计算机导论

第五章 操作系统简介

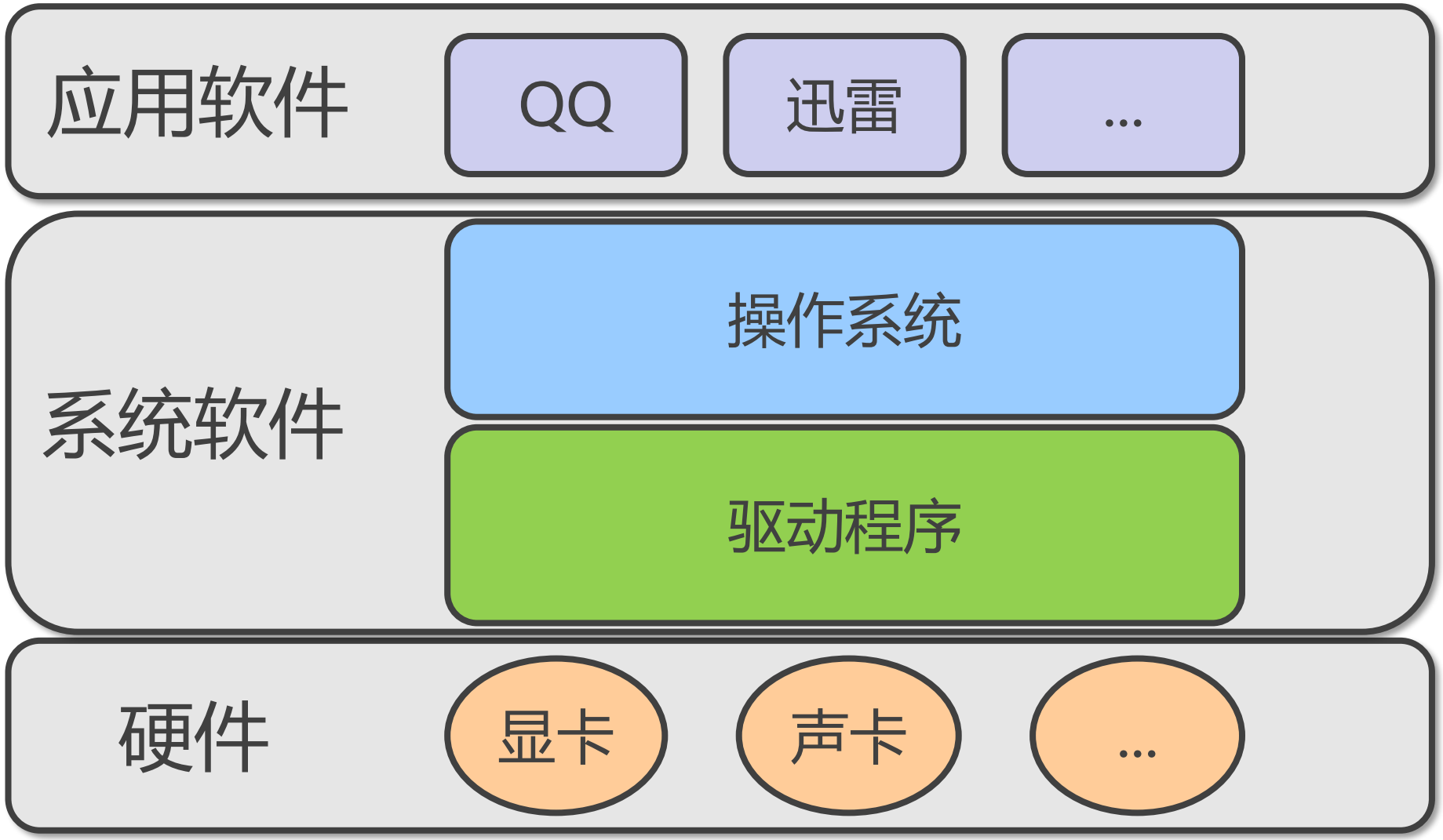
本章内容

- 认识操作系统
- 操作系统分类
- 操作系统对硬件的管理
- 文件系统

本章内容

- 认识操作系统
- 操作系统分类
- 操作系统对硬件的管理
- 文件系统

计算机系统的层次



计算机的启动

- BIOS是一组程序，包括基本输入输出程序、系统设置信息、开机后自检程序和系统自启动程序。
- 这些程序都被固化到了计算机主板的ROM芯片上。用户可以对BIOS进行设置。
- 计算机的启动过程

1. 启动自检阶段
2. 初始化启动阶段
3. 启动加载阶段
4. 内核装载阶段
5. 登录阶段

操作系统

- 操作系统(OS, Operating System)：是控制和管理计算机系统内各种硬件和软件资源、合理有效地组织计算机系统的工作，为用户提供一个使用方便可扩展的工作环境，从而起到连接计算机和用户的接口作用。


