第一章

1.1

操作系统的目标：

·方便性：使计算机系统更易于使用

·有效性：以一种高效率的方式使用资源

·可扩充性：采用曾思华结构，易于增删改

·开放性：要求统一的开放环境，能通过网络集成化，并正确、有效地协同工作，实现应用程序的移植。

操作系统的作用：

OS作为用户与计算机硬件系统之间的接口；OS作为计算机系统资源的管理者；OS实现了对计算机资源的抽象

推动操作系统发展的主要动力：

不断提高计算机资源利用率；方便用户；器件的不断更新换代。

1.2

操作系统的发展过程：未配置操作系统的计算机系统（人工操作，脱机输入输出）→单道批处理系统→多道批处理系统→分时系统→实时系统

微机操作系统分类：单用户单任务操作系统；单用户多任务操作系统；多用户多任务操作系统。

1.3

操作系统的基本特性：

并发性：在一段时间内同时存在多道运行的程序（进程）

**进程**是指在系统中作为**资源分配**的基本单位并能独立运行的活动实体

**线程**，一个进程可以包括若干个**线程，把线程作为独立运行和独立调度的基本单位**

共享性：指系统中的资源不再为某道程序所独占，而是共多道程序共同使用

虚拟性：指把一个物理实体映射为若干个对应的逻辑实体

异步性：也称为不确定性，指进程在执行中，其执行时间、顺序、向前推进的速度和完成的实践等都是不可预知的。（不确定性并不是指操作系统的功能不确定，也不是同一程序在相同的数据集上的多次运行结果不确定，而是指统一到程序的多次运行所需的时间不确定，同一批程序的多次运行序列和总运行时间不确定）

操作系统的主要功能：

处理机管理：

任务：对处理及进行分配；对处理机运行进行有效的控制和管理；处理机的分配和运行以进程为基本单位，因此对处理机的管理可归结对进程的管理

功能：**进程控制；进程同步；进程通信；进程调度**

并发是最重要的特性，其他三种特性以此为前提。

存储器管理：

任务：为多道程序的运行提供良好的环境；方便用户使用存储器；提高存储器的利用率；从逻辑上扩充内存。

功能：内存分配；内存保护；地址映射；内存扩充

设备管理器：

任务：完成用户提出的I/O请求，位用户分配I/O设备；

提高I/O设备的利用率及速度，方便用户使用I/O设备

功能：缓冲管理；设备分配；设备处理；虚拟设备

文件管理器：

任务：对用户文件和系统文件进行管理；方便用户使用文件；保证文件的安全性

功能：文件存储空间的管理；目录管理；文件的读写管理；文件的共享与保护

▲微内核OS结构：当下比较流行的操作系统几乎全部采用微内核结构

从四个方面描述：

足够小的内核：内核是精心设计的、能实现现代OS最基本核心功能的小型内核，通常包括：与硬件处理紧密相关的部分，一些较基本的功能，客户和服务器之间的通信。

基于客户/服务器模式：将操作系统中最基本的部分放入内核，而把操作系统的绝大部分功能放在内核外面的一组服务器（进程）中实现，客户和服务器之间的通信靠消息传递机制

应用“机制与策略分离”原理：

机制：实现某一功能的具体执行机构。

策略：在机制的基础上借助于某些参数和算法来实现该功能的方法

软件是策略，硬件是机制；命令是策略，实施是机制。

采用面向对象技术：把计算机中的实体当作对象处理，；通过重用以前的对象来构建新的系统，降低开发成本；通过继承，可显著减少开发的时空开销；通过封装可以藏对象中的变量和方法，修改时不会影像其他部分；还能防止不正常修改

