

操作系统复习课 2022 总结

第一部分 绪论

1. 操作系统的基本概念、操作系统在计算机系统中的地位和作用

- 操作系统的定义：**
- a. 用户的观点：是方便用户使用计算机的用户界面
 - b. 系统管理的观点：是计算机系统资源的管理者
 - c. 开放的观点：是能够协调系统有条不紊允许的程序集合

操作系统的地位：是裸机之上的第一层软件，是其他软件与硬件之间的接口。

操作系统的作用：作为用户与计算机硬件系统之间的接口；
作为计算机系统资源的管理者。

2. 操作系统的五大功能模块

处理机管理、储存管理、设备管理、文件管理、用户接口

3. 操作系统的两种用户接口类型



程序性接口，提供一组系统调用供用户和其他系统程序调用
作业级接口，提供一组控制操作命令供用户组织和控制自己作业的运行

4. 操作系统的三个基本类型及其特点

多道批处理系统

- a. 多道性：内存可同时驻留多道程序
- b. 无序性：作业进入内存顺序与完成顺序无对应关系
- c. 调度性：作业经过两次调度

分时系统

- a. 同时性：若干个用户可同时使用系统资源
- b. 独立性：系统中各个用户可以彼此独立操作，互不干扰或破坏
- c. 及时性：用户能在很短时间内得到系统的响应
- d. 交互性：用户可通过终端方便地与系统进行交互

实时系统

能及时地响应外部请求，在规定时间内完成对该事件的处理

5. 多道程序设计技术及其本质

多道程序设计技术：根据其对资源的要求和一定的调度原则，调入几个程序进入内存，交替允许，共享系统中的资源。

本质：计算机内存中同时存放多道相互独立的程序，进入系统的几道程序都处于运行状态，轮流使用 CPU，交替执行。

6.现代操作系统的特征

并发性：两个或多个事件在同一时间间隔内发送。

注：与并行性（两个或多个事件在同一时刻发送）区分

共享性：系统中的资源可供内存中多个并发执行的进程共同使用。

虚拟性：通过某种技术把一个物理实体变成若干个逻辑上的对应物。

异步性：进程所处状态是不确定的。

第二部分 处理机管理(进程管理+处理机调度)

1.进程的基本概念、进程控制块的概念和作用

进程：一个正在执行的程序

进程控制块 PCB：系统为了管理进程设置的一个专门的数据结构

作用：是使一个在多道程序环境中不能独立运行的程序，成为一个能独立运行的基本单位，一个能与其他进程并发执行的进程。是支持多进程和提供多进程处理的关键工具

2. 现代操作系统对进程与线程的分工

进程是资源分配的基本单位，线程是系统调度的基本单位。

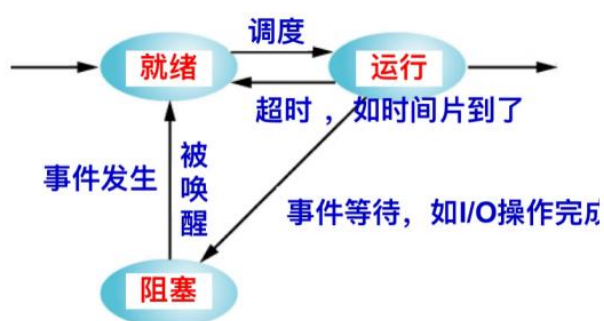
3. 进程三状态模型(包括可能发生的状态转换及其引发状态转换原因，会用图示加以说明)；