Manage memory


Manage processor resources


Keep track of storage resources


Ensure that input and output proceed in an orderly manner


Establish basic elements of the user interface

本章内容

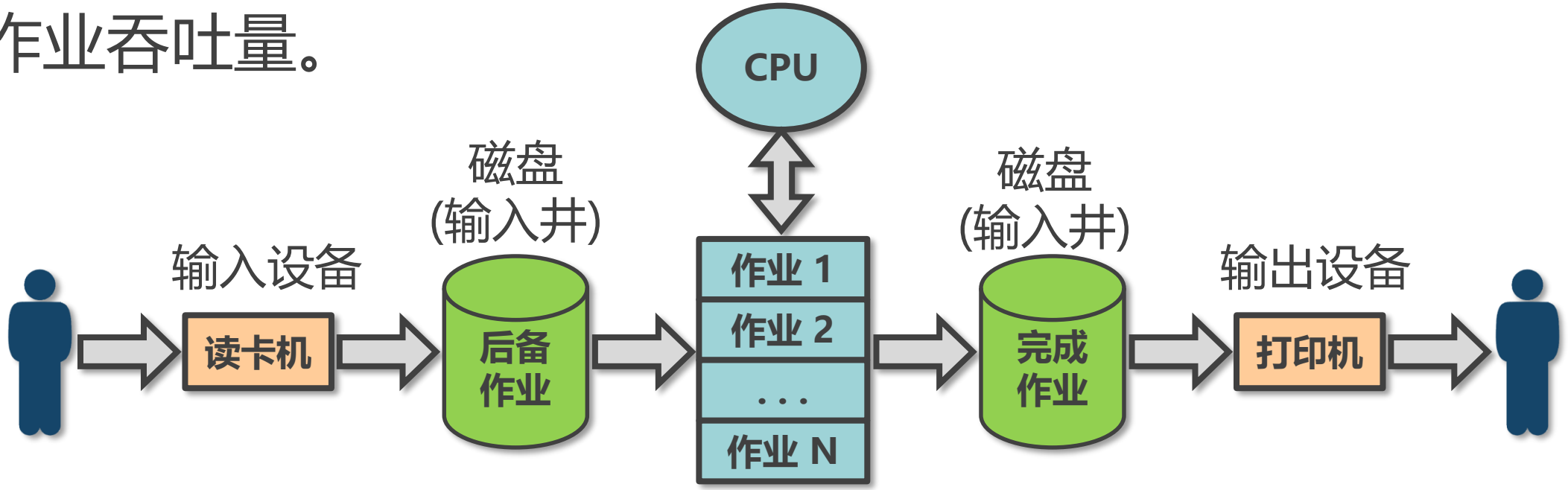
- 认识操作系统
- 操作系统分类
- 操作系统对硬件的管理
- 文件系统

操作系统分类

- 批处理操作系统
- 分时操作系统
- 实时操作系统
- 嵌入式操作系统

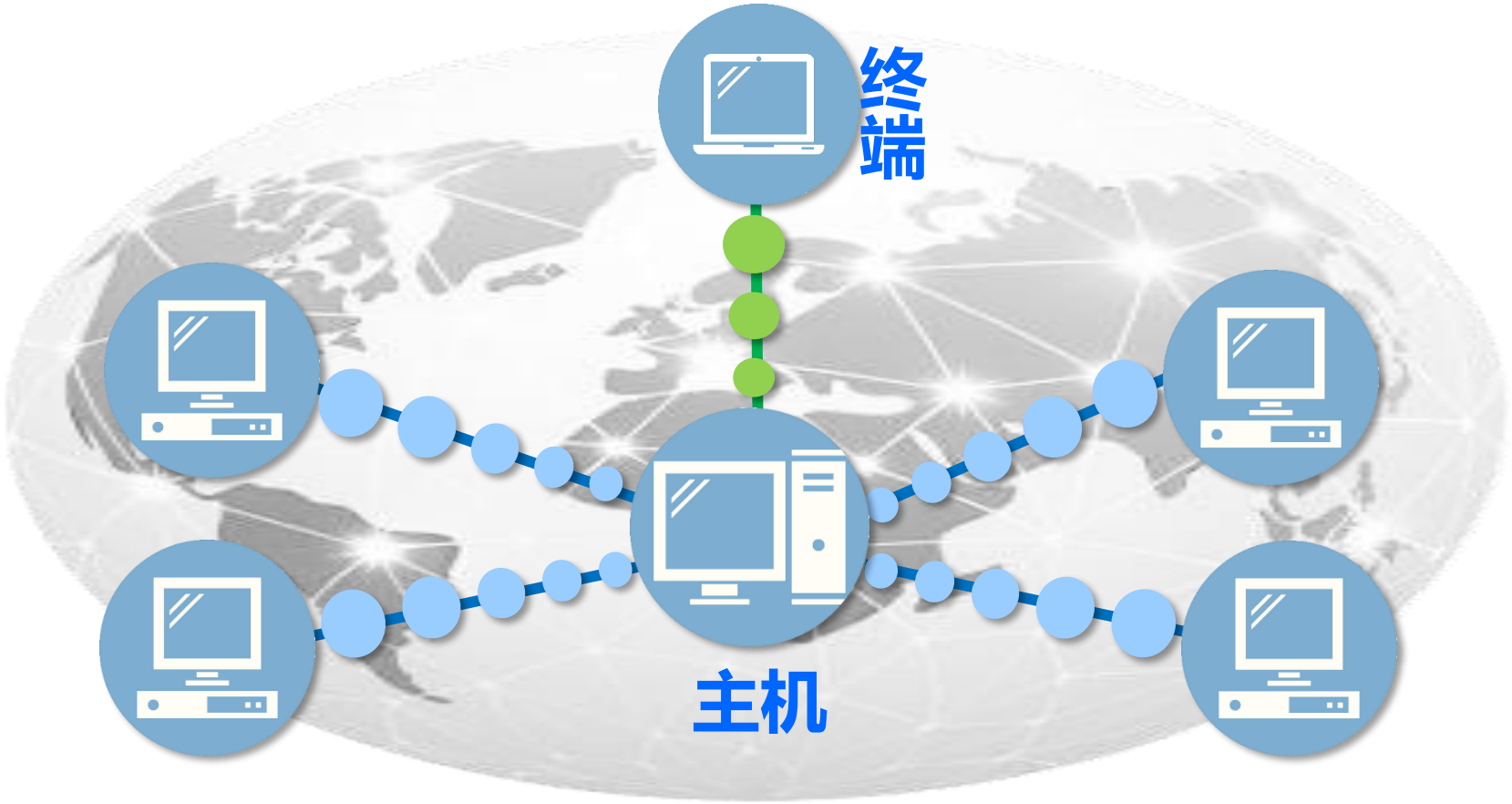
批处理操作系统

- 批处理操作系统——批处理是指计算机系统对一批作业自动进行处理的技术。 由于系统资源为多个作业所共享，其工作方式是作业之间自动调度执行。并在运行过程中用户不干预自己的作业，从而大大提高了系统资源的利用率和作业吞吐量。



分时操作系统

- 分时操作系统——将计算机系统的CPU时间划分成一些小的时间片，按时间片轮流把处理机分给各联机作业使用。



分时操作系统的特点

- 交互性：用户与系统进行人机对话。
- 多路性：多用户同时在各自终端上使用同一CPU。
- 独立性：用户可彼此独立操作，互不干扰，互不混淆。
- 及时性：用户在短时间内可得到系统的及时回答。

实时操作系统

- 实时操作系统——所谓"实时",即"及时",是指系统能及时(或即时)响应外部 事件的请求,在规定的时间内完成对该事件的处理，并控制所有实时任务协调一致地运行。它必须保证实时性和高可靠性，对系统的效率则放在第二位。
- 主要应用于工业控制、军事控制、电子设备等领域。

嵌入式操作系统

- 嵌入式操作系统
  - 嵌入式操作系统通常包括与硬件相关的底层驱动软件、系统内核、设备驱动接口、通信协议、图形界面、标准化浏览器等。
  - 嵌入式操作系统负责嵌入式系统的全部软、硬件资源的分配、任务调度，控制、协调并发活动。它必须体现其所在系统的特征，能够通过装卸某些模块来达到系统所要求的功能。
  - 目前在嵌入式领域广泛使用的操作系统有：嵌入式Linux、Windows Embedded、VxWorks等，以及应用在智能手机和平板电脑的Android、iOS等。

