## 操作系统

操作系统是计算机系统中的核心系统软件，负责管理和控制计算机系统中硬件和软件资源，合理组织计算机工作流程和有效利用资源，在计算机与用户之间起接口的作用。

操作系统的类型（依据使用环境和对作业的处理方式）分为批处理、分时、实时、网络、分布式、嵌入式和微内核等。

1. **批处理操作系统（Batch Processing Operating System，BPOS）**：用户将一批作业提交给操作系统后就不再干预，由操作系统控制它们自动运行。批处理操作系统不具有交互性，它是为了提高CPU的利用率而提出的一种操作系统。系统资源的利用率和作业的吞吐量是多道批处理操作系统的主要设计目标，同时也要兼顾作业的周转时间。

多道批处理系统有两个特点：

* 多道：系统内可同时容纳多个作业。
* 成批：系统能成批地自动运行多个作业，在运行过程中不允许用户与其作业发生交互作用。

1. **分时操作系统（Time Share Operating System，TSOS）**：采用分时技术，将处理机时间与内存空间按一定的时间间隔，轮流地切换给各终端用户的程序使用。由于时间间隔很短，每个用户的感觉就像他独占计算机一样。例如UNIX系统就采用剥夺式动态优先的CPU调度，有力地支持分时操作。特征有交互性、多用户同时性、独立性和及时性等。
2. **实时操作系统（Real Time Operating System，RTOS）**：保证在一定时间限制内完成特定功能的操作系统。实时操作系统有硬实时和软实时之分，硬实时要求在规定的时间内必须完成操作，这是在操作系统设计时保证的；软实时则只要按照任务的优先级，尽可能快地完成操作即可。实时系统包括实时过程控制系统（生产过程控制等）和实时信息处理系统（实验数据采集等）。并不强调资源利用率，更关心及时性、可靠性和完整性。
3. **网络操作系统（Network Operating System，NOS）**：网络操作系统是使网络上各计算机能方便而有效地共享网络资源以及为网络用户提供所需的各种服务的软件和有关规程的集合。它提供的服务：文件服务、打印服务、数据库服务、通信服务、信息服务、分布式服务、名字服务、网络管理服务、Internet与Intranet服务等。我们常见的Windows Server2012就属于网络操作系统。
4. **分布式操作系统（Distributed Operating System，DOS）**：分布式计算机系统是由多台计算机组成并满足下列条件的系统：系统中任意两台计算机通过通信方式交换信息；系统中的每一台计算机都具有同等的地位，即没有主机也没有从机； 每台计算机上的资源为所有用户共享；系统中的任意若干台计算机都可以构成一个子系统，并且还能重构；任何工作都可以分布在几台计算机上，由它们并行工作、协同完成。用于管理分布式计算机系统的操作系统称为分布式计算机系统。它是具有高度并行性、故障检测和重构能力的一种高级软件系统。
5. **嵌入式操作系统（Embedded Operating System，EOS）**：将操作系统和功能软件集成于计算机硬件系统之中。嵌入式操作系统多数也是实时操作系统。
6. **微内核操作系统（Microkernel Operating System，MOS）**：由于它能有效地支持多处理机运行，故非常适用于分布式系统环境。当前比较流行的、能支持多处理机运行的操作系统，几乎全部都采用了微内核结构，如当前广泛使用的Windows操作系统，就采用了微内核结构。Linux系统属于单内核，不是微内核。微内核技术的优点：（1）统一的接口；（2）可伸缩性好；（3）可移植性好；（4）实时性好；（5）安全可靠性高，安全是微内核的特性；（6）支持分布式系统、支持多处理器的架构和高度并行的应用程序；（7）真正面向对象的操作系统。

操作系统的5项基本功能，包括进程管理（又称处理机管理）、存储管理、设备管理、文件管理和作业管理。

### 进程管理

进程(process)是资源分配和对立运行的基本单位，进程就是一个程序从开始到结束的一个完整的生命周期，进程就像一艘船，程序就是游客，当游客要过河买票了，就创建了一个进程，船在向对岸行驶的过程中就是进程运行周期，靠岸了游客下船了，那就拜拜了，进程被注销了。

