

**1.1 操作系统的目标：**有效性 方便性 可扩充性 开放性

**1.2 操作系统的作用** 1.OS 作为用户与计算机硬件系统之间的接口（命令方式，系统调用方式，图像和窗口式。） 2.OS 作为计算机系统资源的管理者 3.OS 实现了对计算机资源的抽象

**1.3 操作系统的定义：**操作系统是一组控制和管理计算机硬件和软件资源，合理地各类作业进行调度，以及方便用户使用的程序集合。

**1.4 操作系统的基本特性** 1.并发性 2.平行性 3.引入进程 4.引入线程 5.共享性：是指系统中的资源可供内存中多个并发执行的进程共同使用。互斥共享、同时访问方式 6.虚拟技术 是指通过某种技术把一个物理实体变为若干个逻辑上的对应物。分为时分复用和空分复用技术。7.异步性 进程是以人们不可预知的速度向前推进，此即进程的异步性。

**1.5 操作系统的主要功能** 1.处理机管理功能：进程控制，进程同步，进程通信，调度 2.存储器管理功能：内存分配、内存保护、地址映射、内存扩充 3.设备管理功能：缓冲管理、设备分配、设备处理 4.文件管理功能：文件存储空间的管理、目录管理、文件的读/管理和保护。操作系统与用户之间接口 用户接口、程序接口

**2.1 进程的特征：**1.结构特征 2.动态性 3.并发性 4.独立性 5.异步性。

**2.2 进程的概念：**进程是进程实体的运行过程，是系统进行资源分配和调度的一个独立单位。**进程的状态：**基本状态 1.就绪状态 2.执行状态 3.阻塞状态。挂起状态，创建状态和终止状态。

**2.4 进程通信类型：**1.共享存储器系统 2.消息传递系统 3.管道通信 4.基于共享数据结构的通信方式 5.基于共享存储区通信方式

**2.5 线程与进程的区别：**进程是进程实体的运行过程，是系统进行资源分配和调度的一个独立单位。线程是比进程更小的单位。通常在一个进程中可以包含若干个线程，他们可以利用进程所拥有的资源。OS 中把进程作为分配资源的基本单位，而把线程作为独立运行和独立调度的基本单位。

**2.6 程序并发执行的特征：**1.间断性 2.失去封闭性 3.不可再现性

**3.1 低级调度：**通常也把低级调度称为进程调度或短进程调度，它所调度的对象是进程。在多批道处理、分时和实时三种类型的 OS 中，都必须配置这级调度。主要功能：1. 保存处理机的现场信息 2 按某种算法选取进程 3.把处理器分配给进程。

**3.2 调度算法的若干准则：**1) 面向用户的准则：周转时间短、响应时间快、截止时间的保证、优先权准则；2) 面向系统的准则：系统吞吐量高、处理机利用率好、各类资源的平衡利用。

**3.3 短作业（进程）优先调度算法 SJ(P)F：**是指对短作业或短进程优先调度的算法。短作业优先(SJF)的调度算法是从后备队列中选择一个或若干个估计运行时间最短的作业，将他们调入内存运行。而短进程优先（SPF）调度算法则是从就绪队列中选出一个估计运行时间最短的进程，将处理机分配给它，使它立即执行并一直执行到完成，或发生某件事而被阻塞放弃处理机时再重新调度。该算法有效的降低了作业的平均等待时间，提高系统吞吐量。缺点：1) 对长作业不利；2)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；3) 该算法不一定能真正做到短作业优先调度。

**3.4 高响应比优先调度算法：**为每个作业引入动态优先权，并使作业的优先权随着等待时间的增加而以速率  $a$  提高，则长作业在等待一定时间后，必然有机会分配到处理机。

$$\text{优先权} = (\text{等待时间} + \text{要求服务时间}) / \text{要求服务时间} = \text{响应时间} / \text{要求服务时间} = R_p$$

由上式可看出：1.有利于短作业；2.它实现的是先来先服务；3.对于长作业，作业的优先级可以随等待时间的增加而提高，当其等待时间足够长，其优先级便可升到很高，从而也可获得处理机。总之，该算法既照顾了短作业，也考虑了作业到达的先后次序，不会使长作业长期得不到服务，但每要进行调度之前，都要做相应比的计算，增加系统开销