主流操作系统

UNIX

本章内容

- 认识操作系统
- 操作系统分类
- 操作系统对硬件的管理
- 文件系统

硬件的管理

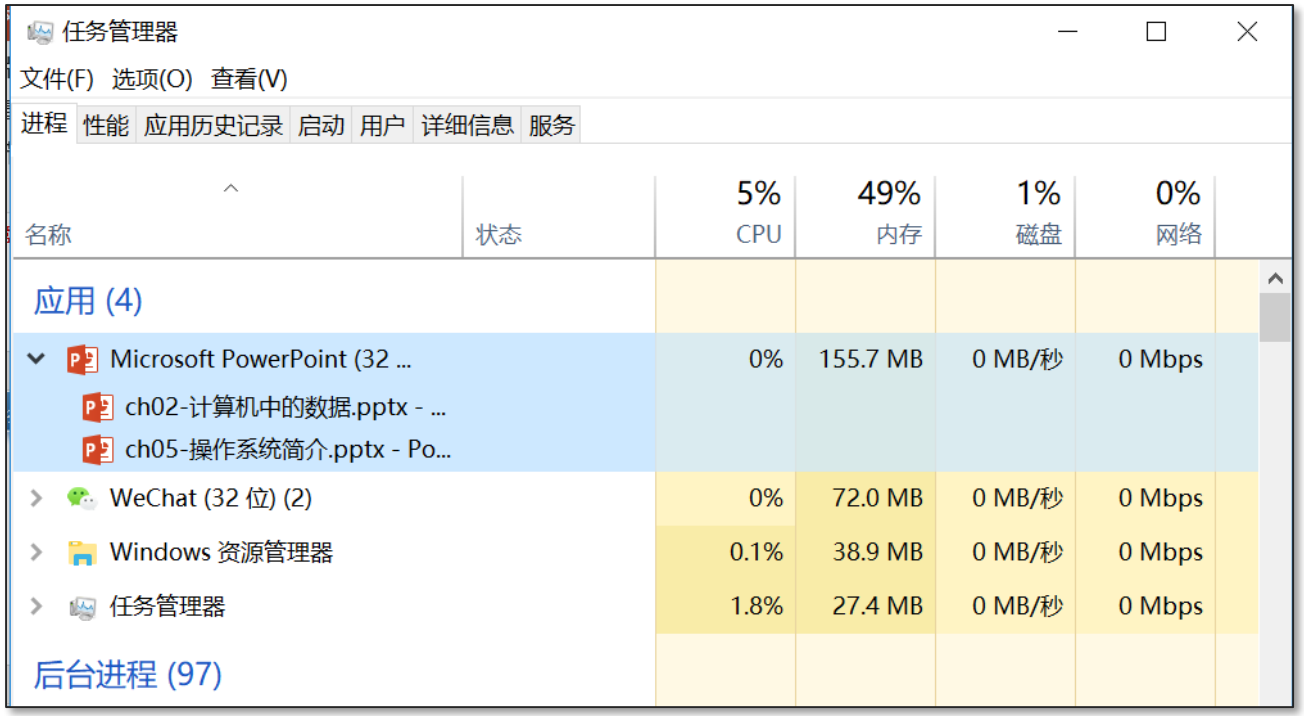
- 处理器的管理
- 存储器管理

CPU管理

- 为了满足系统的性能要求，提高任务处理的效率，现在主流的计算机通常都有一个或多个CPU，每个CPU中又有多个核。
- 然而核的数量是远远小于需要执行的程序的数量。
- 多任务在同一个核上进行执行
  - 分时间片
  - 中断——进程
  - 调度——调度策略

进程

- 在多道程序设计的环境下，为了描述程序在计算机系统内的执行情况，必须引入新的概念——进程。
- 进程：是一个程序的一次执行，包含了其执行时所有的环境信息。
- 进程是执行中的程序。