三种基本状态：

运行态：该程序正在执行。

就绪态：进程做好了准备，只要被调度就可执行。

阻塞态：进程在某些事件发生前不能被执行



(1) 就绪态>>运行态：处于就绪态的进程被程序调度程序选中后，就分配到处理机上运行。

(2) 运行态>>就绪态：处于运行态的进程在其运行过程中，分给它的处理机时间片用完而让出处理机。

(3) 运行态>>阻塞态：当进程请求某个资源且必须等待时。

(4) 阻塞态>>就绪态：当进程的某等待事件到来时，进程从阻塞态转换到就绪态。

4.进程五状态模型及其状态转换：如上，加入新建态和终止态

5.进程之间存在的制约关系及其定义，能够分析具体是哪种制约关系

并发执行的进程，由于资源共享存在着间接制约，需要互斥；

由于进程合作存在着直接制约，需要同步。

同步：进程之间通过在执行时序上的某种现实而达到相互合作的约束关系。

互斥：进程之间本来是无关系的，但是由于竞争使用同一共享资源而产生相互制约的关系，这种因共享资源而产生的制约关系称为互斥。

6.临界资源、临界区

临界资源

概念：凡是以互斥方式使用的共享资源都称为临界资源。

特征：具有一次只允许一个进程使用

临界区

概念：每个进程中，访问临界资源的那段代码称为临界区

分类：

进入区：在进入临界区之前先检测是否能够进入。若能进入，允许进入，同时设置标志表明有进程正在临界区内。

退出区：退出临界区后，要修改标志为无进程使用该临界资源

6. 信号量的含义、信号量的取值范围

定义：是一种不存在“忙等”现象的进程同步机制。

物理意义：代表可分配的临界资源

取值范围：整数（包括负整数和正整数）

7. 用信号量机制解决同步和互斥的算法设计问题（见 PPT 综合题）

解决互斥问题

```

/* 互斥 */
semaphore s=1;
void P(int i)
{
    while(true)
    {
        Wait(s);
        ...//临界区
        Signal(s);
        ...//剩余部分
    }
}
void main()
{
    parbegin( P(1),P(2),...,P(n) );
}

```

挂起主程序执行，
初始化并发过程
P1,P2,...Pn

解决同步问题

empty、full—同步信号量，表示该进程是否已经开始或该进程是否已经结束。

```

semaphore empty=1,full=0;
void Pc( )
{
    while(true)
    {
        计算
        Wait(empty)
        放数
        Signal(full)
    }
}
void Pp( )
{
    while(true)
    {
        Wait(full)
        取数
        Signal(empty)
        打印
    }
}
void main()
{
    parbegin( Pc,Pp );
}

```

8. 死锁的基本概念及产生死锁的四个必要条件

死锁定义：进程间因循环等待资源出现了永久性的阻塞现象。

产生的原因：竞争资源、进程推进顺序不当

产生的条件

互斥条件：一次只有一个进程可以使用该资源。

请求且保持条件：一个进程在等待资源时，继续占有已分配得到的资源。

不可抢占条件：不能强行抢占进程中已占有的资源。

循环等待条件：存在一个封闭的进程-资源链，使得每个资源至少占有此链中下一个进程所需要的资源

常用的解决方法：死锁预防、死锁避免、死锁检测与接触、死锁忽略

9. 安全状态与不安全状态，银行家算法避免死锁；非安全状态一定死锁吗？

安全状态：安全状态是指系统能按某种进程推进顺序，为每个进程分配所谓资源，直至满足每个进程对资源的最大需求，使每个进程都顺利完成，若能找到这样的进程推进顺序，则称其为安全序列

银行家算法（见综合题）

10.、 处理机调度的调度层次、处理机调度的两种调度方式及其特点

调度层次：

高级调度：又称为**作业调度**或**长程调度**，主要用于批处理系统。负责从后备队列中选择多个作业调入内存，为它们创建进程并分配必要的资源，然后链接到就绪队列。

低级调度：又称**进程调度**或**短程调度**。负责选择一个就绪进程占有 CPU，进入运行态。

中级调度：又称**交换调度**或**中程调度**。按照一定的策略，将进程从外存调到内存或从内存调到外存。

处理机的两种调度方式：

非抢占调度方式：一旦进程占用 CPU，将一直执行，直至完成或阻塞，才将 CPU 分配给其他进程。

抢占调度方式：进程在执行过程中，系统可以根据抢占原则，将已分配的 CPU 重新分配给其他进程。

11.处理机调度算法：FCFS，RR，SPN,HRRN，最高优先级调度；在某种调度算法下，分析作业流或进程流的调度顺序、周转时间等指标值

FCFS（先来先服务调度算法）-最简单的调度算法，用于作业调度和进程调度-非抢占按照作业或进程到达系统的先后次序进行调度。

算法特点：a、简单，易于实现。

b、FCFS 算法有利于长进程，不利于短进程。

c、有利于 CPU 繁忙的进程，不利于 I/O 操作繁忙的时候

SPF（短进程优先算法）-非抢占式

后备队列中选择一个或多个估计运行时间最短的作业，将它们调入内存运行。

算法优点：

可以缩短作业平均周转时间，提高系统的吞吐量

算法缺点：

a、不利于长作业。

b、不利于紧急任务。

c、由于估计的运行时间不准确，不一定能真正做到短作业优先。

3RR（轮转调度算法）-抢占式

采用基于时钟的抢占策略。系统将所有的就绪进程按 FIFO 排成一个队列，将 CPU 分配给队首进程，且规定只能执行一个时间片，时间片到了，就会调度下一个队首进程。

