

# 实验一 可变分区存储管理

---

## ✦ (一) 实验题目

编写一个C程序，用`char *malloc(unsigned size)`函数向系统申请一次内存空间（如`size=1000`，单位为字节），用循环首次适应法

`addr = (char *)lmalloc(unsigned size)` 和  
`lfree(unsigned size, char * addr)`

模拟UNIX可变分区内存管理，实现对该内存区的分配和释放管理。



## (二) 实验目的

---

- ✦ 1. 加深对可变分区的存储管理的理解;
- ✦ 2. 提高用C语言编制大型系统程序的能力, 特别是掌握C语言编程的难点: 指针和指针作为函数参数;
- ✦ 3. 掌握用指针实现链表和在链表上的基本操作。



## （三） 实验要求

空闲存储区表可采用结构数组的形式（基本要求）或双向链接表的形式（提高一步），建议采用的数据结构为：

结构数组的形式：

```
struct map {  
    unsigned m_size;  
    char * m_addr;  
};
```

```
struct map coremap[N];
```

要分配函数**lmalloc**的参数**size**和释放函数**lfree**的参数**size**、**addr**以键盘命令的形式输入，每次分配和释放后显示自己的空闲存储区表。

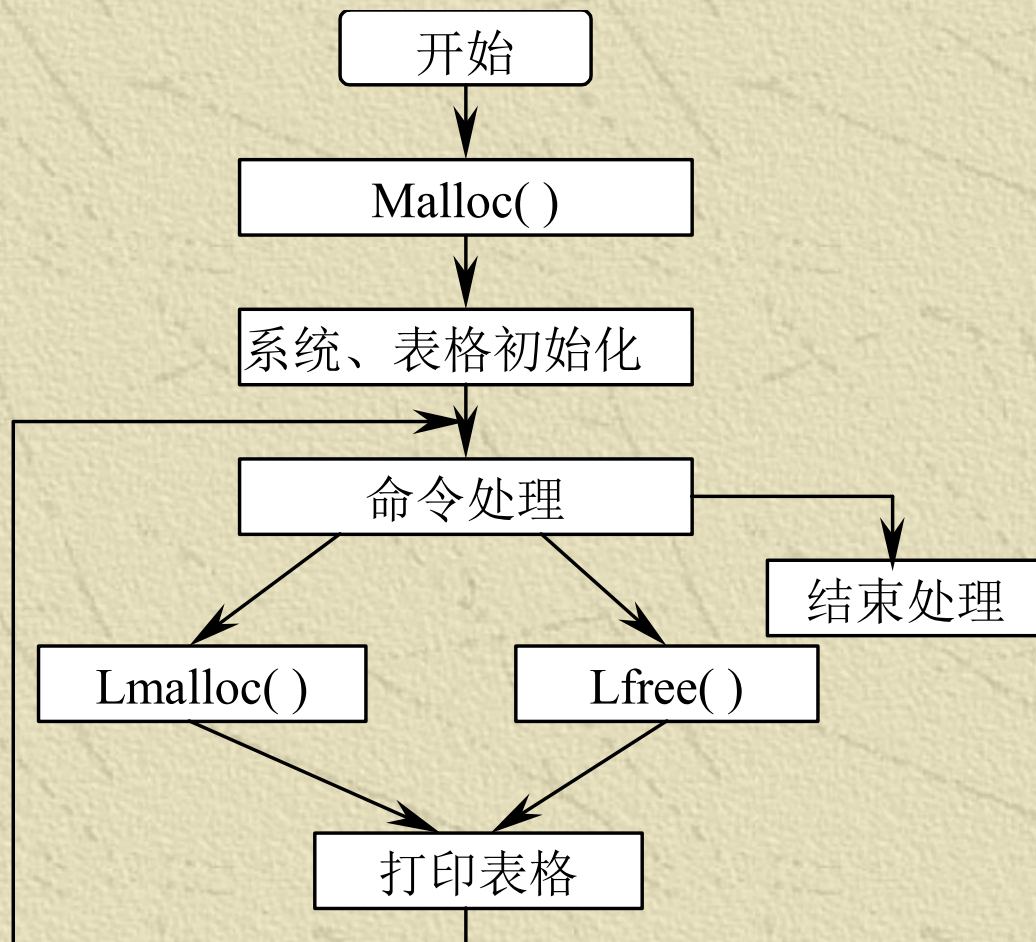


# 或双向链接表的形式:

---

```
struct map {  
    unsigned m_size;  
    char *m_addr;  
    struct map *next, *prior;  
};  
  
struct map *coremap;
```

