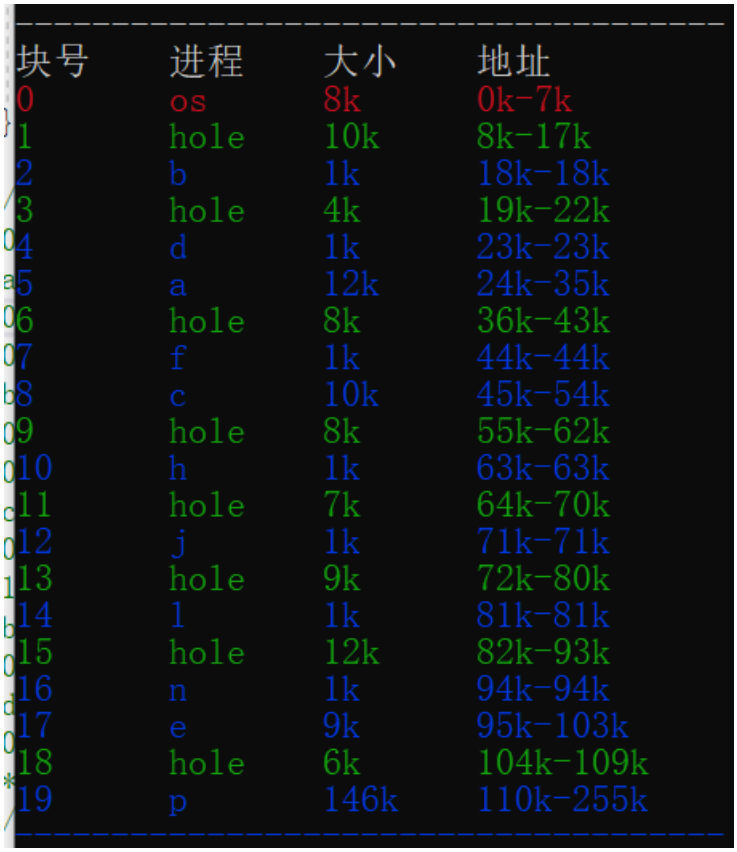
块号	进程	大小	地址
0	os	8k	0k-7k
1	hole	10k	8k-17k
2	b	1k	18k-18k
3	hole	4k	19k-22k
4	d	1k	23k-23k
5	a	12k	24k-35k
6	hole	8k	36k-43k
7	f	1k	44k-44k
8	c	10k	45k-54k
9	hole	8k	55k-62k
10	h	1k	63k-63k
11	hole	7k	64k-70k
12	j	1k	71k-71k
13	e	9k	72k-80k
14	hole	0k	81k-80k
15	l	1k	81k-81k
16	hole	12k	82k-93k
17	n	1k	94k-94k
18	hole	15k	95k-109k
19	p	146k	110k-255k

最佳匹配法：



块号	进程	大小	地址
0	os	8k	0k-7k
1	c	10k	8k-17k
2	hole	0k	18k-17k
3	b	1k	18k-18k
4	hole	4k	19k-22k
5	d	1k	23k-23k
6	hole	20k	24k-43k
7	f	1k	44k-44k
8	hole	18k	45k-62k
9	h	1k	63k-63k
10	hole	7k	64k-70k
11	j	1k	71k-71k
12	e	9k	72k-80k
13	hole	0k	81k-80k
14	l	1k	81k-81k
15	a	12k	82k-93k
16	hole	0k	94k-93k
17	n	1k	94k-94k
18	hole	15k	95k-109k
19	p	146k	110k-255k

最坏匹配法：



块号	进程	大小	地址
0	os	8k	0k-7k
1	hole	10k	8k-17k
2	b	1k	18k-18k
3	hole	4k	19k-22k
4	d	1k	23k-23k
5	a	12k	24k-35k
6	hole	8k	36k-43k
7	f	1k	44k-44k
8	c	10k	45k-54k
9	hole	8k	55k-62k
10	h	1k	63k-63k
11	hole	7k	64k-70k
12	j	1k	71k-71k
13	hole	9k	72k-80k
14	l	1k	81k-81k
15	hole	12k	82k-93k
16	n	1k	94k-94k
17	e	9k	95k-103k
18	hole	6k	104k-109k
19	p	146k	110k-255k

在初始时申请 x->30k(此时内存中空闲块均小于 30k)内存不足时执行内存紧缩：

【附录】

代码位置：

https://github.com/Xhofe/Operating-system/tree/master/src/lab3