从静态角度看，进程实体由程序块、进程控制块(简称PCB)和数据块三部分组成。

程序块描述该进程所要完成的任务；

数据块包括程序在执行时所需要的数据和工作区；

进程控制块包括进程的描述信息、控制信息、资源管理信息和CPU现场保护信息等，反映了进程的动态特性，如下图所示：



PCB是进程存在的唯一标志，PCB描述了进程的基本情况。

在创建一个进程时，首先创建其PCB，然后才能根据PCB中的信息对进程实施有效的管理和控制。当一个进程完成其功能后，系统则释放PCB，进程也随之消亡。

#### 进程状态

进程状态分为三态模型和五态模型。

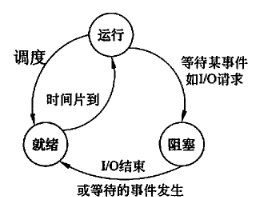
**三态模型**：包括运行、就绪和阻塞。

运行态🡪就绪态：运行时间片到；出现更高优先权进程。

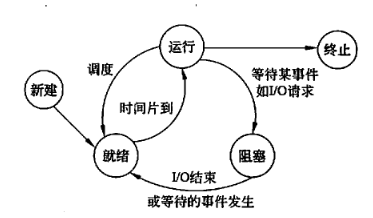
运行态🡪阻塞态：等待使用资源，如等待外设传输；等待人工干预。

就绪态🡪运行态：CPU空闲时调度选择一个就绪进程。

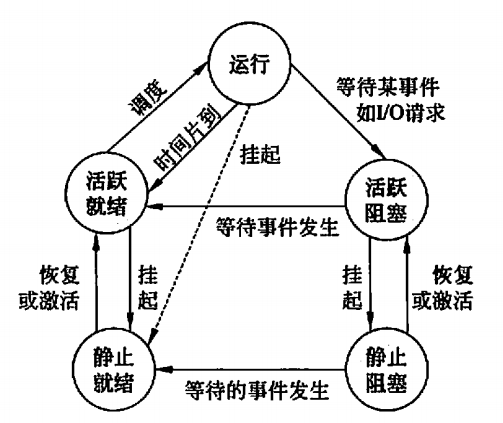
阻塞态🡪就绪态：资源得到满足，如外设传输结束；人工干预完成。



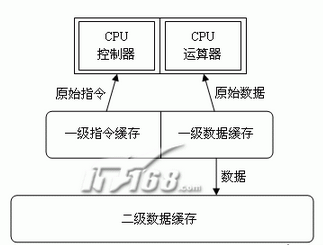
**五态模型**：对于一个实际的系统，进程的状态及其转换更为复杂。引入新建态和终止态构成了进程的五态模型。



具有挂起状态的进程状态转换如下图所示：



进程了解一下后，看到了进程的5态模型，看到了这个模型，让我有一种豁然开朗的感觉，以前老是随口所说的 CPU的一级，二级缓存原来基本原理是这样的。



二级缓存就是进程模型中的“静止就绪”、“静止阻塞”两种状态，就是将优先级较低的，或者需要等待的进程先暂时挪移到二级缓存中。一级缓存就是进程模型中的“活跃就绪”、“活跃阻塞”，就是优先级高的，或者已经具备运行条件等待执行的。而运行就是CPU的运行过程。 这也说明了一些事情，为什么CPU ，磁盘等等二级缓存越大就越贵了，因为缓存越大，可供中间等待的任务就越多，那么任务间的切换就越快，速度就越快了。

#### 进程互斥与同步

互斥原则

* 有空让进
* 忙则等待
* 多种选一
* 有限等待
* 让权等待

信号量：semaphore[英][ˈseməfɔ:(r)] 它负责协调各个线程, 以保证它们能够正确、合理的使用公共资源。

P原语： Passeren(占有)，阻塞原语，负责把当前进程由运行状态转换为阻塞状态，操作为：申请一个空闲资源（把信号量减1），若成功，则退出；若失败，则该进程被阻塞；

V原语： V是荷兰语Vrijgeven（释放）的首字母。为唤醒原语，负责把一个被阻塞的进程唤醒，它有一个参数表，存放着等待被唤醒的进程信息。操作为：释放一个被占用的资源（把信号量加1），如果发现有被阻塞的进程，则选择一个唤醒之。