# 整个系统的基本框架为：





## (四) 实验注意事项

键盘命令的简单形式如：

**m[alloc] 100**

通过lmalloc函数申请100字节的内存空间。

**f[ree] 100 2567899**

通过lfree函数释放起始地址为2567899的那个内存区，该地址先前通过lmalloc申请的有效存储区地址。

命令输入可用C函数

scanf ("%c", &cmdchar); cmdchar = getchar ( ) 或  
scanf ("%s", &cndstring)后跟  
scanf ("%u", &size) 或  
scanf ("%u %u", &size, &addr)。

如采用通用的命令串的形式，还要调用一个串匹配函数，以判断是哪个命令。



如采用简单的读取单个命令字符的形式，  
由于getchar或scanf可能读入的是前一个  
命令留在键盘输入缓冲区内的‘，’，

‘\t’，‘\n’等空白字符，为了正确地读入m，  
f等本次命令字符，需要预先滤去输入缓  
冲区的空白字符：

**do**

**c=getchar();**

**while(c=='\n' || c=='\t' || c==' ')**



# 程序输入、调试和测试

---

可以在Linux的环境下进行，也可在Windows或turbo C的环境中进行，但Windows和turbo C不支持I/O转向，故在完成此项工作时应将编译后产生的执行文件如myproc.exe存入磁盘的某个目录中，如是Windows，运行cmd命令进入DOS环境，运行程序，用I/O转向生成结果文件。如：

**myproc.exe > result.txt**

在Linux的环境下可直接运行：

**myproc > resul**



## （五） 测试要求

---

- ❖ 程序调试基本通过后，应进行较为全面的测试，建议采用白盒法的路径测试方法，测试路径应包含lmalloc的“循环”、“首次”，lfree的与邻近空闲分区联系的四种情况，还要包含一些必要的出错处理路径。
- ❖ 可在测试通过后，可用I/O转向使屏幕的输出结果转至指定文件。注意，由于此时看不到提示字符串和空闲存储区表的输出内容，欲释放存储地址较难正确输入，可采用以下几种方法：



✧ 测试通过后可用I/O转向使屏幕的输出结果转至指定文件。注意，由于此时看不到提示字符串和空闲存储区表的输出内容，欲释放存储地址较难正确输入，可采用以下几种方法：

✧ 1. 在不采用输出转向时先完整地运行一遍，并记下所有的命令和参数，再用I/O转向时“摸瞎”似地依次输入各个命令和参数，由于malloc所分配的存储区地址时用户程序的虚地址空间，故同一程序的两次运行的结果应当是一样的。为了使输出结果文件也含有输入的命令串，应在输入命令后，再将命令和参数输出至标准输出设备。



✧ 2. 如**malloc**和**free**，是采用相对地址的管理方法（**malloc**所分配的地址+偏移地址），那么输入**free**的释放地址就容易得多。

---

✧ 3. 输入也可采用**I/O**转向，将预先设计好的所有输入次序（部分数据可在不采用**I/O**转向运行时输入）写入输入文件，再用：

**myprog.exe < inputfile.txt > result.txt**

的形式执行。

✧ 4. 对于文件的**I/O**较熟悉的学生可采用每次在屏幕输出后重复向文件输出的方法，这样上述的问题就很容易解决了。



## (六) 实验数据和结果

---

学生最后应当将程序、输出结果文件和输入命令文件（如果有的话）合并成一个文件，如 `progres.txt` 上载到作业系统中。

在DOS中的合并方法是：

```
copy myprog.c+input.txt+result.txt progres.txt
```

在Linux中的合并方法是：

```
cat myprog.c input result > progres.txt
```

每一个程序的开头应当注上班级、姓名和学号。



## (七) 实验报告要求

实验报告包含以下几个方面：

1. 题目。
2. 算法思想和概要设计和重要模块的详细设计及功能和接口说明。
3. 重要数据结构和变量的说明。
4. 源程序、注释和结果。注释如在上机时来不及打入，可在写报告时补上。
5. 测试方法，对测试结果的分析，错误的分析。
6. 程序及测试的改进、本次上机经验及体会。
7. 交一份包括以上内容的打印的实验报告。