**3.5 最低松弛度优先算法（LLF）：**该算法是根据任务紧急（或松弛）的程度，来确定任务的优先级。任务的紧急程度愈高，为该任务所赋予的优先级就愈高，以使之优先执行。

$$A \text{ 的松弛度} = \text{必须完成的时间} - \text{其本身的运行时间} - \text{当前时间}$$

**3.6 死锁的概念：**指多个进程在运行过程中因争夺资源而造成的一种僵局，当进程处于这种僵持状态时，若无外力作用，他们都将无法再向前推进。**产生死锁的必要条件：**1.互斥条件；2.请求和保持条件；3.不剥夺条件；4.环路等待条件。**产生死锁的原因：**1) 竞争资源：当系统中供进程共享的资源，其数目不足以满足诸进程的需要时，会引起诸进程对资源的竞争而产生死锁。2) 进程间推进顺序非法：进程在运行过程中，请求和释放资源的顺序不当，也同样会导致进程产生死锁。**预防死锁的方法：**1.摒弃“请求和保持”条件；2.摒弃“不剥夺”条件；3.摒弃“环路等待”条件。**死锁的解除：**1.剥夺资源 2.撤销进程。

**4.1 程序的装入：**绝对装入方式；2.可重定位装入方式；3.动态运行时装入方式。

**4.2 分页和分段的主要区别：**A 分页和分段都采用离散分配的方式，且都要通过抵制映射机构来实现地址变换，这是他们的共同点，B 对于他们的不同点有三，第一：从功能上页是信息的物理单位，分页是实现离散分配方式，以消减内存的外零头提高内存的利用率，即满足系统管理的需要而不是用户的需要，而段式信息的逻辑单位，他含有一组其意义相对完整的信息，目的是为了能更好的满足用户的需要；第二：页的大小固定且由系统确定，而段的长度却不固定，决定于用户所编写的程序；第三：分页的作业地址空间是一维的，而分段的作业地址空间是二维的。

**4.3 虚拟存储器的概念：**所谓虚拟存储器是指具有请求调入功能和置换功能，能从逻辑上对内存容量加以扩充的一种存储器系统。**虚拟存储器的实现方法：**1.分页请求系统 2.请求分段系统。**虚拟存储器的特征：**1.多次性 2.对换性 3.虚拟性。

**4.4 局部性原理：**1.程序执行时，除了少部分的转移和过程调用指令外，在大多数情况下仍是顺序执行的；2.过程调用将会使程序的执行轨迹由一部分区域转至另一部分区域，但经研究看出，过程调用的深度在大多数情况下都不超过 5；3.程序中存在许多循环结构；4.程序中还包许多对数据结构的处理。局限性还表现在：时间局限性和空间局限性。

**5.1 设备控制器的基本功能：**1.接收和识别命令 数据交换 标识和报告设备的状态 地址识别 数据缓冲 差错控制 2.检查用户 I/O 请求的合法性，了解 I/O 设备的状态，传递有关参数，设置设备的工作方式 3.发出 I/O 命令 4.及时响应由控制器或通道发来的中断请求，并根据其中断类型调用相应的中断处理程序进行处理 5.对于设置有通道的计算机系统，驱动程序还应能够根据用户的 I/O 请求，自动地构成通道程序。

**5.2 I/O 通道类型：**字节多路通道 数组选择通道 数组多路通道。主要目的是为了建立独立的 IO 操作，不仅使数据的传送能独立于 cpu，而且也希望有关对 IO 操作的组织、管理及结束处理尽量独立，以保证 cpu 有更多的时间去进行数据处理。

**5.3 设备驱动程序的特点:** 1.驱动程序主要是指在请求 I/O 的进程与设备控制器之间的一个通信和转换程序 2.驱动程序与设备控制器和 I/O 设备的硬件特性紧密相关,因而对不同类型的设备应配置不同的驱动程序 3.驱动程序与 I/O 设备所采用的 I/O 控制方式紧密相关 4.由于驱动程序与硬件紧密相关,因而其中的一部分必须用汇编语言书写。5.驱动程序应允许可重入。6.驱动程序不允许系统调用。