微内核操作系统的优点：

提高了系统的可扩展性（添加、删除服务器）

提高了系统的可靠性：某个服务器（即进程）出错不会影响其他服务器

可移植性好：服务器与硬件平台无关

适用于分布式系统

缺点：运行效率低：消息传递比直接调用效率低

**第二章**

2.1前驱图和程序执行

前驱图：有向无环图。描述一个程序的各部分（程序段或者语句）间的依赖关系，或者一个大的计算的各个子任务间的因果关系

名词：前结点、后继结点、初始接点、终止结点、权重即时间

语句转化为前驱图→前驱图转化为前驱关系

**程序顺序执行**：按照某个先后次序逐个执行

特征：顺序性，封闭性，可再现性

**程序并发执行**：逻辑上相互独立的程序或程序段在执行时间上客观上互相重叠，即一个程序或程序段的执行尚未结束，另外一个程序的执行已经开始的方式

特征：间断性（异步性）、失去封闭性、不可再现性

2.2

**进程的定义**：

进程是程序的一次执行；

进程是一个程序及其数据在处理机上顺序执行时所发生的活动；

进程是具有独立功能的程序在一个数据集合上运行的过程，它是系统进行资源分配和调度的独立单位。

进程的特征：

**动态性**：进程的实质是程序在处理机上的一次执行过程，因此是动态。**动态性是进程的最基本的特征**。同时动态性还表现在进程是有生命周期的，它**因创建而产生，因调度而执行，因得不到资源而暂停，因撤销而消亡**

**并发性**：指多个进程实体同时存在与内存中，能在一段时间内同时运行。**引入进程的目的就是为了使进程能并发执行，**以提高资源利用率，所以并发誓进程的重要特征，也是OS的重要特征

**独立性：**指进程是一个能独立运行的基本单位，也是系统进行资源分配和调度的独立单位

**异步性：**指进程以各自独立的、不可预知的速度向前推进

**进程的三种基本状态：**

New新建/创建：进程正在创建中的状态

**Ready就绪**：进程已经获得除了处理机以外的所有资源，等待分配处理机执行的等待状态

**Running运行/执行**：当一个进程获得必要的资源并正在处理机上执行的状态

**Waiting等待/阻塞**：正在执行的进程由于发生某件事而暂时无法执行下去，此时进程所出的状态

Terminated终止/撤销/退出：进程执行完毕，释放所占资源的状态

**进程的挂起状态**：挂起状态使进程置于静止状态；一般情况下，挂起状态的进程将从内存移到外存；正在执行的进程暂不接受调度；就绪的进程暂不执行调度；阻塞的进程即使阻塞事件释放也不能继续执行

**引起挂起状态的原因**：终端用户的需要；父进程的需要；操作系统的需要；对换的需要；符合调节的需要；

**进程控制块PCB：**是操作系统为了管理和控制进程的运行，而为每一个进程定义的一个数据结构，它记录了系统管理进程所需的全部信息。PCB常驻内存，存放在操作系统中专门开辟的PCB区内。

**PCB的作用**：**作为独立运行基本单位的标志**；PCB是进程存在的唯一标志。系统创建一个新进程时，就为他建立了一个PCB；进程结束时又回收其PCB，进程于是也随之消亡。 **能实现间断性运行方式**。 **提供进程管理所需要的信息。** **提供进程调度所需要的信息。 实现与其他进程的同步和通信。**