进程的特征

- 进程的特征（与程序比较）
  1. 结构特征

     进程控制块(PCB) + 程序 + 数据 = 进程实体
  2. 动态性——最基本特征

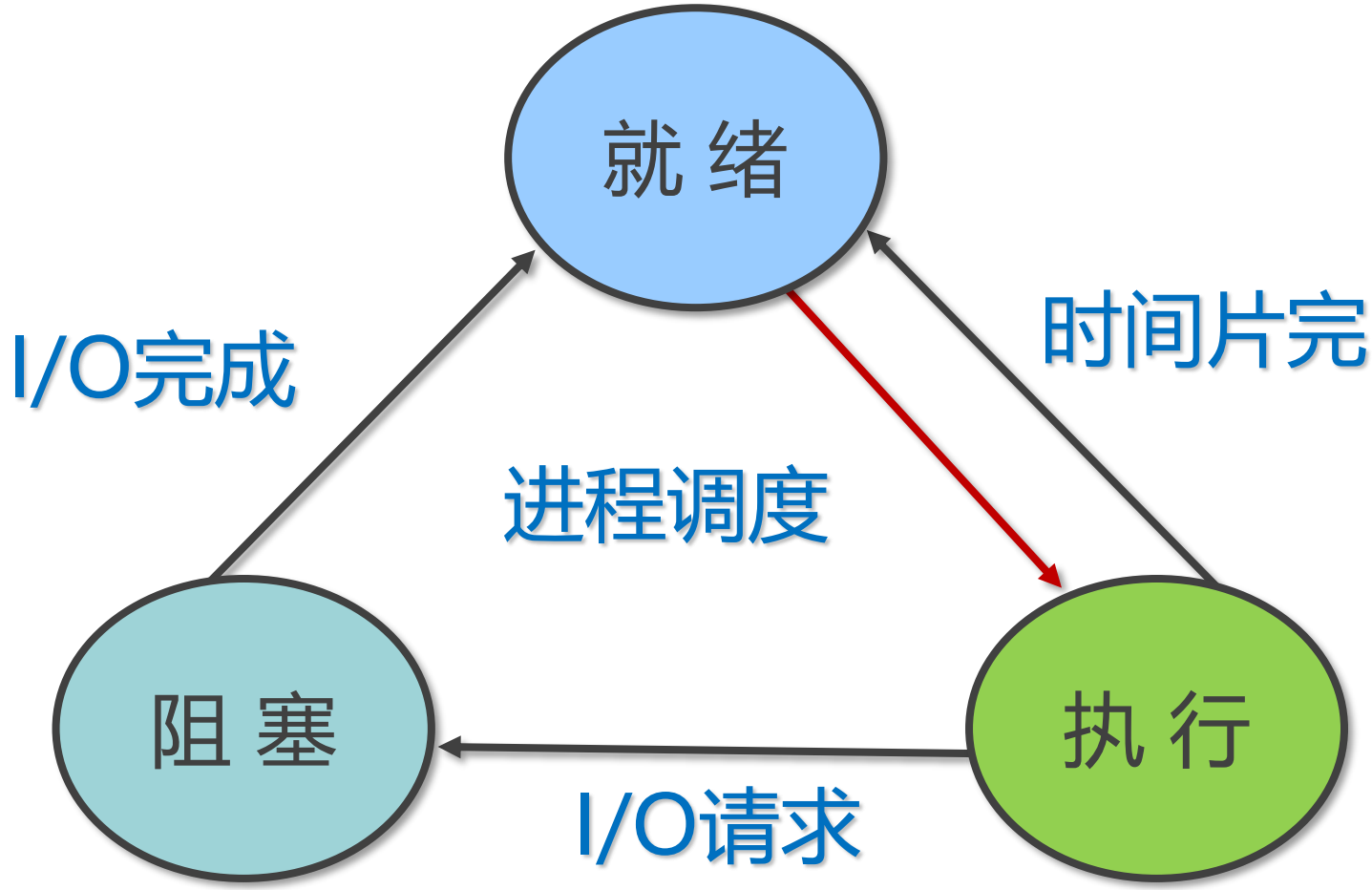
     进程：进程实体的一次执行过程，有生命周期。

     程序：程序是一组有序指令的集合，是静态的概念。
  3. 并发性
  4. 独立性
  5. 异步性

进程的三种状态

- 进程的三种状态
  - 就绪状态(Ready)：进程已获得除处理机之外的所有必需的资源，一旦得到处理机控制权，立即可以运行。
  - 运行状态(Running)：进程已获得运行所必需的资源，它的程序正在处理机上执行。
  - 阻塞状态(Blocked)：正在执行的进程由于发生某事件而暂时无法执行时，便放弃处理机而处于暂停状态，称该进程处于阻塞状态或等待状态。
- 就绪队列与阻塞队列

进程三种状态的转换

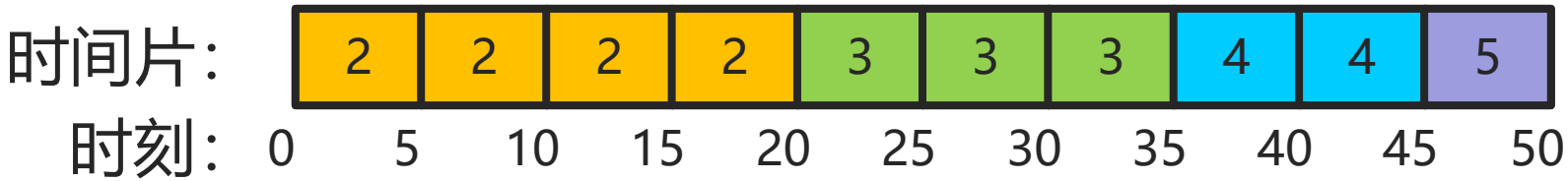


进程调度

- 调度程序，按照调度策略，动态地把CPU分配给处于就绪队列中的进程，并将该进程从就绪态转换到运行状态。
- 对于不同的系统和系统目标，通常采用不同的调度算法。衡量调度策略的好坏，一个重要的指标是：
  - 周转时间（平均周转时间）
- 简单介绍两种进程调度策略
  - 先来先服务调度算法（FCFS）
  - 短任务优先调度算法（SJF）

先来先服务调度的例子

- 先来先服务调度



时间片：5个时间单位

进程	到达时间	执行时间	开始时间	结束时间	周转时间
2	0	20	0	20	20
3	0	15	20	35	35
4	4	10	35	45	41
5	5	5	45	50	45

平均周转时间：35.25

先来先服务调度算法的特点

- 先来先服务调度（FCFS）调度算法的特点：

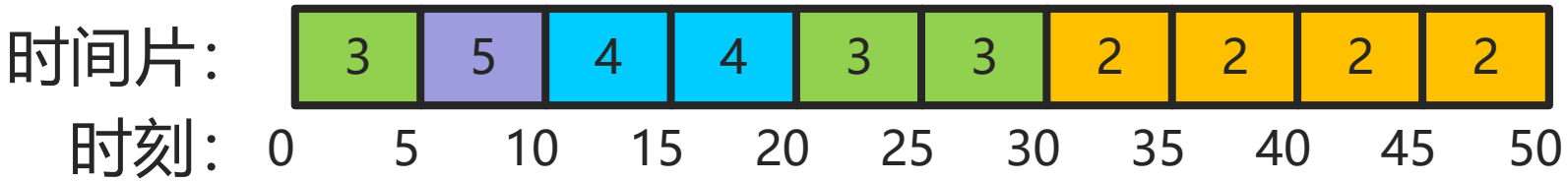
FCFS调度算法有利于CPU繁忙型的作业，而不利于I/O繁忙型的作业（进程）

– CPU繁忙型作业：如通常的科学计算。

– I/O繁忙型作业：指CPU进行处理时，需频繁的请求I/O。

短作业优先调度的例子

- 短作业优先调度



时间片：5个时间单位

进程	到达时间	执行时间	开始时间	结束时间	周转时间
2	0	20	30	50	50
3	0	15	0	30	30
4	4	10	10	20	16
5	5	5	5	10	5

平均周转时间：25.25

短作业优先调度算法的特点

- 短作业优先调度（SJF）调度算法的优缺点：

  优点：有效降低作业的平均等待时间，提高系统吞吐量。

  缺点：

  - 对长作业不利。
  - 该算法完全未考虑作业的紧迫程度，因而不能保证紧迫性作业（进程）会被及时处理。
  - 由于作业（进程）的长短含主观因素，不一定能真正做到短作业优先。