在PV原语执行期间不允许有中断的发生，根本原理是机器指令。

信号量包含两类，一类是公用信号量，他实现进程间的互斥，初值为1或资源数目；另一类是私用信号量，它实现进程间的同步，初值为0或某个正数。

PV操作是对信号量的操作。

P是给信号量减1、V是给信号量+1。

P操作的是自己的私有信号量、V是操作别人的私有信号量。公有信号量每个进程的PV都能操作。

#### 前趋图

#### 进程调度

调度算法

* 先来先服务
* 优先数调度
* 轮转法

#### 死锁

死锁是指多个进程之间互相等待对方的资源，而在得到对方资源之前又不释放自己的资源，这样，造成循环等待的一种现象。

1、死锁发生的必要条件

产生死锁的根本原因在于系统提供的资源个数少于并发进程要求的该类资源数。

2、银行家算法

指在分配资源之前先看清楚，资源分配后是否会导致系统死锁。如果会死锁，则不分配，否则就分配。

3、解决死锁的策略

（1）死锁预防。破坏导致死锁必要条件中的任意一个就可以预防死锁。

（2）死锁避免。指进程在每次申请资源时判断这些操作是否安全。

（3）死锁检测。判断系统是否处于死锁状态，如果是，则执行死锁解除策略。

（4）死锁解除。将某进程所拥有的资源强行收回，分配给其他的进程。

#### 线程

线程可共享进程的资源与地址空间，通过线程的活动，进程可以提供多种服务或实行子任务并行。多线程实现的并行避免了进程间并行的缺点：创建线程的开销比创建进程要小，同一进程的线程共享进程的地址空间。

线程共享的环境包括：进程代码段、进程的公有数据(利用这些共享的数据，线程很容易的实现相互之间的通讯)、进程打开的文件描述符、信号的处理器、进程的当前目录和进程用户ID与进程组ID。

进程拥有这许多共性的同时，还拥有自己的个性。有了这些个性，线程才能实现并发性。这些个性包括：

    1.线程ID  
      每个线程都有自己的线程ID，这个ID在本进程中是唯一的。进程用此来标

   识线程。

    2.寄存器组的值  
       由于线程间是并发运行的，每个线程有自己不同的运行线索，当从一个线

   程切换到另一个线程上时，必须将原有的线程的寄存器集合的状态保存，以便

   将来该线程在被重新切换到时能得以恢复。

     3.线程的堆栈  
       堆栈是保证线程独立运行所必须的。  
       线程函数可以调用函数，而被调用函数中又是可以层层嵌套的，所以线程

   必须拥有自己的函数堆栈，使得函数调用可以正常执行，不受其他线程的影

   响。

    4.错误返回码  
       由于同一个进程中有很多个线程在同时运行，可能某个线程进行系统调用

   后设置了errno值，而在该线程还没有处理这个错误，另外一个线程就在此时

   被调度器投入运行，这样错误值就有可能被修改。  
       所以，不同的线程应该拥有自己的错误返回码变量。

    5.线程的信号屏蔽码  
       由于每个线程所感兴趣的信号不同，所以线程的信号屏蔽码应该由线程自己管理。但所有的线程都共享同样的信号处理器。

    6.线程的优先级  
       由于线程需要像进程那样能够被调度，那么就必须要有可供调度使用的参数，这个参数就是线程的优先级。

### 设备管理

设备管理程序提供的功能：

（1）提供与进程管理系统相连接的接口；（2）进行设备分配；（3）实现设备和设备、设备和CPU等之间的并行操作；（4）进行缓冲区管理。

#### 数据传输控制方式

在计算机中，I/O系统可以有5个不同的工作方式，分别为程序控制方式、程序中断方式、DMA、通道方式和输入/输出处理机。

（1）程序控制方式：CPU直接利用I/O指令编程，实现数据的输入输出。

（2）程序中断方式：CPU利用中断方式完成数据的输入/输出。分为为多中断信号线法、中断软件查询法、雏菊链法、总线仲裁法和中断向量法。

（3）直接存储访问（Direct Memory Access，DMA）方式。使用DMAC控制器来控制和管理数据传输，DMA和CPU共享系统总线，并且具有独立访问存储器的能力。DMAC获取总线的方式有三种：暂停方式、周期窃取方式和共享方式。

