

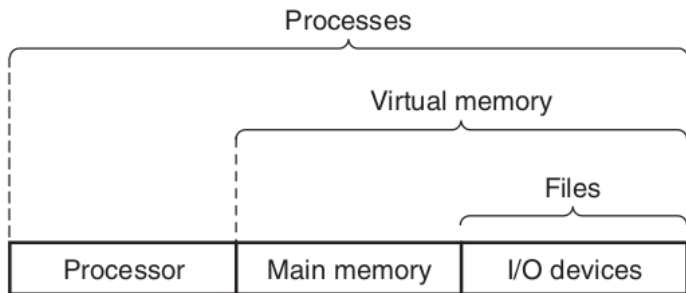
文件系统

May 2, 2012

操作系统中三个最重要的概念

1. 把处理器CPU抽象为进程
2. 把物理内存抽象为虚拟内存和虚拟地址空间
3. 把永久存储介质（如硬盘）抽象为文件

操作系统中三个最重要的概念



文件的类型

1. 普通文件(.doc, .c, .cpp, .pdf, .exe)
2. 目录
3. 字符设备文件(第5章)
4. 块设备文件(第5章)

文件的存取方法

1. 顺序存取(**sequential access**): 只能按前后顺序访问每个字节, 不能随机访问。例如磁带上的文件系统
2. 随机存取(**random access**): 可以按照任意顺序访问文件的各个字节, 例如磁盘文件系统

文件的属性

- ▶ protection
- ▶ password
- ▶ creator
- ▶ owner
- ▶ read-only flag
- ▶ lock flag
- ▶ creation time
- ▶ time of last access
- ▶ current size

文件的属性

```
struct stat {  
    unsigned long    st_dev;  
    unsigned long    st_ino;  
    unsigned short   st_mode;  
    unsigned short   st_nlink;  
    unsigned short   st_uid;  
    unsigned short   st_gid;  
    unsigned long    st_rdev;  
    unsigned long    st_size;  
    unsigned long    st_blksize;  
    unsigned long    st_blocks;  
    unsigned long    st_atime;  
    unsigned long    st_mtime;  
    unsigned long    st_ctime;  
};
```

针对文件的操作

- ▶ create
- ▶ delete
- ▶ open
- ▶ close
- ▶ read
- ▶ write
- ▶ append
- ▶ seek (仅限可随机访问的文件)
- ▶ get attributes
- ▶ set attributes
- ▶ rename

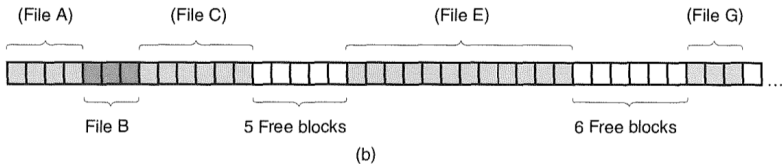
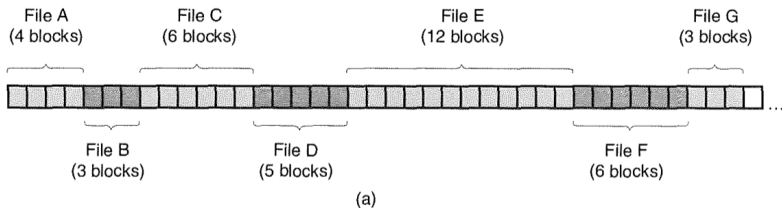
针对目录的操作

- ▶ create
- ▶ delete
- ▶ opendir
- ▶ closedir
- ▶ readdir
- ▶ rename
- ▶ link (图示及命令ln)
- ▶ unlink (图示, 命令?)