硬件的管理

- 处理器的管理
- 存储器管理

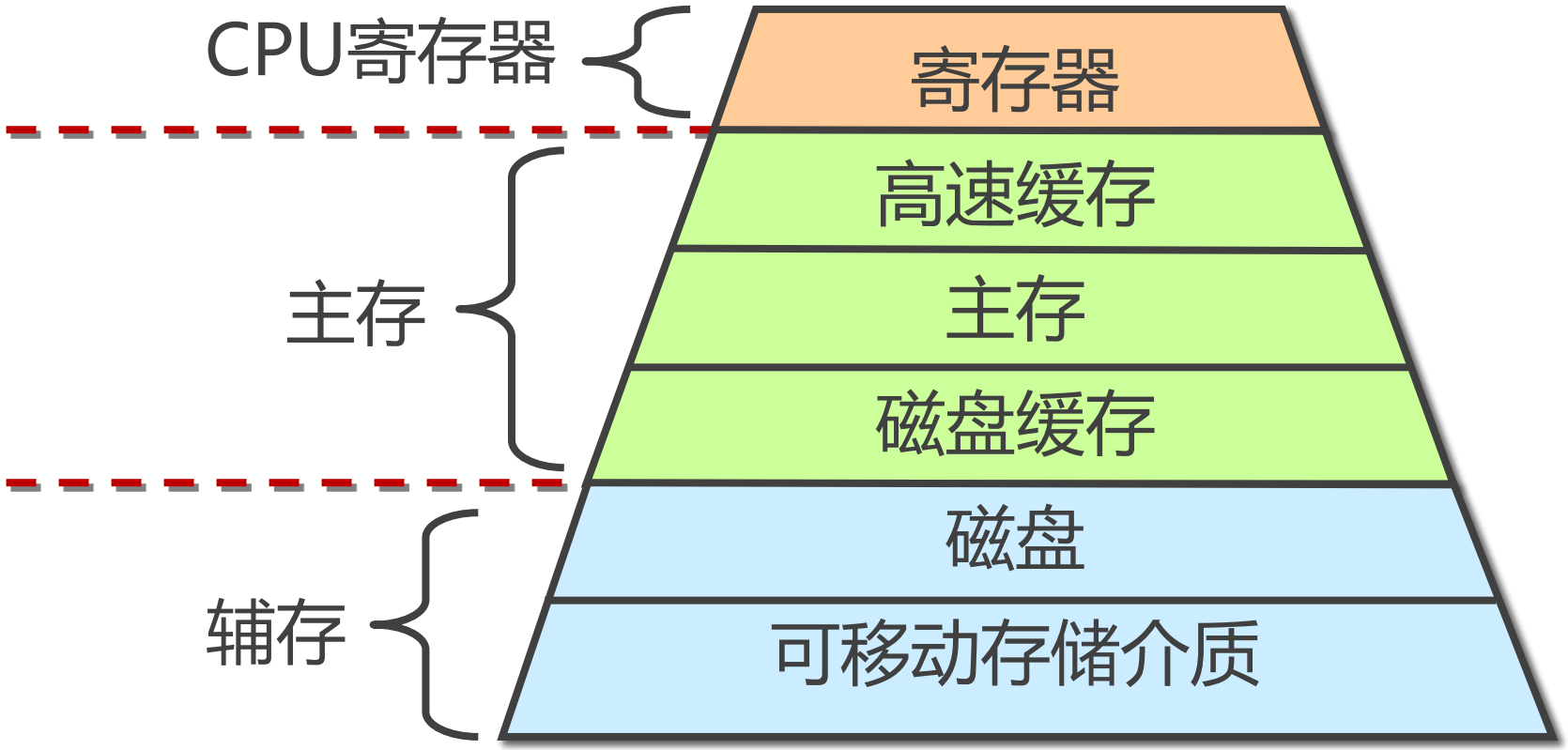
存储器管理

存储器是计算机系统的重要组成部分之一。随着计算机技术的发展，存储器容量一直在扩充，但仍不能满足现代软件和用户的需要，因此存储器仍是一种宝贵、紧俏的资源。对存储器加以有效管理，不仅直接影响存储器的利用率，而且对系统性能有重大影响。

存储器管理的主要对象是内存，对外存的管理在文件管理中。

存储器的层次结构

- 多级存储器结构



存储器的层次结构

- 主存储器与寄存器
  - 主存储器：是计算机硬件的一个重要部件，其作用是存放指令和数据，并能由CPU直接随机存取。
  - 寄存器：访问速度最快。
- 高速缓存和磁盘缓存
  - 高速缓存：访问速度快于主存储器。
  - 磁盘缓存：利用主存中的存储空间。

程序的装入

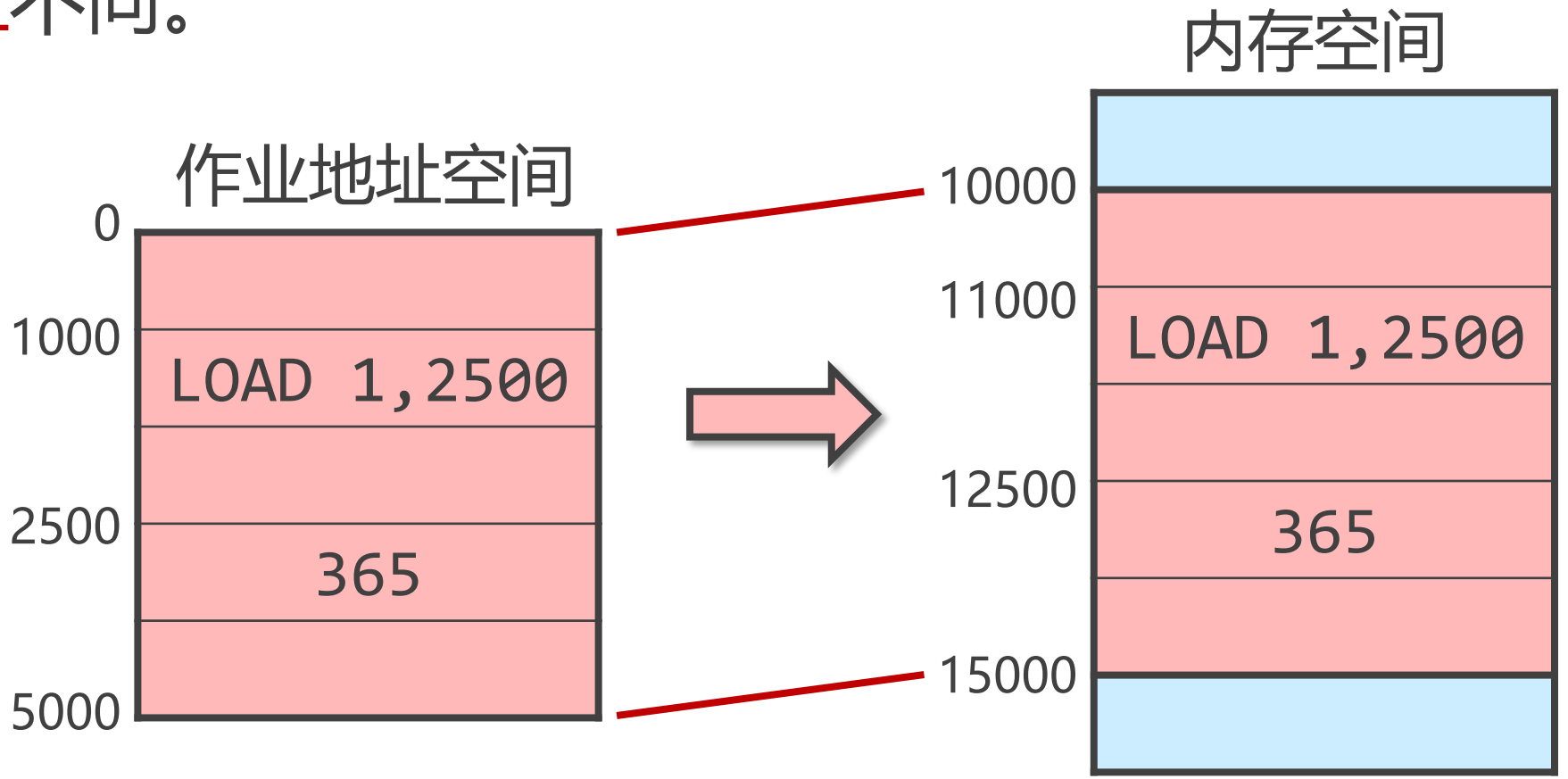
在多道程序环境下，要使程序运行，必须为之先建立进程。创建进程的第一件事是将程序和数据装入内存。

