# 实验二 主存空间的分配和回收

1、实验内容

在分页式管理方式下采用位示图来表示主存分配情况，实现主存分配和回收。

2、基本思想

“在采用分页存储机制的过程中， 为了能够记录个物理块的分配使用情况，以及未分配的物理块的总数，可以对整个内存建立一张存储分块表(MBT)。而存储分配表可以用位示图的方法，这种方法实在内存中划分一块固定的区域，每个单元的每个二进制位代表一个物理块号，初始化死每个二进制位置为 0，如果某块已被分配，则将对应的二进制位置 1。”

在本实验中，我编写的程序可以给进程分配空间，释放某进程的空间，查看所有的进程信息和位示图。具体的做法是将所有进程的 PCB 组成一个双向链表，每个 PCB 中保存页表信息。插入进程的方法是为新进程建立一个 PCB，然后在位示图中寻找相应的空闲空间。给新进程分配这些空间，在 PCB 中记录，并将位示图中的相应位至 1。删除进程的方法是通过查看该进程的 PCB，找到分配给这个进程的空间然后释放，并且将位示图中的相应位置 0。具体请看流程图。

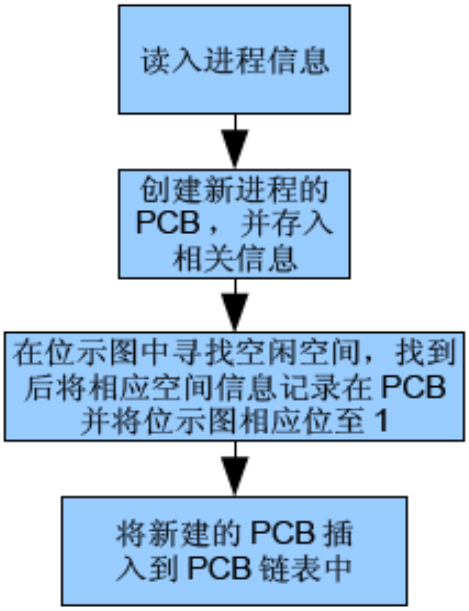


*Illustration 1:* 插入进程，并为其分配内存

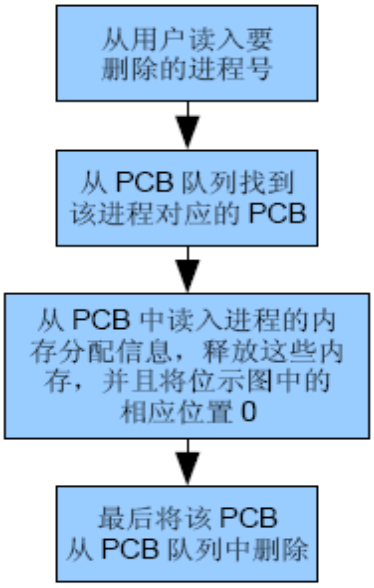


*Illustration 2:* 进程及其空间的回收

**3**、实验平台及软件

Debian Linux 4.1.1-21 with Linux Kernel 2.6.18-6-686, GCC version 4.1.2 and GDB version 6.4.90

**4**、调试过程

（1）输入数据

```
1 p1 8          //插入一个名称为 p1，占用内存大小为 8 的进程
1 p2 8          //插入一个名称为 p2，占用内存大小为 8 的进程
1 p3 8          //插入一个名称为 p3，占用内存大小为 8 的进程
4               //显示当前位示图
2 1             //删除 1 号进程
4               //显示当前位示图
1 p4 16         //插入一个名称为 p4，占用内存大小为 16 的进程
4               //显示当前位示图
3               //显示当前进程列表
5 2             //显示 2 号进程对应的内存分配情况
6               //退出进程空间管理模拟程序
```