算法特点：简单易行，平均效应时间短，但不利于处理紧急任务。

HRRN（高响应比优先调度算法）-非抢占式

调度时，选择响应比最高的进程占有 CPU 执行

$$\text{响应比} R_p = \frac{\text{估计周转时间}}{\text{要求服务时间}} = \frac{\text{等待时间} + \text{要求服务时间}}{\text{要求服务时间}}$$

算法结论：

- a、若等待时间相同，则短作业的响应比高，短作业优先
- b、若执行时间相同，则等待时间长的优先，先来先服务
- c、长作业会随着等待的时间增长，响应比提高
- d、但每次调度前，都需要计算响应比，增加了系统开销

具体计算和应用见综合题

第三部分 存储管理(常规存储管理方法和虚拟内存管理)

1.物理地址、逻辑地址的区别，地址重定位的种类及各自的优缺点

物理地址是绝对地址，逻辑地址是相对地址

地址重定位：对装入的指令和数据的地址进行变换的过程！即逻辑地址>>物理地址。

静态地址重定位

发生在装入内存的过程中，在程序运行之前就完成了地址重定位。

物理地址 = 装入到内存的起始地址+逻辑地址

优点：实现简单，无需硬件支持

缺点：

- a、用户程序只能装入内存的一个连续储存区域上，不能装入多个离散区域。
- b、用户程序地址重定位后不允许在内存中移动

动态地址重定位

发生在程序执行期间，当执行到某条指令且该指令需要进行内存访问时，再将逻辑地址转换为相应的物理地址。

物理地址 = 重定位寄存器的值+逻辑地址

优点：

- a、支持程序执行过程中在内存的移动

原理：将内存的用户区划分为多个区域，每个区域称为一个分区，每个分区允许装入至多一个用户程序，以实现内存同时存储多个运行程序的目的。

b、支持程序在内存的离散存储

缺点：

需要附加硬件支持

2. 分区存储管理的工作原理、采用的数据结构、

A、概念：将内存空间划分为若干个固定分区，每个分区可装入一道作业。是实现多道程序的最简单的一种存储管理。

B、划分方式：

a、分区大小相等 b、分区大小不等

C、缺点：

a、分区的数目在系统生成阶段已经确定，限制了系统中活动的进程的数目。

b、分区固定不变，内存碎片数量较多，导致内存空间的浪费。

(2) 动态分区存储管理

A、概念：分区不是事先固定好的，而是在作业的处理过程中，依据其所需内存大小动态划分。

B、采用的数据结构：

a、空闲分区表 b、空闲分区表和空闲分区链

(3) 动态重定位分区存储管理

实施紧凑技术的动态分区方式

(4) 三者关系

为了满足多道程序同时运行 → 固定分区 → 为了解决内部碎片问题 → 动态分区 → 为了提高内存空间利用率 → 动态重定位分区

3. 基本分页存储管理的工作原理、结合图示描述地址变换过程、分页存储管理的特点，页表、快表数据结构的作用

基本分页存储管理

表示为：页号，页内相对地址

页表：

a、概念：记录页与物理块的对应关系。

b、作用：实现从页号到物理块号的地址映射 c、每个页表在内存中占有一块固定的存储

区

地址变换

A、地址变换机构：用于实现从逻辑地址转换为内存空间的物理地址

B、页表寄存器：用于存放页表在内存的始址和页表的长度

C、地址变换过程

a、由分页地址变换机构将逻辑地址分为页号和页内地址两部分；

b、根据页表寄存器存放的页表始址找到页表；

c、若页号>=页表长度，则出现错误，发生越界中断；

d、若没有，则找到页号对应的物理块号；

e、物理地址=物理块号*页面大小+页内地址

设置快表的目的：提高地址变换的速度。

优点

A、能够适应多道程序设计技术

B、相比分区存储管理，实现了离散存储，减少碎片，可以提高内存利用率

缺点

A、仍存在页内碎片

B、不方便进行信息共享

C、不支持动态增长

D、当没有足够空间装下整个作业时，无法运行

4. 基本分段存储管理的工作原理、结合图示描述地址变换过程、分段存储管理的特点，快表数据结构的作用，分页与分段的区别

分段：分段管理中，将程序按逻辑分成若干个段，每个段都有自己的名字

段的长度不等，从 0 开始编址，允许动态增长。

逻辑地址：段号 s ，段内相对地址 w

逻辑地址采用的是二维地址

B、数据结构

a、分区说明表：记录内存分配情况

b、可用分区表/可用分区自由链：记录空闲分区情况

c、段表：用于实现从逻辑段到物理内存的映射。