**5.4 设备驱动程序的主要功能:** 1) 接收由设备独立性软件发来的命令和参数,并将命令中的抽象要求转换为具体要求; 2) 检查用户 I/O 请求的合法性,了解 I/O 设备的状态,传递有关参数,设置设备的工作方式; 3) 发出 I/O 命令; 4) 及时响应由控制器或通道发来的中断请求,并根据其中断类型调用相应的中断处理程序进行处理; 5) 对于设置有通道的计算机系统,驱动程序还应能根据用户的 I/O 请求,自动地构成通道程序。

**信号量机制:** 整形信号量,记录性信号量,mutex 信号量,信号集

**7. 并发及并行的区别:** 并行是指两个或多个事件在同一时刻发生,而并发性是多个事件在同一时间间隔内发生。在多道程序环境下,并发是指在一段时间内宏观上有多个程序在同时运行,微观上这些程序只能是分时地交替执行,在计算机系统中有多个处理器,则这些并发执行程序被分配到多个处理器上实现并行执行,利用每个处理器来处理一个并发执行程序。

**8.临界区:** 不论是硬件临界资源还是软件临界资源,多个进程必须互斥地对它进行访问,人们把每个进程中访问临界资源的那段代码称为临界区。

**12. 作业调度(高级调度):** 主要功能是根据作业控制块中的信息,审查系统能否满足用户作业的资源需求,以及按照一定的算法,从外存的后备队列中选取某些作业调入内存,并为他们创建进程,分配必要的资源,然后再将新创建的进程插入就绪队列准备执行

**13.抢占的原则有:** 1.优先权原则。(2) 短作业(进程)优先原则。(3) 时间片原则。

**14.调度算法(抢占,非抢占的区别及联系):** 非抢占方式:在采用这种调度方式时,一旦把处理器分配给某进程,不管他要运行多长时间,都一直让他运行下去,直到该进程完成,资源释放处理器。或发生某事件而被阻塞时才把处理器分配给其他进程,这种调度方式实现简单系统开销小,但难以满足紧急任务的要求

抢占方式: 允许调度程序根据某种原则去展延某个正在执行的进程,将已分配给该进程的处理器重新分配给另一进程。优点是可以防止一个长进程长时间占用处理器,能为对大多数进程提供更公平的服务,特别是能满足对相应时间有着严格要求的实时任务的需求,但抢占方式比非抢占方式调度所需付出的开销较大,抢占式基于优先权原则,短作业优先原则,时间片原则,

**16.实现实时调度的基本条件** 1. 提供必要的信息 : 就绪时间。开始截止时间和完成截止时间。处理时间。资源要求。优先级 2. 系统处理能力强 3.采用抢占式调度机制 4.具有快速切换机制。实时调度算法的分类: 1.非抢占式调度算法: 非抢占式轮转调度算法和非抢占式优先调度算法 2.抢占式调度算法: 基于时钟中断的抢占式优先权调度算法和立即抢占。

**31.设备处理方式:** 1.为每一类设备设置一个进程,专门用于执行这类设备的 I/O 操作。2.在整个系统中设置一个 I/O 进程,专门用于执行系统中所有各类设备的 I/O 操作。3.不设置专门的设备处理进程,而只为各类设备设置相应的设备处理程序,供用户进程或系统进程调用

**32.为何要引入设备独立性?**在现代操作系统中,为了提高系统的可适应性和可扩展性,都毫无例外地实现了设备独立性也即是设备无关性,其基本含义是应用程序独立于具体使用的物理设备,即应用程序以逻辑设备名称来请求使用某类设备,作用: (1)设备分配时的灵活性(2)易于实现 I/O 重定向(指用于 I/O 操作的设备可以更换即重方向而不必改变应用程序)。

**33.如何实现设备独立性**为了实现设备的独立性,应引入逻辑设备和物理设备两个概念,在应用程序中,使用逻辑设备名称来请求使用某类设备,而系统执行时,是使用物理设备名称,鉴于驱动程序是一个与硬件(或设备)紧密相关的软件,必须在驱动程序之上设备一层软件,称为设备独立性软件,以执行所有设备的公有操作,完成逻辑设备名到物理设备名的转换(为此应设备一张逻辑设备表)并向用户层(或文件层)软件提供统一接口,从而实现设备的独立性