绝对装入方式只能将目标模块装入到内存中事先指定的位置。在多道程序环境下，编译程序不能预知所编译的目标模块应放在内存何处，因此绝对装入方式只适用于单道程序环境下。

在多道程序环境下，目标模块的起始地址通常从0开始，程序中的其他地址都是相对于起始地址计算的。因此应采用可重定位装入方式，根据内存的当前情况，将装入模块装入到内存的适当位置。

程序的装入

- 注意：在采用可重定位装入方式将装入模块装入内存后，会使装入模块中的所有逻辑地址与实际装入内存的物理地址不同。

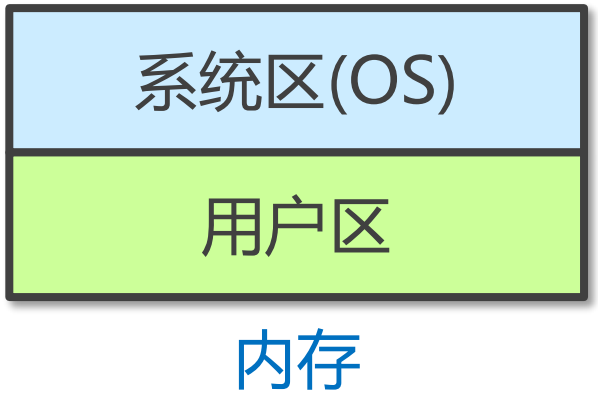


连续分配方式

- 连续分配方式：是指为一个用户程序分配一个连续的内存空间。
- 分类：
  - 单一连续分配
  - 固定分区分配
  - 动态分区分配
  - 可重定位分区分配

单一连续分配方式

- 最简单的一种存储管理方式，但只能用于单用户、单任务的操作系统中。
- 采用这种存储管理方式时，可把内存分为系统区和用户区两部分，系统区仅提供给OS使用，通常放在内存低址部分，用户区是指除系统区以外的全部内存空间，提供给用户使用。



固定分区分配

1. 原理

将内存用户空间划分为若干个固定大小的区域，在每个分区中只装入一道作业，这样把用户空间划分为几个分区，便允许有几道作业并发执行。当有一空闲分区时，便可以再从外存的后备作业队列中，选择一个适当大小的作业装入该分区，当该作业结束时，可再从后备作业队列中找出另一作业调入该分区。

固定分区分配

2. 划分分区的方法

可用两种方法将内存的用户空间划分为若干个固定大小的分区：

① 分区大小相等：缺乏灵活性，用于一台计算机控制多个相同对象的场合

② 分区大小不等：把内存区划分成含有多个较小的分区、适量的中等分区及少量的大分区，可根据程序的大小为之分配适当的分区。

固定分区分配

3. 实现

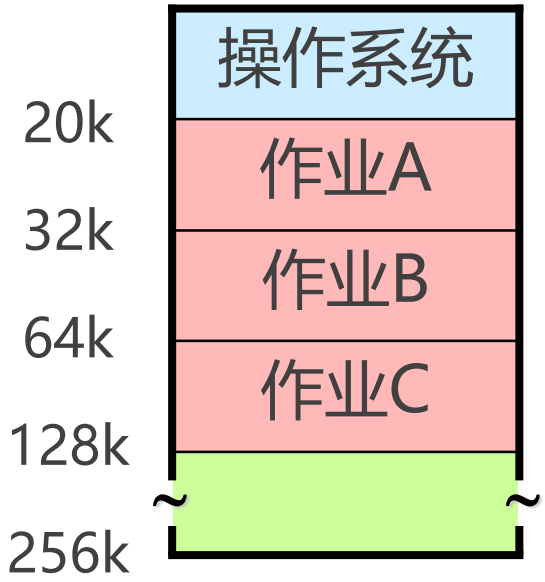
为便于内存分配，通常将分区按大小进行排队，并为之建立一张分区使用表，其中各表项包括每个分区的起始地址、大小及状态(是否已分配)。

当有一用户程序要装入时，由内存分配程序检索该表，从中找出一个能满足要求的、尚未分配的分区，将之分配给该程序，然后将该表项中的状态置为"已分配"；若未找到大小足够的分区，则拒绝为该用户程序分配内存。

固定分区分配

分区号	大小(K)	起址(K)	状态
1	12	20	已分配
2	32	32	已分配
3	64	64	已分配
4	128	128	未分配

分区说明表



存储空间分配情况

动态分区分配

1. 原理

动态分区分配是根据进程的实际需要，动态地为之分配内存空间。作业装入内存时，把可用内存分出一个连续区域给作业，且分区的大小正好适合作业大小的需要。分区的大小和个数依装入作业的需要而定。

