**存储管理**  
**1.物理内存**：

系统区：存放进程的PCB，能够确定PCB的存在，即PCB数组（数据结构由进程来定）

用户区：根据进程需要占用的内存空间分配内存给进程。

使用位图法来标记一个块是否被使用

假设内存只有64K，每一页有1K，系统区有10K，用于分配PCB，即根据PCB所占用的内存大小可知道能最多运行多少个进程，用户区有54K，即54页。进程M number这个数字是申请的地址空间大小，即申请的页数，申请后生成一个页表，页表的大小为申请空间的大小，页表左边是指令的编码，右边是存放在内存中的位置（用一个数组存储剩余的内存，把这个内存分配给这个页表），即10-64的其中一个页。

每个页可以存储多个指令，指令寻址时要先根据虚拟内存中的页表找到物理内存的表项，再根据页内地址进行寻址，每条指令占8个字节。当执行内存中完整的一个页的所有指令后，则把虚拟内存中为在物理内存中的指令存放到物理内存中，并更新页表。

**2.虚拟内存（外存）：**

功能：保存着用户输入的指令，保存每个程序的页表，当进程需要执行时，将指令转移到物理内存中，当指令从虚拟内存存到内存中，并更新页表，为CPU提供实际的物理地址（即在内存中的页表号，CPU根据页表号调出指令） 每次CPU需要读取指令时，都需要先来页表中查询到实际的物理块号，再根据块内地址查找到指令

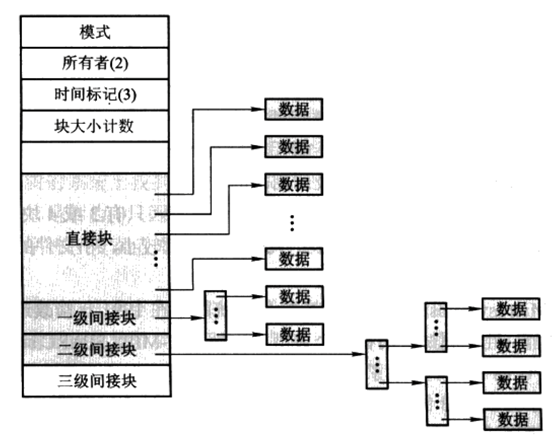
**3.磁盘：**

使用位图法来标记一个块是否被使用，需要提供当前文件的信息

使用一个索引表来标明某文件的索引块序号，系统可以通过文件的唯一标识号来查看整个目录

需要一个文件使用的索引表来标识当前文件是否在被使用

采用混合索引组织的方式对磁盘进行管理，类似下图所示，假设每个块的大小为64B，每个字符大小为1B，一个中文字符占2B，文件管理系统可以打开并查看某个文件的内容，也可以修改内容写回，一级所以最多索引8个文件块，即文件的大小最大为8\*64B



接口：

1. CPU需要时刻询问PCB中存放有什么进程，根据进程的PCB来决定调用哪个进程。给一个PCB的name，返回PCB结构
2. 创建一个进程时，CPU提供一个PCB块，根据PCB中需要分配的内存大小，在内存中分配空间
3. CPU告知PID来删除进程，把进程的PCB和用户数据区释放，更新内存的位图
4. 返回CPU一个进程指令的初始的地址，CPU可以根据地址找到物理内存中的指令，并对指令进行解析。（若每条指令大小为8个字节，则每次执行完一条指令后PC+8）
5. 文件管理系统可以通过传输文件的唯一标记来实现读取文件，修改文件后可以重新写回文件。
6. 创建文件时，需要文件管理系统提供文件的唯一标识和占用的存储大小，磁盘管理再根据需要占用的大小来分配是直接块还是索引块
7. 文件系统需要访问某个文件时，输入一个文件的唯一标识符，可以给设备管理返回一个文件或者磁盘是否在被使用的信号。

函数：

物理内存中：