**34.磁盘调度算法:** 1.先来先服务 2 最短寻道时间优先 3 扫描(scan) 算法 4 循环扫描(cscan)算法 5 NStepSCAN 和 FSCAN 调度算法

**35.顺序文件的优缺点:** 优点:a 对诸记录进行批量存取时,存取效率最高 b 只有顺序文件才能存储在磁带上并能有效的工作; 缺点 a 交互应用时性能差 b 增加或修改一个记录比较困难,为了解决这一问题可以为顺序文件配置一个运行记录文件(log file)或称为事物文件(transactionFile)把试图增加删除或者修改的信息记录于其中,规定每个一定时间例如 4 个小时,将运行记录文件于原来的主文件加以合并,产生一个按关键字排序的新文件

**36.索引文件**使用索引文件的主要问题是,它除了有主文件外,还须配置一张索引表,而且每个记录都要有一项索引项,因此提高了存储费用。

**37.文件分配方式:连续分配**连续分配要求为每一个文件分配一组相邻接的盘块。一组盘块的地址定义了磁盘上的一段线性地址。连续分配的主要优点如下: 1. 顺序访问容易。2.顺序访问速度快。连续分配的主要缺点如下: 1. 要求有连续的存储空间。2.必须事先知道文件的长度。**链接分配(隐式链接,显示链接)**隐式链接: 用采用隐式链接分配方式时,在文件目录的每个目录项中,都须含有指向链接文件的第一个盘块和最后一个盘块的指针。

显示链接:这是指把链接文件各物理块的指针,显示的存放在内存的一张链接表中。该表整个磁盘仅设置一张。

**41.快表概念:** 具有并行查询能力的特殊高速缓冲寄存器,用以存放当前访问的那些页表项,虚拟存储器的基本概念,是指具有请求调入功能和置换功能,能从逻辑上对内存容量加以扩充的一种存储器系统,其逻辑容量,由内容容量和外存容量之和所决定,其运行速度接近于内存速度,而每位的成本却有接近于外存。

**42.进程控制块 PCB 的概念** 它是就弄成实体的一部分.是操作系统中最重要的记录性数据结构,PCB 中记录了操作系统所需的用于描述进程的当前情况以及控制进程运行的全部信息信号量的应用 : 利用信号量进行实现进程互斥,利用信号量实现前趋关系

**43.选择调度方式和调度算法的准则** 面向用户的准则: 周转时间短 响应时间快 截止时间的保证 优先权准则 面向系统的准则: 系统吞吐量高 处理器利用率好 各类资源的平衡利用

**4.4.目录管理的要求:** 1. 实现“按名存取”2 提高对目录的检索速度  
3 文件共享 4.允许文件重名。

**1.先来先服务算法:** A.作业调度中: 从后备的队列中选择最优先进入队列的作业放入内存中, 并分配资源创建进程, 再放入就绪队列。B 进程调度中, 从就绪队列中选择多个(先进先出)进程再分配处理机运行, 进程将一直运行到完成或遇到阻塞后才放弃处理机。此算法有利于长作业不利于短作业。(周转时间=完成时间-到达时间; 带权周转时间=周转时间/服务时间)

**3.最早截止时间 EDF:** 根据任务的开始截止时间来确定任务的优先级, 截止时间越早, 优先级越高。A 非抢占式调度方式用于非周期实时任务 B 抢占式调度方式用于周期实时任务。

**5.银行家算法:**某系统中有10台打印机, 有三个进程 P1 P2 P3 分别需要8台 7台 4台 若 P1 P2, P3已申请到4台, 2台和2台。试问: 按银行家算法能安全分配吗? 答:申请后 系统把2台机分配给 p3, p3完成后释放所有的资源4, 再分配给 p1, p1完成后释放8, 再分配给 p2 安全状态:指系统能按着某种进程顺序 (p1,p2,...pu)来为每个进程 pi 分配其所需资源,知道满足每个进程对资源的最大需求,是每个进程都可顺利的完成