2. 实现

在实现过程中涉及如下问题：

- 分区分配中的数据结构
- 分区分配算法
- 分区分配及回收操作

动态分区分配

① 分区分配中的数据结构

- 空闲分区表示
  - 空闲分区表：记录每个空闲分区的情况。每个空闲分区占一个表目。表目中包括：分区序号、分区始址、分区的大小等。
  - 空闲分区链：在每个分区的起始部分，设置一些用于控制分区分配的信息，以及用于链接各分区所用的前向指针；在分区尾部则设置一后向指针，在分区末尾重复设置状态位和分区大小表目。
- 已占分区说明表
  - 结构：作业号；起始地址；大小

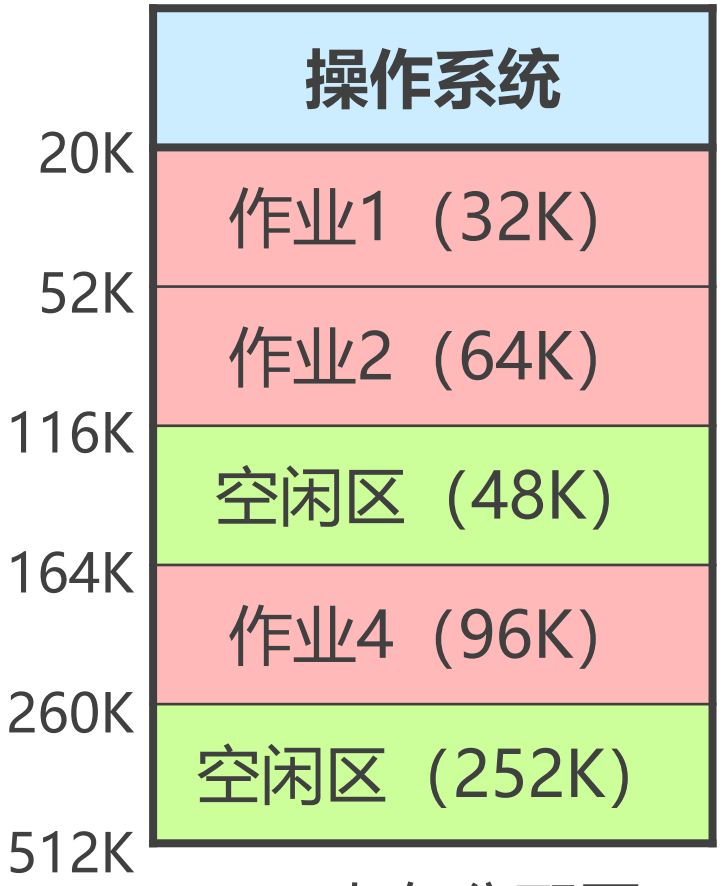
动态分区分配

序号	大小(K)	起址(K)	状态
1	48	116	可用
2	252	260	可用

（a）空闲分区表

作业号	大小(K)	起址(K)
1	32	20
2	64	52
4	96	164

（b）已占分区表



（c）内存分配图

动态分区分配

② 分区分配算法

为把一个新作业装入内存，需按照一定的分配算法，从空闲分区表或空闲分区链中选出一分区分配给该作业。

常用的分配算法：

- 首次适应算法FF
- 循环首次适应算法
- 最佳适应算法

动态分区分配

- **首次适应算法FF**

FF算法要求空闲分区表以地址递增的次序排列。在分配内存时，从表首开始顺序查找，直至找到一个大小能满足要求的空闲分区为止；然后按照作业的大小，从该分区中划出一块内存空间分配给请求者，余下的空闲分区仍留在空闲分区表中。若从头到尾不存在满足要求的分区，则分配失败。

优点：优先利用内存低址部分的内存空间。

缺点：低址部分不断划分，产生小碎片（内存碎块、内存碎片、零头）；每次查找从低址部分开始，增加了查找的开销。

动态分区分配

- **循环首次适应算法**

在分配内存空间时，从上次找到的空闲分区的下一个空闲分区开始查找，直到找到一个能满足要求的空闲分区，从中划出一块与请求大小相等的内存空间分配给作业。

为实现算法，需要：

> 设置一起始查寻指针

> 采用循环查找方式

优点：使内存空闲分区分布均匀，减少查找的开销。

缺点：缺乏大的空闲分区。

动态分区分配

- **最佳适应算法**

所谓“最佳”是指每次为作业分配内存时，总是把能满足要求、又是最小的空闲分区分配给作业，避免“大材小用”。

要求将所有的空闲分区按其容量以从小到大的顺序形成一空闲分区链。

缺点：产生许多难以利用的小空闲区。

动态分区分配

② 分区分配及回收操作

- 分配内存

利用某种分配算法，从空闲分区链(表)中找到所需大小的分区。设请求的分区大小为**u.size**，表中每个空闲分区的大小表示为**m.size**，若**m.size - u.size ≤ size**(规定的不再切割的分区大小)，将整个分区分配给请求者，否则从分区中按请求的大小划出一块内存空间分配出去，余下部分留在空闲链中，将分配区首址返回给调用者。

动态分区分配

- 回收内存

当进程运行完毕释放内存时，系统根据回收区首址，在空闲分区链(表)中找到相应插入点，此时可能有四种情况：

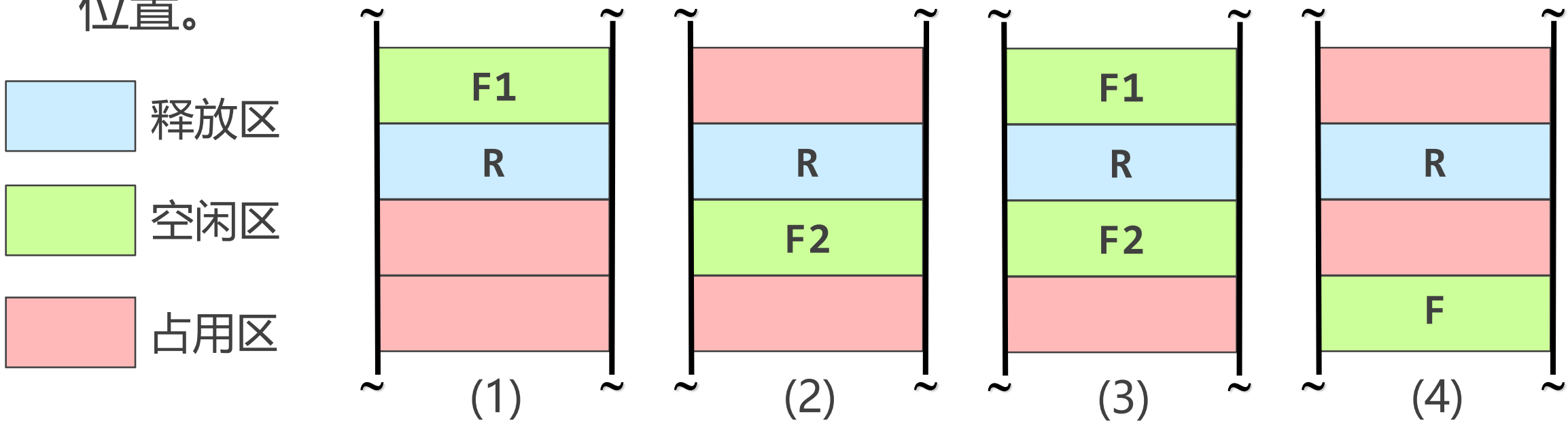
(1) 回收区与插入点的前一个分区F1邻接：将回收区与F1合并，修改F1的表项的分区大小。