文件的实现

核心问题是记录每个文件在存储介质（如磁盘）上存放的位置。

文件的实现: 连续存储方式



文件的实现: 连续存储方式

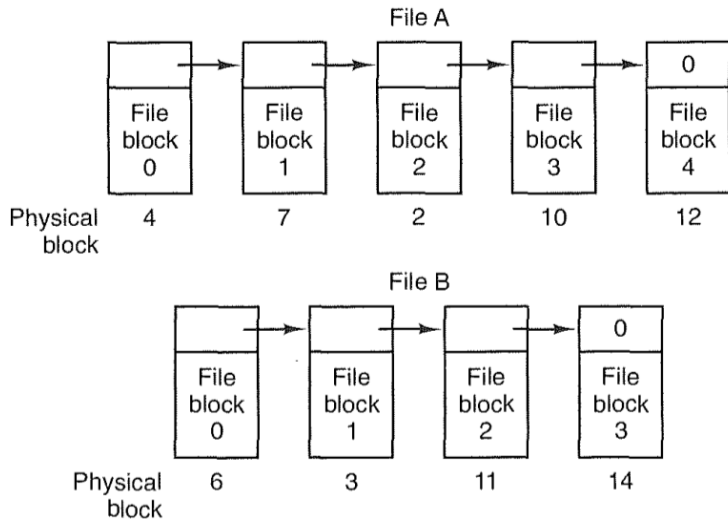
连续存储方式有两个重要优点:

- ▶ 实现容易: 只需记录每个文件在磁盘上起始块地址, 以及所占的总块数
- ▶ 读写速度非常快: 每次读写仅需1次机械移动寻址过程

缺点? — 容易导致文件系统大量碎片(前图)

应用? — CD-ROM上的文件系统, why?

文件的实现: 线性链表存储方式



注: 每个磁盘块的前几个字节用于记录存储该文件的下一个磁盘块地址

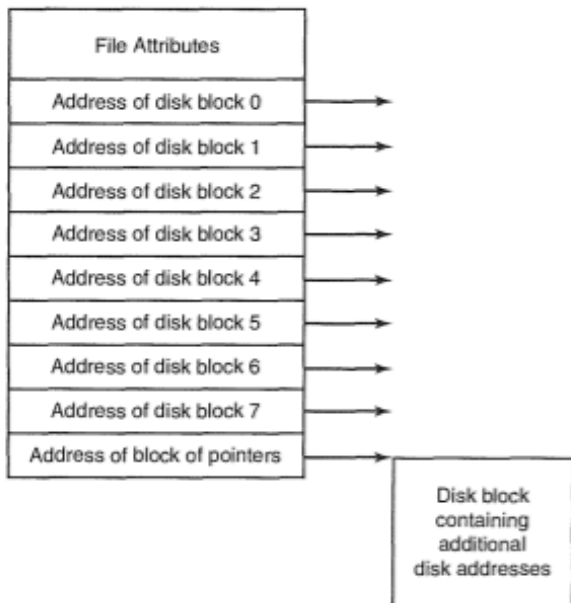
文件的实现: 线性链表存储方式

线性链表存储方式的优点:

- ▶ 只需记录每个文件在磁盘上的起始地址
- ▶ 避免碎片问题

缺点: 不利于随机访问, why?

文件的实现: I-nodes



文件的实现: I-nodes

结合命令`ln`以及`ls -li`理解i-node实现方法
目录的实现方法?

文件系统的布局

考虑磁盘上的文件系统:

- ▶ 磁盘可以进行分区, 用磁盘分区表记录每个分区的边界
- ▶ 第0扇区: **MBR(Master Boot Record, 主引导记录)**, 存放引导程序及磁盘分区表
- ▶ **MBR**中的引导程序选择某个分区中的操作系统进行引导
- ▶ 每个分区中有一个磁盘块用于启动该分区上的操作系统(**boot block**)
- ▶ 超级块(**superblock**): 存放该分区上的文件系统参数(类型、块数等信息)

文件系统的布局

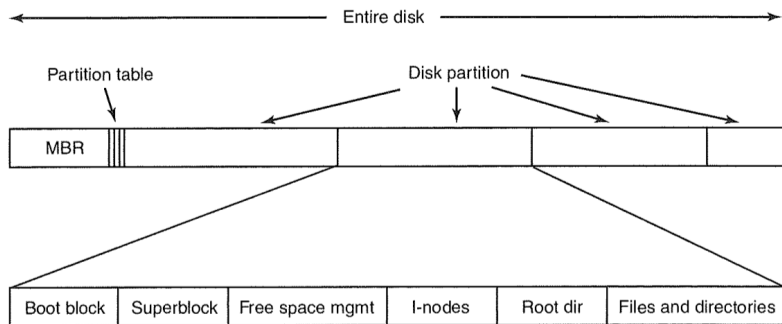


Figure 4-9. A possible file system layout.