（4）通道方式：一种通过执行通道程序管理I/O操作的控制器，它使主机与I/O操作之间达到更高的并行程度。

（5）输入输出处理机，也称为外围处理机，是一个专用处理机，也可以使一个通用的处理机，具有丰富的指令系统和完善的中断系统。

#### 磁盘调度算法

访问磁盘的时间由三部分组成：寻道（寻找数据所在的磁道）时间、等待（旋转等待扇区）时间和数据传输时间，其中寻道时间是决定因素。

算法种类为FCFS先来先服务算法、SSTF最短寻道时间优先、SCAN电梯算法、N步SCAN算法、C-SCAN循环扫描算法（磁头单向移动）。

#### 虚设备与SPOOLING外部设备同时联机操作技术

又称假脱机输入输出操作或排队转储技术，采用一组程序或进程模拟一台输入输出处理器。

1、SPOOLING系统主要包括以下三个部分：

（1）输入井和输出井；（2）输入缓冲区和输出缓冲区；（3）输入进程和输出进程。

2、SPOOLING技术的主要特点

（1）提高了I/O 速度；（2）设备并没有分配给任何进程；（3）实现了虚拟设备的功能。

### 存储管理

****

顺序存取：磁带存储器；

直接存取：磁盘存储器；

随机存取：主存储器；

相联存取：Cache；

#### 存储管理的方式

#### 分区存储管理

##### ****一、单用户连续分区存储管理****

早期的计算机或个人微机中，每次只有一个用户使用计算机，无多道程序设计，这些机器上运行的操作系统，其存储管理都采用单一连续分区的分配策略

###### ****1. 基本思想****

把内存分为两个区：操作系统使用的和用户使用的

###### ****2. 系统特点：****

a.**系统总是把整个用户区分给一个用户使用**

b. 实际上，内存用户区又可分为：**使用区**、**空闲区**

使用区：用户作业真正占用的连续存储区域

空闲区：分配给了用户，但未使用的区域

**内部碎片**：分配给了用户，但用户并未使用的存储区域（这是对资源的一种浪费）

c. **任何时刻，内存中的用户区只有一个作业在运行**，这种系统只适用于单用户（或单道程序）的情况

d. **进入内存的作业，独享系统中的所有资源，包括内存中的整个用户区**

e. 作业进入用户区后，**没有移动的必要**，采用这种存储分配策略，需要对用户程序实施静态地址重定位

f. **实行静态地址重定位**，不能阻止用户通过不恰当指令进入操作系统占用的存储区域。通常，为了进行存储保护，在 CPU 中设置一个用于存储保护的专用寄存器——界限寄存器。界限寄存器中总是存放着内存用户区的起始地址，当 CPU 在管态下工作时，允许访问内存中的任何地址，在目态下工作时，对内存的每一次访问，都要在硬件的控制下，与界限寄存器的内容进行比较，若发生地址越界，则产生地址越界中断，阻止本次访问的进行

###### ****3. 缺点：****

a. 每次之只能有一个作业放入内存，不适合多道程序设计，真个系统的工作效率不高，资源利用率低

b. 只要作业比用户区小，在用户区中就会形成内部碎片，造成内存资源的浪费

c. 若用户作业的相对地址空间比用户区大，该作业无法运行（大作业无法在小内存上运行）

##### ****二、固定分区存储管理****

###### ****1. 作业的组织****

一般的，固定分区存储管理总是把内存用户区划分成几个**大小不等**的**连续分区;**

分区尺寸在划分后保持不变，系统可以为每一个分区设置一个后备作业队列，形成多队列管理方式。

在这种组织方式下，一个作业到达时，总能进入到“能容纳该作业的最小分区”的后备作业队列中去排队。

可能会产生有的分区队列忙碌，有的分区队列闲置的情形。

**一种改进办法**：采用多个分区只设置一个后备作业队列的方法，当某个分区空闲时，都到这个后备队列中去挑选作业，装入运行。

###### ****2. 分区的分配与释放****

为了具体管理内存中的各个分区，操作系统中设置一张名为“分区分配表”的表格，用它记录各分区的信息以及当前的使用情况

分区分配表中，应该有：每个分区的起始地址、分区长度、使用标志（使用或者空闲）

**过程：**当需要将一个作业装入内存时，按照分区号扫描分区分配表，找到空闲的分区，把作业尺寸与分区长度对比，若能容纳作业且符合所采用的分区分配策略，就把它分配给这个作业，同时修改分区分配表中该分区的使用标志。当一个作业运行结束时，只需根据作业名，找到所使用的分区号，将该分区的使用标志修改，即可完成分区的释放工作