**进程控制块的组织方式：链接方式、索引方式**

**进程的创建：用户登录、作业调度、提供服务、应用请求会导致进程创建**

**操作系统一旦发现了要求创建进程的事件后，便调用进程创建原语创建新进程。**

**系统中发生了要撤销进程的事件，操作系统便调用撤销原于Kill去撤销进程。**

**当进程期待的事件尚未出现时，该进程调用阻塞原语Block把自己阻塞起来。**

**处于阻塞状态的进程，当期待的事件出现时，有其他相关进程调用唤醒原语Wakeup把阻塞的进程唤醒，使其进入就绪状态。**

**当引起进程挂起的事件发生时，系统就将利用挂起原语Suspend将指定进程挂起。**

**当发生激活进程的事件时，系统就将利用激活原语Active将指定进程激活**

**2.4**

**进程同步：**是指对多个相关进程在执行次序上进行协调，它的目的是使系统中诸多进程之间能有小弟共享资源和相互合作，从而使程序的执行具有可再现性。

系统中诸多进程在逻辑上存在两种制约关系：

直接制约关系：进程同步 间接制约关系：进程互斥

**临界资源**：一次只允许一个进程使用的资源

进程之间采用**互斥方式**实现对临界资源的共享，从而实现并行程序的封闭性。

引起不可再现性是因为对临界资源没有进行互斥访问。

每一个进程在访问临界资源的那一段代码称为**临界区**

对临界区需要进行保护（互斥访问）

同步机制需要遵循的规则：空闲让进、忙则等待、有限等待、让权等待

**信号量机制：**P、V操作（原子操作）P、V操作应作为一个整体实施，不允许分割。 在同一程序段成对出现同一信号量P、V操作，否则会造成系统瘫痪

信号量：用于表示资源数目或请求使用某一资源的进程个数的整型量。

整型信号量：非负整数，用于表示资源数目。除了初始化，只能P、V操作。

记录型信号量：大于零表示剩余资源数目，小于零表示当前阻塞进程数。

And型信号量：同时请求所有需要资源，当前无满足的资源则进入等待队列。

**生产者消费者问题**

先执行资源信号量的P操作，后执行缓冲区信号量的P操作，否则容易出现死锁

（full，empty）必须成对出现，但可以分别处于不同的程序中

**哲学家进餐问题**

·至多允许四位哲学家进餐→设置一个count=4，作为一个缓冲区，最多允许四人访问。

·仅当哲学家的左右手筷子都能拿起来时才允许进餐→Swait，Ssignal

·规定奇数号哲学家先拿左筷子再拿右筷子，偶数号哲学家相反

**读者写者问题**

读者优先：第一位读者把控制写的信号量给P掉，最后一个把他给V掉，读者计数

写者优先：添加一个互斥信号量，读者写者同时访问，读者写者都去竞争这个资源，获得资源即执行响应的操作

进程与线程的却别：

**进程是程序资源分配的最小单位，线程是程序执行的最小单位**

进程有自己的内存地址空间，线程包含在进程的地址空间中

进程的分配开销比线程大，但hi进程的健壮性比线程高，进程之间相互不影响，线程一个挂掉了可能会造成进程崩溃

**第三章**

**作业调度算法：**先来先服务FCFS、短作业优先SJF、

**优先权调度算法：动态优先权调度算法：高响应比优先调度算法（HRRN）**

优先权 = （等待时间+要求服务时间）/要求服务时间

***多级反馈队列调度算法***

**3.5**

**死锁产生的原因：竞争不可抢占资源引起死锁；竞争可消耗资源引起死锁；进程间推进顺序不当引起死锁。**

**死锁的定义：**指两个或者两个以上的进程在执行中，因争夺资源而造成的一种相互等待的现象，他们都将无法推进下去。此时称系统处于死锁状态，这些永远在互相等待的进程称为死锁进程。

参与死锁的进程数至少为两个

参与死锁的进程均等待资源

参与死锁的进程至少有两个已经占有资源

死锁晋城市系统中当前进程集合的一个子集