(2) 回收区与插入点的后一个分区F2邻接：将回收区与F2合并，修改F2的表项的首址、分区大小。

动态分区分配

(3) 回收区与插入点的前后两个分区F1、F2邻接：将三个分区合并，使用F1的表项和F1的首址，取消F2的表项，大小为三者之和。

(4) 回收区既不与F1邻接，又不与F2邻接：为回收区单独建立新表项，填写回收区的首址与大小，根据其首址插到空闲链中的适当位置。



本章内容

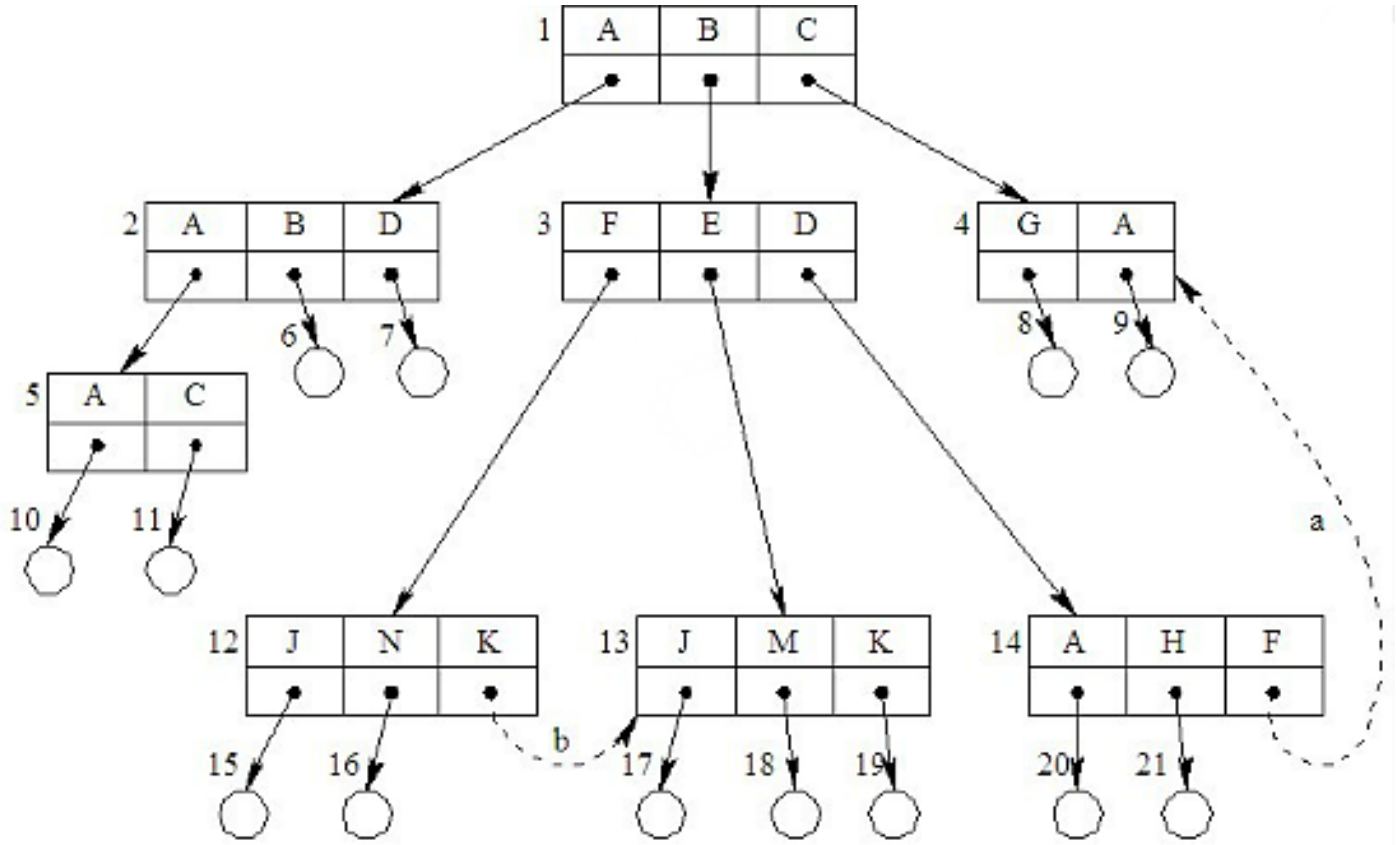
- 认识操作系统
- 操作系统分类
- 操作系统对硬件的管理
- 文件系统

文件系统

- 在现代计算机系统中，要用到大量的程序和数据，由于内存容量有限，且不能长期保存，故而平时总是把他们以文件的形式存放在外存中，需要时调入内存。
- 但用户不能够胜任管理文件的工作，于是在OS中又增加了文件管理功能，构成一个文件系统，负责管理在外存上的文件。
- 文件夹：计算机磁盘空间里面为了分类储存电子文件而建立独立路径的目录。

目录树结构

- 文件目录用于标识系统中的文件及其物理地址，供检索时使用。对目录管理的要求如下：
  - 实现 "按名存取"
  - 提高对目录的检索速度
  - 文件共享
  - 允许文件重名



目录查询技术

- 当用户要访问一个已存文件时，系统首先利用用户提供的文件名对目录进行查询，找出该文件控制块或对应索引结点；然后根据FCB或索引结点中所记录的文件物理地址，换算出文件在磁盘上的物理位置；最后通过磁盘驱动程序，将所需文件读入内存。

Questions?