###### ****3. 地址重定位与存储保护****

固定分区存储管理中，采用静态地址重定位，不仅要防止用户程序对操作系统形成的侵扰，也要防止用户程序之间形成侵扰。需要在 CPU 中设置一对专用寄存器用于存储保护

**低界限寄存器：**作业所在分区的低边界地址

**高界限寄存器：**作业所在分区的高边界地址

其实就是界地址保护法

###### ****4. 固定分区的优缺点****

优点：实现简单

缺点：不允许两个作业同时存放于同一个分区中剩下的空闲部分，降低了内存利用率

##### ****三、可变分区存储管理****

###### ****1. 可变分区存储管理基本思想****

固定分区存储管理中：分区数目固定、分区尺寸固定

**基本思想：**在作业要求装入内存时，如果当时内存中有足够的存储空间满足改作业的需求，就划分一个与该作业相对地址空间同样大小的分区分配给他

可变长分区存储管理使用动态地址重定位

可变分区存储管理中：**分区数目可变**、**分区边界划分随作业需求可变**

**2. 分区管理与组织方式**

内存中有两种性质的的分区：**已分配区**和**空闲区**

**用表格法实现对分区的管理**

设置两张表格：已分配表、空闲区表

表中有：分区序号、分区的起始地址、分区的尺寸、分区的状态（空闲或占用）

当作业进入并提出存储申请时，查看空闲区表里状态为空闲的表目，如果该项的尺寸能够满足所求，就将它一分为二，分配出去的那部分在已分配表中找一个状态为空的表目进行登记，剩下的部分仍在空闲区表中占据一个表目。如果有一个作业完成运行，则根据作业名找到已分配表中对应的表目，将该项的状态改为空，随之在空闲区表中寻找一个状态为空表目，把释放分区的信息填入，将表目状态改为空闲，这里可能涉及到空闲区的合并工作

###### ****3. 空闲区分配的算法****

常用的分区分配算法：最先适应法、最佳适应法、最坏适应法

**a. 最先适应法**

方法：把最先找到的满足需求的地址最小的那个空闲分区作为分配对象

出发点：尽量减少查找时间

特点：实现简单，可保证高地址端有大的空闲分区，保证大作业的执行

**b. 最佳适应法**

方法：总是从当前所有空闲分区中找出一个能够满足存储需求的、最小的空闲分区作为分配的对象

出发点：尽可能的不把大的空闲区分割成为小的分区，以保证大作业的需要

特点：时间比较费时、麻烦

**c. 最坏适应法**

方法：总是从当前所有空闲区中找出一个能够满足存储需求的、最大的空闲分区作为分配的对象

出发点：照顾中、小作业的需求

###### ****4. 移动技术（空闲分区的合并）****

**不会出现内部碎片**：分区的划分是按照作业的尺寸进行的

可变分区方案：内存中会出现一些分散的、较小的空白区仍然不能充分利用，称之为存储器的“零头”或者叫做**外部碎片**

可变分区实行动态地址重定位，用户程序不会被“钉死”在分配给自己的存储分区中。**必要时，它可以在内存中移动**

空闲分区的合并也被称为“**存储紧凑**”，操作系统有**两种合并时机的选择方案**：

a. 调度到某个作业时，系统中的每个空闲分区的尺寸都比它所需要的存储量小，但空闲区的总存储量大于它的需求，于是进行空闲存储区的合并

b. 只要有作业运行完毕归还他所占用的存储分区，系统就进行空闲分区合并

比较两种方案：

前者花费较多的精力去管理空闲区，但空闲区合并的频率低，系统在合并上的开销小

后者总是在系统中保持一个大的空闲分区，对空谈不上更多的管理，但是空闲分区合并的频率高，系统在合并上的开销大

一般系统采用第一种合并方案

**合并需要注意的问题：**

a. 合并会增加系统的时间开销、

b. 合并是有条件的，正在进行 I/O 的进程不能移动

c. 合并时尽量减少信息的移动量

###### ****5. 分区回收算法****

内存区域中的一个分区被释放时，与它前后相邻接的分区可能有四种关系出现:

**a. 释放区的前邻接分区和后邻接分区都是已分配区**

不存在分区合并的问题，释放区自己形成一个新的空闲区

**b. 释放区的前邻接分区是一个空闲区，后邻接区是一个已分配区**

释放区和前邻接的空闲分区合并

**c. 释放区的前邻接分区是一个已分配区，后邻接区是一个空闲区**

释放区和后邻接的空闲分区合并

**d. 释放区的前邻接区和后邻接区都是一个空闲分区**

释放区和前、后邻接空闲分区合并

###### ****6. 可变分区的存储保护和地址重定位****

**a. 界地址保护法**

采用基址寄存器（存放分区的物理起始地址）和限长寄存器（存放分区长度）进行存储保护

基址寄存器和动态重定位的定位寄存器合并使用一个寄存器（节约硬件成本）

程序运行时，硬件自动进行地址比较，若地址溢出，发生地址越界中断

**b. 存储键保护法**

为每个分区设置一个保护键，相当于一把锁

为每个进程分配一个存储键，相当于一把钥匙，存放在 PSW 中

每次访问内存，都要检查锁和钥匙是否匹配，若不匹配，发生保护性中断

###### ****7. 可变分区存储管理的优点****

算法简单、实现容易、内存额外开销小、存储保护措施简单

支持多道程序设计、整个系统的工作效率提高（与固定分区相比）

###### ****8. 可变分区存储管理的缺点****

存在外部碎片、造成内存资源浪费、合并外部碎片需要花费大量时间

若用户作业的相对地址空间比用户区大，该作业无法运行（**大作业无法在小内存上运行**）

#### 页式存储管理

#### 段式存储管理

#### 段页式存储管理

#### 虚拟存储管理

**虚拟存储器**又称为虚拟存储系统，是建立在主存-辅存(外存)物理结构基础上的，由负责信息划分及主存-辅存之间信息调度的辅助硬件及操作系统的存储管理软件所组成的一种存储体系。它将主存和辅存的地址空间统一编址，形成一个庞大的存储空间，在这个大空间里，用户自由编程，完全不必考虑程序在主存中是否装得下，用户感觉到的不再是处处受到主存容量限制的存储系统，而是好像具有一个容量充分大的存储器。解决了用较小容量的内存运行大容量软件的问题。

在虚拟存储器中要注意如下三个概念。  
　　① 虚拟地址空间。又称为虚存地址空间，是应用程序员用来编写程序的地址空间，与此相对应的地址称为虚拟地址或逻辑地址。

② 主存（内存）地址空间。又称为实存地址空间，是存储、运行程序的空间，其相应的地址称为主存物理地址或实地址。

③ 辅存（外存）地址空间也就是磁盘存储器的地址空间，是用来存放程序的空间，相应的地址称为辅存地址或磁盘地址。

不难看出，主存与辅存的关系极类似于主存与高速缓存的关系，但主存—cache体系和主存—辅存体系还有一些差别。  
　　① 主存—cache体系的目的是满足程序对速度的要求，而主存—辅存体系是为了满足容量的要求。所以前者容量小，传送信息块的长度短，读/写速度快；而后一种体系容量大，传送数据块的长度长，读/写速度相对较慢。  
　　② 在主存—cache体系中，CPU可以直接访问cache和主存；而在主存—辅存体系结构中，CPU不可以直接访问辅存。

③ 为了保证速度，主存—cache体系的存取信息过程、地址变换和替换策略全部采用硬件来实现，而主存—辅存体系基本上由操作系统的存储管理软件辅助一些硬件进行数据块的划分来实现主存—辅存之间的调度，所以需要设计存储管理软件来实现这些功能。

虚拟存储器的工作原理是：在执行程序时，允许将程序的一部分调入主存，其他部分保留在辅存。即由操作系统的存储管理软件先将当前要执行的程序段（如主程序）从辅存调入主存，暂时不执行的程序段（如子程序）仍保留在辅存，当需要执行存放在辅存的某个程序段时，由CPU执行某种程序调度算法将它们调入主存。

**直接存储器访问（DMA）**，是一种完全由硬件执行I/O交换的工作方式。在这种方式中，DMA控制器从CPU完全接管对总线的控制，数据交换不经过CPU，而直接在内存和I/O设备之间进行。DMA方式一般用于高速传送成组数据。