*Drawing 1:* 输入数据

（2）输出数据

```
There are 64 free blocks of memory left!
1 --- Insert a process into memory
2 --- Remove a process from memory
3 --- Display current processes in memory
4 --- Display current bitmap
5 --- Display page_table of a specific process
6 --- Exit
Select your choice:1
Please input the process's name(less than 64 chars):p1
Please input the process's size(larger than 0):8

There are 56 free blocks of memory left!
1 --- Insert a process into memory
2 --- Remove a process from memory
3 --- Display current processes in memory
4 --- Display current bitmap
5 --- Display page_table of a specific process
6 --- Exit
Select your choice:1
Please input the process's name(less than 64 chars):p2
Please input the process's size(larger than 0):8

There are 48 free blocks of memory left!
1 --- Insert a process into memory
2 --- Remove a process from memory
3 --- Display current processes in memory
4 --- Display current bitmap
5 --- Display page_table of a specific process
6 --- Exit
Select your choice:1
Please input the process's name(less than 64 chars):p3
Please input the process's size(larger than 0):8

There are 40 free blocks of memory left!
1 --- Insert a process into memory
2 --- Remove a process from memory
3 --- Display current processes in memory
4 --- Display current bitmap
5 --- Display page_table of a specific process
6 --- Exit
Select your choice:4

There are 40 free blocks of memory left!
Current Bitmap:
        0       1       2       3       4       5       6       7
0       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1
2       1       1       1       1       1       1       1       1
3       0       0       0       0       0       0       0       0
4       0       0       0       0       0       0       0       0
5       0       0       0       0       0       0       0       0
6       0       0       0       0       0       0       0       0
```

*Drawing 2:* 输出样例，第一部分

```
7       0       0       0       0       0       0       0       0

There are 40 free blocks of memory left!
1 --- Insert a process into memory
2 --- Remove a process from memory
3 --- Display current processes in memory
4 --- Display current bitmap
5 --- Display page_table of a specific process
6 --- Exit
Select your choice:2
Current Processes:
Process id 0, name p1, occupies 8 blocks of memory.
Process id 1, name p2, occupies 8 blocks of memory.
Process id 2, name p3, occupies 8 blocks of memory.
Which one(id) do you want to remove(-1 to return)?:1

There are 48 free blocks of memory left!
1 --- Insert a process into memory
2 --- Remove a process from memory
3 --- Display current processes in memory
4 --- Display current bitmap
5 --- Display page_table of a specific process
6 --- Exit
Select your choice:4

There are 48 free blocks of memory left!
Current Bitmap:
        0       1       2       3       4       5       6       7
0       1       1       1       1       1       1       1       1
1       0       0       0       0       0       0       0       0
2       1       1       1       1       1       1       1       1
3       0       0       0       0       0       0       0       0
4       0       0       0       0       0       0       0       0
5       0       0       0       0       0       0       0       0
6       0       0       0       0       0       0       0       0
7       0       0       0       0       0       0       0       0

There are 48 free blocks of memory left!
1 --- Insert a process into memory
2 --- Remove a process from memory
3 --- Display current processes in memory
4 --- Display current bitmap
5 --- Display page_table of a specific process
6 --- Exit
Select your choice:1
Please input the process's name(less than 64 chars):p4
Please input the process's size(larger than 0):16

There are 32 free blocks of memory left!
1 --- Insert a process into memory
2 --- Remove a process from memory
3 --- Display current processes in memory
4 --- Display current bitmap
```

*Drawing 3:* 输出样例，第二部分

```
5 --- Display page_table of a specific process
6 --- Exit
Select your choice:3
Current Processes:
Process id 0, name p1, occupies 8 blocks of memory.
Process id 2, name p3, occupies 8 blocks of memory.
Process id 3, name p4, occupies 16 blocks of memory.

There are 32 free blocks of memory left!
1 --- Insert a process into memory
2 --- Remove a process from memory
3 --- Display current processes in memory
4 --- Display current bitmap
5 --- Display page_table of a specific process
6 --- Exit
Select your choice:5

Current Processes:
Process id 0, name p1, occupies 8 blocks of memory.
Process id 2, name p3, occupies 8 blocks of memory.
Process id 3, name p4, occupies 16 blocks of memory.

Which one(id) do you want to display(-1 to return)?:2

Page_Number     Block_Number
        0               16
        1               17
        2               18
        3               19
        4               20
        5               21
        6               22
        7               23

There are 32 free blocks of memory left!
1 --- Insert a process into memory
2 --- Remove a process from memory
3 --- Display current processes in memory
4 --- Display current bitmap
5 --- Display page_table of a specific process
6 --- Exit
Select your choice:6
```

*Drawing 4:* 输出样例，第三部分

## **5**、总结

在这个实验中，我实现了使用双向链表技术的 PCB 管理，实现了利用位示图法进行的进程空间的分配与回收。通过这次实验，我对操作系统中内存的管理有了进一步的理解。

## **6**、参考文献

黄水松，黄干平，曾平，李蓉蓉。（2003）。《计算机操作系统》。武汉：武汉大学。