**6.物理块分配算法:** A 平均分配算法: n 个物理块, m 个进程, 则每个进程分配  $n/m$

B 按比例分配算法: n 个进程, 每个进程页面数为 a 则  $S=n$  个页面总数相加。系统中可用的物理块数为 m 则每个进程分配的物理块数为  $b=a/s * m$ 。C 考虑优先级的分配算法。将物理块分为两部分: 一部分按比例分配给进程, 另一部分则根据进程的优先级分配。

**7.先进先出 (FIFO) 置换算法:** 该算法是淘汰最先进入内存的页面。用队链实现, 并设置一指针指向最老页面。缺点: 与进程实际运行规律不适应, 有些常用页面会被淘汰

**8.LRU 页面置换算法:** 根据页面调入内存后的使用情况进行决策选择最近最久未使用的页面予以淘汰。该算法需要两硬件之一的支持: a 寄存器: 每个寄存器初始化为 0, 某个页面被访问, 将其寄存器最高位 R (n-1) 置为 1; 每隔一段时间 (如 100ms) 将所有寄存器右移一位; 如果寄存器最后停止时, 将 R 值 (二进制) 最小的淘汰。 B 栈: 每当页面被访问时, 该页面的页面号从栈中移出, 将它存入栈顶。而在栈底便是最久未使用的页面号将淘汰

**9.FAT 技术: (以 FAT12 为例)** 1.以盘块为基本分配单位: 对于 1.2MB 的软盘, 每个盘块的大小为 512B, 则每个 FAT 中含有  $(1.2MB/512B=2.4K)$  个表项, 由于 FAT12 每个表项占 12 位, 则 FAT 表占用  $12/8*2.4=3.6KB$  的存储空间。由于每个 FAT 表项为 12 位, 则 FAT 最多允许有  $2^{12}=4096$  个表项, 每个表项为 512 字节, 则每个磁盘分区的容量为 2MB。一个物理磁盘有 4 个逻辑磁盘分区, 则最大容量为 8MB。2.簇的基本概念: 簇是一组连续的扇区, 在 FAT 中她是作为一个虚拟扇区, 簇的大小一般是 2N 个盘块。最大可包含 8 个扇区 64mb

优点: 能适应磁盘容量不断增大的情况。缺点: 对所允许的磁盘容量存在着严重的限制, 通常只能是数兆字节, 随着支持的硬盘容量的增加, 相应的簇内碎片也将随之成倍地增加

FAT16 同理就是  $2^{16}*64*512=2G$ , 就是扩到了 16 位。但 FAT12 FAT16 都不支持长文件名, 文件名收到了 8 个字符文件名和三个

字符文件扩展名的长度限制。FAT32 优点: 支持更小的簇和更大的磁盘容量, 这就大大减少了磁盘空间的让费, 使区分空间分配更有效率。缺点: 1.文件分配表的扩大运行速度比较慢 2.有最小管理空间的限制, 对于小分区不支持。3.棋单个文件长度不能大于 4gb5.最大的限制在于不支持向下兼容。

**10.NTFS 技术:** 1.新特性: 64 位磁盘地址, 可以支持 2 的 64 次方字节的磁盘分区; 在 ntfs 中可以很好的支持长文件名; 具有系统容错功能; 提供了数据的一致性; 此外还提供了文件加密、文件压缩等功能。

**11.伙伴系统:** 固定分区方式和动态分区方式的一种折衷方案, 伙伴系统规定, 无论已分配分区或空闲分区, 其大小均为 2 的 K 次幂, K 整数,  $1 \leq K \leq M$ , 其中: 2 的一次幂表示最小分区的大小, 2 的 m 次幂表示分配的最大分区的大小, 通常 2 的 m 次幂是整个可分配内存的大小。