优点

满足用户的需要，方便编程、信息的共享、支持动态增长等。

缺点

段与段之间可以离散存储，但同一个段必须连续存储，若找不到连续的存储区，则分配失败

分页和分段的基本区别：

页是信息的物理单位；段是信息的逻辑单位，它含有一组有意义的相对完整的信息。

分页存储的目的是为了离散分配，提高内存的利用率，仅仅是为了系统管理的需要；

分段的目的是为了更好的满足用户的需要。

页的大小是固定的，有系统决定；而段的长度是不固定的，取决于用户所编写的程序。

分页的逻辑地址是一维的，只需一个编号就可以表示一个逻辑地址；分段的逻辑地址是二维的，标识一个地址时，需要给出段名和段内地址。

5. 虚拟存储器的基本概念、特点

是指具有调入功能和置换功能，且能从逻辑上对内存容量进行扩充的一种储存器系统，其逻辑容量由内存容量和外存容量之和所决定，其运行速度接近于内存速度，而成本又接近于外存。

特征

多次性：虚拟储存器将一个作业分成多次调入内存。是虚拟储存器最重要的特征。

对换性：作业运行期间，虚拟储存器运行将那些暂不使用的程序或数据从内存调至对换区，待以后需要时再调入内存。

虚拟性：对内存的扩充是逻辑上的。虚拟性是实现虚拟储存器的目标。

6. 局部性原理、抖动

局部性原理：在一段时间内，CPU 总是集中地访问程序中的某一部分而不是随机地对程序所有部分具有平均访问概率。

表现方面

A、空间局限性：程序在执行时访问的内存单元会局部在一个比较小的范围内。这反映了程序

顺序执行的特性，也反映了程序顺序访问数据结构的特性。

B、时间局限性：产生原因是由于程序中存在大量的循环操作。

地位：局部性原理为虚拟存储器的引入奠定了理论基础。

抖动：刚刚被换出的页，随后又要访问该页面，造成频繁的磁盘输入输出，降低了系统运行效率。

产生原因：操作系统难以预测程序执行的轨迹，导致刚刚被换出的页，随后又要访问该页面，从而导致频繁的磁盘输入输出

7.请求调页存储管理、请求调段存储管理的工作原理、结合图示描述地址变换过程、各自的特点、页面置换算法（见综合题）

请求分页存储管理

在基本分页存储管理的基础上，增加了请求调页功能和页面置换功能

缺页中断机构

工作流程：

查页表，获得缺页在外存起始地址

内存有可用的物理块：启动磁盘 I/O 将其调入

内存没有可用物理块：页面置换

若淘汰页做过修改，置换前需重写回外存。

最后将缺页调入内存指定的物理块。

C、地址变换机构

- 根据逻辑地址**计算页号和页内地址**。检查**页号是否越界**；若是，则产生越界中断；反之，**先去快表检索**。
- 若在快表中检索到，更新访问位、改动位，利用物理块号和页内地址组合，**形成物理地址**。
- 若在快表中未找到，**再到页表中找**。
 - 若状态位显示在内存，则利用物理块号和页内地址，**形成物理地址**，更新页表项和快表。
 - 若状态位显示不在内存，则产生**缺页中断**，转缺页中断程序，**待缺页调入内存**后，根据加载的物理块号和页内地址，**形成物理地址**。

请求分段存储管理

在基本分段存储管理的基础上，增加了请求调段功能和段置换功能的虚拟存储系统。

- ▶查段表，获得缺段在**外存的起始地址**。
- ▶内存有**合适空闲区**吗？
 - ◆有，启动磁盘I/O，将**缺段调入**内存；
 - ◆没有，那**多个空闲区之和满足要求**吗？
 - 满足，**合并空闲区，调入缺段**；
 - 不满足，**段置换**。
- ▶**更新段表**和相应的数据结构。

C、地址变换机构

检查段号是否越界 ——越界中断

检查段长是否越界 ——越界中断

检查是否符合段的存取方式 ——保护权中断

检查段是否在内存

产生缺段中断，由缺段中断机构负责将缺段调入，并更新段表信息。

将段在内存的始址与段内地址组合，形成物理地址。

页面置换算法：

OPT（最佳置换算法）

选择淘汰那些以后再也不会被访问，或者是在未来最长时间不会再被访问地页面。

优点：可以保证获得最低的缺页率

缺点：只是一种理想化的算法，实际无法应用。

FIFO（先进先出置换算法）

选择最先进入内存的页面，即在内存中驻留时间最久的页面予以淘汰。

优点：实现简单

缺点：缺页率很高，抖动现象严重

LRU（最近最久未使用置换算法）

选择最近最久未使用的页面予以淘汰。

8. 分页或分段虚拟内存的地址变换计算(十进制地址、十六进制地址)