**产生死锁的四个必要条件：**

**互斥条件：**进程对所分配的资源进行排他性使用

**请求和保持条件：**进程已经保持了至少一个资源，但又提出了新的资源请求，而该资源已经被其他进程占有，此时请求进程被阻塞，但对自己已经获得的资源保持不放

**不可剥夺条件：**进程已获得的资源，在未使用完之前不能被抢占，只能在进程使用完时由自己释放

**环路等待条件：** 发生死锁的时候，必然存在一个资源-进程的循环链

**处理死锁的方法：**

**预防死锁：**通过设置某些限制条件，去破坏产生死锁的四个必要条件中的一个或者几个，来防止死锁的发生

**避免死锁：**在资源的动态分配过程中，用某种方法去防止系统进入不安全状态，从而避免死锁的发生

**检测死锁：**允许系统在运行过程中发生死锁，但可以设置检测机构及时检测死锁的发生，并采取适当措施加以清除

**解除死锁：**当检测出死锁后，便采取适当措施将进程从死锁状态中解脱出来

**预防死锁是提前破坏死锁发生的必要条件；避免死锁是在资源分配过程中避免系统进入不安全状态**

**银行家算法：**

可利用资源向量Available[j]=k，表示Rj类资源有k个

最大需求矩阵Max[i,j]=k，表示Pi最大请求k个Rj资源

分配矩阵Allocation[i,j]=k，进程Pi已经分配到k个Rj类资源

需求矩阵Need[i,j]=k，进程Pi还需要k个Rj类资源

Need = Max – Allocation

资源分配图：请求边，分配边

**第四章**

逻辑地址：相对地址、虚地址

物理地址：绝对地址、实地址

地址映射：地址转换

单一连续分区分配（静态分区技术）：仅用于单用户单任务系统

固定分区分配（静态分区技术）：可用于多道系统

动态分区分配（动态分区技术）

动态可重定位分配（动态分区技术）：引入了动态**重定位**和内存**紧凑技术**

连续分配存储管理方式产生的问题：要求连续的存储区、碎片问题

4.3.4**基于顺序搜索的动态分区分配算法：**

**首次适应算法（First Fit）：**空闲分区按地址递增的次序排序，内存分配时，从空闲分区表/链首开始顺序查找，直到找到第一个满足大小要求的空闲区

**循环首次适应算法（Next Fit）：**首次适应算法演变而来，每次不从空闲分区表/链首开始查找，而是上次找到空闲分区的下一个空闲分区开始查找，直到找到满足大小要求的空闲分区为止

**最佳适应算法（Best Fit）**：空闲分区表/链按照容量大小递增的次序排列，在进行内存分配时，从空闲分区表/链首开始查找，直到找到第一个满足大小要求的空闲分区

**最坏适应算法（Worse Fit）**

**几种算法比较**

|  |  |
| --- | --- |
| **算法名称** | **特点** |
| **First Fit** | **优先利用内存低地址部分空闲区。由于低地址被不断划分，留下很多难以利用的很小的空闲分区，每次查找又是从头开始的，增加了查找可用空闲分区的开销** |
| **Next Fit** | **从上一次找到的空闲区往下开始查找，使存储空间的利用更加均衡，较小的空闲区不会集中在存储区的一端，但会导致缺乏大的空闲区** |
| **Best Fit** | **排序空闲区，每次都能找到最适合的空闲区存储作业（作业的大小略小于存储区的大小），但是会导致留下很小、难以利用的小空闲区** |
| **Worse Fit** | **总挑选最大的分配空闲区分割给作业用，产生的空闲区不至于太小，但是大作业来的时候容易找不到对应的空闲区** |

变连续分配为离散分配，允许将作业离散放到多个不想临界的分区中

分页式存储管理：离散分配的基本单位是页

分段式存储管理：离散分配的基本单位是段

段页式存储管理：离散分配的基本单位是段、页

**4.5分页存储管理方式：**

**具有快表的地址变换机构：CPU产生逻辑地址的页号，首先在快表中寻找，若命中则直接找出对应的物理块；若未命中，再到页表中找其对应的物理块，并将之复制到快表中。快表满的话按照某种算法淘汰。**

|  |  |  |
| --- | --- | --- |
|  | **页式存储管理** | **段式存储管理** |
| **目的** | 实现非连续分配，解决碎片问题 | 更好地满足用户需要 |
| **信息单位** | 页（物理单位） | 段（逻辑单位） |
| **大小** | 固定（由系统定） | 不定（由用户程序定） |
| **内存分配单位** | 页 | 段 |
| **作业地址空间** | 一维 | 二维 |
| **优点** | 有效解决了碎片问题（没有外碎片，每个内碎片不超过页大小）；有效提高内存的利用率；程序不必连续存放 | 更好地实现数据共享与保护；段长可动态增长；便于动态链接 |