**12.SPOOLing 技术:**为了缓和 cpu 的高速型于 IO 设备低速性间的矛盾而引入了脱机输入,脱机输出技术,该技术是利用专门的外围控制机,将低速 IO 设备上的数据传送到高速磁盘上,或者相反,事实上当系统中引入了多道程序技术后,完全可以利用其中的一道程序,来模拟脱机输入是的外围控制机功能,把数据从磁盘传送到低速输出设备上 **SPOOLing 系统的组成:** SPOOLing 系统是对脱机 I/O 工作的模拟,其必须有高速随机外存(通常采用磁盘)的支持 **SPOOLING 系统有以下四个部分:** 1.输入井和输出井,为磁盘上开辟的两大存储空间,分别模拟脱机输入/出时的磁盘并用于收容 I/O 设备输入的数据和用户程序的输出数据 2.输入缓冲区和输出缓冲区,在内存中开辟,分别用于暂停由输入设备和输出井送来的数据 3.输入进程 SPi 和输出进程 SPO 分别模拟脱机输入/出时的外围控制机,用于控制 I/O 过程 4.I/O 请求队列,由系统为各个 I/O 请求进程建立的 I/O 请求表构成的队列.廉价磁盘冗余阵列:利用一台磁盘阵列控制器,来统一管理和控制一组磁盘驱动器,组成一个高度可靠的快速的大容量磁盘系统

## 生产者—消费者问题

### 1. 利用记录型信号量解决生产者—消费者问题

假定在生产者和消费者之间的公用缓冲池中，具有  $n$  个缓冲区，这时可利用互斥信号量  $\text{mutex}$  实现诸进程对缓冲池的互斥使用；利用信号量  $\text{empty}$  和  $\text{full}$  分别表示缓冲池中空缓冲区和满缓冲区的数量。又假定这些生产者和消费者相互等效，只要缓冲池未空，生产者便可将消息送入缓冲池；只要缓冲池未空，消费者便可从缓冲池中取走一个消息。

```
Var mutex, empty, full: semaphore := 1, n, 0;
buffer: array [0, ..., n-1] of item;
in, out: integer := 0, 0;
begin
  parbegin
    producer: begin
      repeat
        producer an item nextp;
        wait(empty); wait(mutex);
        buffer(in) := nextp;
        in := (in+1) mod n;
        signal(mutex); signal(full);
      until false;
    end
    consumer: begin
      repeat
        wait(full); wait(mutex);
        nextc := buffer(out);
        out := (out+1) mod n;
        signal(mutex); signal(empty);
        consumer the item in nextc;
      until false;
    end
  parend
end
```

**在生产者—消费者问题中应注意：**首先，在每个程序中用于实现互斥的  $\text{wait}(\text{mutex})$  和  $\text{signal}(\text{mutex})$  必须成对地出现；其次，对资源信号量  $\text{empty}$  和  $\text{full}$  的  $\text{wait}$  和  $\text{signal}$  操作，同样需要成对地出现，但它们分别处于不同的程序中。例如， $\text{wait}(\text{empty})$  在计算进程中，而  $\text{signal}(\text{empty})$  则在打印进程中，计算进程若因执行  $\text{wait}(\text{empty})$  而阻塞，则以后将由打印进程将它唤醒；最后，在每个程序中的多个  $\text{wait}$  操作顺序不能颠倒。应先执行对资源信号量的  $\text{wait}$  操作，然后再执行对互斥信号量的  $\text{wait}$  操作，否则可能引起进程死锁。

### 利用管程解决生产者—消费者问题

(1)  $\text{put}(\text{item})$  过程。生产者利用该进程将自己生成的产品投入到缓冲池中，并用整型变量  $\text{count}$  来表示在缓冲池中已有的产品数目，当  $\text{count} \geq n$  时，表示缓冲池已满。生产者等待。

(2)  $\text{get}(\text{item})$  过程。消费者利用该进程从缓冲池中取出一个产品，当  $\text{count} \leq 0$  时，表示缓冲池中已无可取用的产品，消费者等待。PC 管程可描述如下：

```
type producer-consumer = monitor
var in, out, count: integer;
buffer: array [0, ..., n-1] of item
notfull, notempty: condition;
procedure entry put(item)
```

```
begin
  if count >= 0 then notfull.wait;
  buffer(in) := nextp; in := (in+1) mod n;
  count := count + 1; if notempty.queue
then notempty.signal;
end
procedure entry get(item)
begin
  if count <= 0 then notempty.wait;
  nextc := buffer(out); out := (out+1) mod n;
count := count - 1;
  if notfull.queue then notfull.signal
end
begin in := out := 0; count := 0
end
利用管程解决生产者——消费者问题时。
produce: begin
  repeat produce an item in nextp PC.put(item)
until false;
end
consumer: begin
  repeat PC.get(item) consumer the item in nextc
until false;
end
```