见综合题

第四部分 设备管理

1. 缓冲区的作用

缓和 CPU 与 I/O 设备间速度不匹配的矛盾，提高 CPU 和 I/O 设备之间的并行性。

减少对 CPU 的中断频率，放宽对中断响应时间的限制。

解决数据粒度不匹配的问题

2. 设备独立性的概念

用户编写的应用程序独立于具体使用的物理设备

3. 虚拟设备技术的概念、Spooling 系统的组成：从虚拟设备技术角度理解共享打印机的实现

Spooling 系统由磁盘上的输入井和输出井，内存中的输入缓冲区和输出缓冲区、输入进程和输出进程以及井管理程序构成

共享打印机：

- 当用户进程请求打印输出时，SPOOLing 系统**立即同意**其打印请求，但并不真正把打印机分配给它们，而是为每个进程做两件事情。
 - 由**输出进程在输出井中**为它申请一个**空闲缓冲区**，并将要打印的数据送入其中。
 - **输出进程**再为用户进程申请一张空白的**用户打印请求表**，并将用户的打印请求填入表中，将该表挂到**打印队列**上。

至此，用户进程认为它的打印过程已经完成，而不必等待真正的慢速的打印过程的完成。

当**打印机空闲**时，输出进程将从请求队列队首取出一张打印申请表，根据表中的要求将打印的数据从输出井传送到内存缓冲区，再由打印机进行输出打印。

打印完后，处理队列中的下一个打印请求。

结论：虽然系统中只有一台打印机，但系统并未将它分配给任何进程，而只是为每个提出打印请求的进程在输出井中分配一个存储区（相当于一个逻辑设备），使每个用户进程都觉得自己在独占一台打印机，从而实现了共享。

第五部分 文件管理

1. 文件的基本概念，文件系统的基本功能

实现“按名存取”。

文件存储空间的管理。

对文件及文件目录的管理。

文件组织。

提供文件共享和保护等机制。

2. 文件逻辑结构的种类；文件物理结构的种类

逻辑结构：流式文件/无结构文件

记录式文件/有结构文件（顺序、索引、索引顺序结构）

物理结构：**顺序结构、索引结构、串联结构**

3. 磁盘空间管理方法

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。

特点：

- A、使用方便，简单易行
- B、但分配时需进行穷尽搜索扫描
- C、占用额外的空间存放位图

空闲块链接法

将所有空闲区链接成一条空闲链

优点：相对于位示图不占有额外的空间

缺点：删除和分配非常耗时。

空闲表法

系统为文件存储空间建立一张空闲表，其中的每个表项对应于一个空闲区，并登记有该空闲区的起始块号和块数等信息

优点：空闲区的分配和回收比较容易

缺点：空闲盘块表需要占有大量的存储空间

4. 文件控制块的基本概念、目录组织方式及各自特点

文件控制块 FCB: 为了能对文件进行正确地存取，就必须为文件设置于描述和控制文件的数据结构。

文件与文件控制块一一对应，并分别存放。

三种目录结构形式

单级目录结构

概念：整个文件系统中只建立一张目录表，每个文件占其中一个表项

特点：不允许文件重名、查找速度慢

优点：

- a、简单
- b、能实现“按名存取”
- D、缺点：
 - a、查找速度慢
 - b、不允许重名
 - c、不便于实现文件共享

两级目录结构

两级目录是指用户目录和主目录：

- a、用户目录：系统为每个用户建立一个单独的用户文件目录，将该用户所有文件的文件控制块登记其中。
- b、主目录：系统中还建立一个主目录，它的目录项对应于一个用户，包含该用户的用户名和指向该用户文件目录的指针

优点：

- a、提高了检索速度
- b、在不同的用户目录中，文件可以重名
- c、不同用户可使用不同文件名来共享一个文件

缺点：不提供用户建立子目录的手段，使用起来不方便。

树型目录

三级及三级以上的文件目录结构

优点：检索效率高、允许重名、便于实现文件共享

第六部分 上机部分

掌握常用系统调用功能

题型及分值分配

1. 选择题 20 分(2*10)
2. 填空题 10 分(2*5)
3. 理解分析题 20 分(2-3 个小题)
4. 算法设计题 10 分
5. 应用题 40 分(银行家算法、处理机调度、地址变换)