**第五章**

**5.1：**

**虚拟存储器：是指具有请求调入功能和置换功能，能从逻辑上堆内存容量加以扩充的一种存储器结构**

**逻辑容量 = 内存 + 外存 速度接近于内存，成本接近于外存。**

**特征：多次性、对换性、虚拟性**

**请求分页：**

**请求分段：**

**请求段页式：**

**页面置换算法：**

**最佳置换算法OPT：向即将到的页看，置换掉出现最晚的那个**

**先进先出FIFO：增加物理块，缺页次数也可能减小，唯一一个**

**最近最久未使用算法（LRU）：向已经执行过的页看，置换掉最早出现的**

**最近最少使用算法（LFU）：**

**第六章**

6.4.3对I/O设备的控制方式：

·采用轮询的可编程I/O方式

·使用中断的可编程I/O方式

·直接存储器访问方式

·I/O通道方式

6.6

系统调用

*系统调用是用户程序取得OS服务的唯一途径*

库函数

用户层I/O软件

**假脱机技术（SPOOLing）：SPOOLing是指在多道程序的环境下，利用多道程序中的一道或者两道程序来模拟外围控制机，从而在联机的条件下实现脱机I/O的功能。**

**SPOOLing系统的组成：**

**输入井和输出井（外存中）：**两个大的内存空间，用于模拟输入设备和输出设备；输入井用于暂存I/O设备输入的数据；输出井用于暂存用户程序的输出数据。

**输入缓冲区和输出缓冲区（内存中）：**为了缓和CPU、磁盘、I/O设备之间速度不匹配的矛盾而设置的； I/O设备→输入缓冲区→输入井 && 输出井→输出缓冲区→I/O设备

**输入进程和输出进程：**用于控制I/O设备与磁盘井之间的信息交换 输入进程SPi相当于脱机输入控制器，控制：I/O设备→输入缓冲区→输入井 输出进程SPo相当于脱机输出控制器，控制：输出井→输出缓冲区→I/O设备

**SPOOLing系统的特点：提高了I/O速度；将独占设备改造为共享设备；实现了虚拟设备功能**

**单缓冲区和双缓冲区：**

**单缓冲区：**在设备和处理机之间设置一个缓冲区，只能存放输入数据和输出数据 缓解了CPU和I/O设备间速度不匹配的矛盾

**双缓冲区：**在设备和处理及之间设置两个缓冲区，可同时输入、输出数据

**6.8磁盘存储器的性能和调度：**

提高磁盘I/O速度的主要途径：选择性能好的磁盘、采用好的磁盘调度算法、设置磁盘告诉缓存、其他方法

**存储容量：磁头（盘面）数×磁道（柱面）数×每道扇区数×每扇区字节数**

**寻道时间：**将磁头从当前位置移动到指定磁道所经历的时间

**旋转延迟时间：**指定扇区移动到磁头下面所经历的时间

**传输时间：**将扇区上的数据从磁盘读出或向磁盘写入数据所经历的时间

**控制器时间：**

**磁盘调度算法：先来先服务FCFS、最短寻道时间优先SSTF、SCAN算法**

**·先来先服务FCFS：**按照磁道请求的先后顺序，可能造成磁头臂来回反复移动，增加了等待时间，而对机器结构不利，寻道时间开销大，可用于输入/输出负载较轻的系统

**·最短寻道时间：**请求磁道所要移动的距离大小作为优先因素，距离小的优先，大的靠后。可能导致一些请求长时间内得不到服务的机会

**存在的问题：**

**磁道粘着：**可能出现磁臂停留在某处的情况，即反复请求某一磁道，

**磁道歧视：**某一时刻外磁道请求不断，内磁道的请求可能长时间得不到满足，这种现象称为“磁道歧视”

·**SCAN算法：**往复扫描各个柱面（磁道）并未途径柱面（磁道）的请求服务。从外到内，从内到外，反复移动

**第七章**

**7.2文件的逻辑结构**

所有文件存在着两种形式的结构：**文件的逻辑结构（文件组织）、文件的物理结构（文件的存储结构）**

文件逻辑结构的类型：

·按是否有结构分类：

有结构文件

无结构文件

·按文件的组织方式分类：

顺序文件

索引文件

顺序索引文件

|  |  |  |  |
| --- | --- | --- | --- |
| 文件类型 | 简述 | 优点 |  |
| 顺序文件 | 把逻辑上连续的文件信息依次放在连续编号的物理块中，只要知道文件在存储设备上的起始地址和文件长度，就可以进行存取 | 顺序存取速度快 |  |
| 索引文件 |  |  |  |
| 顺序索引文件 |  |  |  |

**7.3文件目录：**用于检索文件的目录称为文件目录，它是由目录项所构成的有序序列。

**文件控制块FCB：文件控制块是操作系统为管理文件而设置的数据结构，存放了文件的有关说明信息，是文件存在的标志**

FCB包括：

·基本信息类：文件名、文件物理**位置、**文件逻辑结构、文件物理结构（顺序、链接、索引）

·存取控制信息类：权限、用户名、口令、共享计数

·使用信息类：建立日期、最后修改日期等

文件目录：文件控制块的有序集合

目录文件：为了实现对文件目录管理，通常将文件目录以文件的形式保存在外存中，这个文件就叫目录文件

**二者是同一事物，用途角度：文件目录 实现角度：目录文件**

索引结点：索引结点把文件名和文件描述信息分开，将FCB分为**文件名、i结点指针和响应的i结点。**查询时只调入文件目录（仅包含文件名和索引结点指针），找到后才调入响应的结点。

分为磁盘索引结点、内存索引结点

**目录的结构**：

·**单级文件目录**：在整个系统中只建立一张目录表

优点：管理简单，易于实现按名存取

缺点：限制了用户对文件的命名（不允许重名）；文件平均检索时间长平均（N/2）；不便于实现文件共享；只适用于单用户环境

·**两级文件目录**：在整个系统中建立两级目录。每个用户建立一个单独的用户文件目录；系统中为所有用户建立一个主文件目录

优点：提高了检索目录的速度。如n个用户，每个用户有m个文件，则最多检索n+m个目录项而不是n\*m项；不同用户目录中可重名。

缺点：对文件的共享不方便；增加了系统的开销，缺乏灵活性，无法反映真实世界复杂的文件结构形式。

·**树形结构目录：**只有一个根目录，除根目录外，剩下每一个目录或者文件都有唯一的一个上级目录。

优点：层次结构清晰，便于管理和保护；解决重名问题；查找速度加快，因为每个目录下的文件数目较少。

缺点：查找一个文件，需要按路径名逐级访问中间节点增加了磁盘访问次数

**第八章**

8.1.3**FAT文件系统（链接组织方式）**

**8.2.2位示图法：利用二进制的一位（bit）来表示磁盘中一个盘块的使用情况，**每个字节的每一位都对应了一个物理块的状态。取1表示对应物理块已分配；取0表示对应的物理块未分配。 空闲空间表现为位图或位向量

特点：占空间少，可放入内存，易于访问。